

```

package gov.dhs.mobile.efoia;

import android.app.Activity;
import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.content.Intent;

import android.widget.ListView;
import android.widget.RelativeLayout;
import android.widget.Toast;

import java.util.List;
import java.util.Random;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.tasks.AdvertisementTask;
import gov.dhs.mobile.efoia.tasks.GetBackgroundImageTask;
import gov.dhs.mobile.efoia.viewcontrollers.events.EventsActivity;
import
gov.dhs.mobile.efoia.viewcontrollers.exemptions.ExemptionsActivity;
import gov.dhs.mobile.efoia.viewcontrollers.howto.FOIAHowToActivity;
import gov.dhs.mobile.efoia.viewcontrollers.library.FOIALibrary;
import
gov.dhs.mobile.efoia.viewcontrollers.privacypolicy.PrivacyPolicyActivi
ty;
import
gov.dhs.mobile.efoia.viewcontrollers.privacystatement.PrivacyStatement
Activity;
import gov.dhs.mobile.efoia.viewcontrollers.status.FOIAStatus;
import gov.dhs.mobile.efoia.viewcontrollers.statutes.FOIAStatutes;
import gov.dhs.mobile.efoia.tasks.GetMobileMenusTask;
import gov.dhs.mobile.efoia.arrayadapters.HomeViewTableArrayAdapter;

/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,

```

```

* | if any. The intellectual and technical concepts contained
* | herein are proprietary to XXXXXXXXXXXXX
* | and its suppliers and may be covered by U.S. and Foreign Patents,
* | patents in process, and are protected by trade secret or
copyright law.
* | Dissemination of this information or reproduction of this
material
* | is strictly forbidden unless prior written permission is obtained
* | from XXXXXXXXXXXXX.
* |
*****
* | <Insert your Javadoc Here>
* |
*****
*/
public class MainActivity extends Activity {

    private ListView activityListView;
    private List<MobileMenu> menus;
    private List<Advertisement> advertisementList;
    private Context mContext = this;
    private GestureDetector gestureDetector;
    private int currentIndex;
    private RelativeLayout layout;
    private HomeViewTableArrayAdapter adapter;
    public static final String PREF_FILE_NAME = "eFOIA-Prefs";
    public static final String POLICY_ACCEPTED = "accepted";
    public static int mSelectedItem=-1;

    private TaskListener<AdvertisementResponse>
advertisementTaskListener = new TaskListener<AdvertisementResponse>()
{
    @Override
    public void onTaskSuccess(AdvertisementResponse result) {
        if (result != null && result.getAdvertisements() != null
&& result.getAdvertisements().size() > 0) {
            advertisementList = result.getAdvertisements();

            gestureDetector = new GestureDetector(mContext, new
GestureListener());

            layout.setOnTouchListener(new View.OnTouchListener() {
                @Override
                public boolean onTouch(View view, MotionEvent
motionEvent) {
                    return
gestureDetector.onTouchEvent(motionEvent);
                }
            });
        }
    }
};

```

```

        Random rand = new Random();

        rand.setSeed(advertisementList.size());

        currentImageIndex =
rand.nextInt(advertisementList.size());

        Advertisement ad =
advertisementList.get(currentImageIndex);

        String imageUrl =

ad.getImage().getCRIImage().getImageForDensity(CRIImage.IMAGE_TYPE.FULLS
CREEN,
                                                    (Activity) mContext);

        try
        {
            layout.setBackground(new
GetBackgroundImageTask().execute(imageUrl).get());
        }
        catch (Exception e)
        {
        }
    }
}

@Override
public void onTaskFailure(Exception e) {

}

@Override
public void onTaskCancelled() {

}

@Override
public void onTaskTimeOut() {

}

};

private TaskListener<MobileMenuResponse> mobileMenusTaskCallback =
new TaskListener<MobileMenuResponse>() {
    @Override
    public void onTaskSuccess(MobileMenuResponse result) {
        if (result != null && result.getMobileMenus() != null &&

```

```

result.getMobileMenus().size() > 0) {

        menus = result.getMobileMenus();
        adapter = new
HomeViewTableAdapter(MainActivity.this,
                        R.layout.home_table_list_item, menus);
        activityListView.setAdapter(adapter);

        SharedPreferences prefs =
getSharedPreferences(PREF_FILE_NAME,MODE_PRIVATE);

        boolean accepted =
prefs.getBoolean(POLICY_ACCEPTED,false);

        if (!accepted)
        {
            Intent intent = new Intent(MainActivity.this,
PrivacyPolicyActivity.class);
            Bundle bundle = new Bundle();
            bundle.putString("action_object",
"privacy_policy");
            bundle.putBoolean("first_launch",true);
            intent.putExtras(bundle);
            startActivity(intent);
        }
    }

    @Override
    public void onTaskFailure(Exception e) {
        Toast.makeText(MainActivity.this, "Error: " +
e.getLocalisedMessage(), Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        new GetMobileMenusTask(MainActivity.this,
mobileMenusTaskCallback).execute();
        new AdvertisementTask(MainActivity.this,
advertisementTaskListener).execute();

        activityListView = (ListView)
findViewById(R.id.activity_main_lv);
        layout = (RelativeLayout) ((Activity)
mContext).findViewById(R.id.relativeLayout);

        activityListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {

                MobileMenu menu = menus.get(position);

                mSelectedItem = position;
                adapter.notifyDataSetChanged();

                if (menu.getAction().equalsIgnoreCase("subcategory"))
{
                    launchSubMenu(menu, menu.getTitle());

                } else if
(menu.getAction().equalsIgnoreCase("dynamic_contents")) {
                    launchContentScreen((String)
menu.getActionObject());
                } else if
(menu.getAction().equalsIgnoreCase("tagged_contents")) {
                    //Surveys use tagged contents
                } else if
(menu.getAction().equalsIgnoreCase("no_action")) {
                    launchAction(menu);
                }
            }
        });
    }
}

```

```

private void launchContentScreen(String action) {
    Intent intent = null;

    Bundle bundle = new Bundle();

    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this,"Main Menu",action,"","");

    if (action.equalsIgnoreCase("privacy_policy")) {
        intent = new Intent(MainActivity.this,
PrivacyPolicyActivity.class);
    } else if (action.equalsIgnoreCase("library")) {
        intent = new Intent(MainActivity.this, FOIALibrary.class);

    } else if (action.equalsIgnoreCase("statutes")) {
        intent = new Intent(MainActivity.this,
FOIAStatutes.class);
    } else if (action.equalsIgnoreCase("events")) {
        intent = new Intent(MainActivity.this,
EventsActivity.class);

    } else if (action.equalsIgnoreCase("exemptions_exclusions")) {
        intent = new Intent(MainActivity.this,
ExemptionsActivity.class);
    } else if (action.equalsIgnoreCase("foia_status")) {
        intent = new Intent(MainActivity.this, FOIAStatus.class);
    }

    if (intent != null) {
        bundle.putString("action_object", action);
        intent.putExtras(bundle);
        startActivity(intent);
    }
}

private void launchSubMenu(MobileMenu menu, String title)
{
    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this,"Main Menu",title,"","");

    if (title.equalsIgnoreCase("How to Submit a FOIA Request")) {
        Intent howtoIntent = new Intent(MainActivity.this,
FOIAHowToActivity.class);
        Bundle bundle = new Bundle();

        bundle.putInt("menu_id", (int) menu.getId());
        howtoIntent.putExtras(bundle);
    }
}

```

```

        startActivity(howtoIntent);
    }
}

private void launchAction(MobileMenu menu)
{
    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this,"Main Menu",menu.getAction(),"","");

    //Submission starts with a privacy policy
    if (menu.getAction().equalsIgnoreCase("no_action") &&
        menu.getActionObject().equals("privacy_policy")) {
        Intent intent = new Intent(MainActivity.this,
PrivacyStatementActivity.class);

        startActivity(intent);

    }
}

final class GestureListener extends
GestureDetector.SimpleOnGestureListener {

    private static final int SWIPE_THRESHOLD = 100;
    private static final int SWIPE_VELOCITY_THRESHOLD = 100;

    @Override
    public boolean onDown(MotionEvent e) {
        return true;
    }

    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float
velocityX, float velocityY) {
        boolean result = false;
        try {
            float diffY = e2.getY() - e1.getY();
            float diffX = e2.getX() - e1.getX();
            if (Math.abs(diffX) > Math.abs(diffY)) {
                if (Math.abs(diffX) > SWIPE_THRESHOLD &&
Math.abs(velocityX) > SWIPE_VELOCITY_THRESHOLD) {
                    if (diffX > 0) {
                        onSwipeRight();
                    } else {
                        onSwipeLeft();
                    }
                }
            }
        }
    }
}

```

```

        result = true;
    } else if (Math.abs(diffY) > SWIPE_THRESHOLD &&
Math.abs(velocityY) > SWIPE_VELOCITY_THRESHOLD) {
        if (diffY > 0) {
            onSwipeBottom();
        } else {
            onSwipeTop();
        }
    }
    result = true;
} catch (Exception exception) {
    exception.printStackTrace();
}
return result;
}

public void onSwipeRight() {
    currentIndex--;

    if (currentIndex-- < 0) {
        currentIndex = advertisementList.size() - 1;
    }

    Advertisement ad =
advertisementList.get(currentIndex--);

    String imageUrl =

ad.getImage().getCRIImage().getImageForDensity(CRIImage.IMAGE_TYPE.FULLS
CREEN,
                (Activity) mContext);

    try
    {
        layout.setBackground(new
GetBackgroundImageTask().execute(imageUrl).get());
    }
    catch (Exception e)
    {

    }

}

public void onSwipeLeft() {

```



```

        currentImageIndex++;

        if (currentImageIndex == advertisementList.size()) {
            currentImageIndex = 0;
        }

        Advertisement ad =
advertisementList.get(currentImageIndex);

        String imageUrl =

ad.getImage().getCRIImage().getImageForDensity(CRIImage.IMAGE_TYPE.FULLS
CREEN,
                                                    (Activity) mContext);

        try
        {
            layout.setBackground(new
GetBackgroundImageTask().execute(imageUrl).get());
        }
        catch (Exception e)
        {

        }
    }

    public void onSwipeTop() {

    }

    public void onSwipeBottom() {

    }

}

}
package gov.dhs.mobile.efoia;

import android.os.Bundle;
import com.crashlytics.android.Crashlytics;
import XXXXXXXXXXXXXXXX.ui.activities.CRSplashActivity;
import gov.dhs.mobile.efoia.BuildConfig;
import gov.dhs.mobile.efoia.R;

import io.fabric.sdk.android.Fabric;

```

```

/**
 * *****
 * | EventsFargo
 * | *****
 * | Created byXXXXXXXXXXon 5/5/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */
public class HeroActivity extends CRSplashActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Fabric.with(this, new Crashlytics());
    }
}

/**
    protected String getGCMKey() {
        return BuildConfig.GCM_KEY;
    }
*/
    protected String getServerUrl() {
        return BuildConfig.CR_URL;
    }

    @Override
    protected Class<?> getMainClassActivity() {
        return MainActivity.class;
    }

    protected int getHeroScreenResource() {
        return R.drawable.hero;
    }
}

```

```
}

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;

import gov.dhs.mobile.foia.MainActivity;
import gov.dhs.mobile.foia.R;
import gov.dhs.mobile.foia.models.EventCalendarItem;
import gov.dhs.mobile.foia.viewcontrollers.events.EventsActivity;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

/*****
| DHS e-FOIA
| ****
| File : EventCalendarTableArrayAdapter.java
| Created on 3/17/15
| ****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
| *****/

public class EventCalendarTableArrayAdapter extends
ArrayAdapter<EventCalendarItem>
{
```

```

    HashMap<EventCalendarItem, Integer> mIdMap = new
HashMap<EventCalendarItem, Integer>();
    private Context context;

    private Typeface bold;
    private Typeface light;
    private Context mContext;

    public EventCalendarTableArrayAdapter(Context context, int
resourceId,
                                           List<EventCalendarItem>
objects)
    {
        super(context, resourceId, objects);

        light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
        bold =
Typeface.createFromAsset(context.getAssets(),"nexa_bold.otf");

        this.context = context;
        for (int i = 0; i < objects.size(); ++i) {
            mIdMap.put(objects.get(i), i);
        }
    }

    @Override
    public long getItemId(int position) {
        EventCalendarItem item = getItem(position);
        return mIdMap.get(item);
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        // Get the data item for this position
        EventCalendarItem item = getItem(position);
        // Check if an existing view is being reused, otherwise
inflate the view
        if (convertView == null) {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.event_list_item,
parent, false);
        }
    }

```

```

        TextView titleTextView =
(TextView)convertView.findViewById(R.id.calendar_event_title);
        TextView dateTextView =
(TextView)convertView.findViewById(R.id.calendar_event_date);

        ImageView eventBg =
(ImageView)convertView.findViewById(R.id.event_bg);

        if (EventsActivity.mSelectedItem == position) {
            eventBg.setVisibility(View.VISIBLE);
        } else {
            eventBg.setVisibility(View.INVISIBLE);
        }

        convertView.setContentDescription("View "+item.getTitle());

        titleTextView.setTypeface(bold);
        dateTextView.setTypeface(bold);

        titleTextView.setText(item.getTitle());

        SimpleDateFormat dt = new SimpleDateFormat("MM/dd/yy HH:mm:ss
a");
        SimpleDateFormat dt1 = new SimpleDateFormat("MMMM dd, yyyy");
        try {
            Date date = dt.parse(item.getEvent_date());

            dateTextView.setText(dt1.format(date));

        } catch (ParseException e) {
            e.printStackTrace();
        }

        return convertView;
    }
}

package gov.dhs.mobile.foia.arrayadapters;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;
import android.widget.AdapterView;
import java.util.HashMap;
import java.util.List;
import android.content.Context;
import android.widget.ImageView;

import com.androidquery.AQuery;
import XXXXXXXXXXXX.models.CRImage;
import XXXXXXXXXXXX.models.MobileMenu;

import gov.dhs.mobile.foia.MainActivity;
import gov.dhs.mobile.foia.R;
import gov.dhs.mobile.foia.viewcontrollers.howto.FOIAHowToActivity;

/**
 * Created by XXXXXXXXXXXX on 5/5/15.
 */
public class HomeViewTableArrayAdapter extends
ArrayAdapter<MobileMenu>
{
    HashMap<MobileMenu, Integer> mIdMap = new HashMap<MobileMenu,
Integer>();
    private Context context;
    public HomeViewTableArrayAdapter(Context context, int resourceId,
List<MobileMenu> objects)
    {
        super(context, resourceId, objects);
        this.context = context;
        for (int i = 0; i < objects.size(); ++i) {
            mIdMap.put(objects.get(i), i);
        }
    }

    @Override
    public long getItemId(int position) {
        MobileMenu item = getItem(position);
        return mIdMap.get(item);
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        // Get the data item for this position

```

```

        MobileMenu menu = getItem(position);
        // Check if an existing view is being reused, otherwise
inflate the view
        if (convertView == null) {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.home_table_list_ite
m,
                                                    parent, false);
        }
        // Lookup view for data population
        ImageView highlightView = (ImageView)
convertView.findViewById(R.id.cell_bg);

        if (MainActivity.mSelectedItem == position) {
            highlightView.setVisibility(View.VISIBLE);
        } else {
            highlightView.setVisibility(View.INVISIBLE);
        }

        ImageView imageView = (ImageView)
convertView.findViewById(R.id.list_item_imageview);
        imageView.setContentDescription("View "+menu.getTitle());
        String imageUrl =

menu.getImage().getCRImage().getImageForDensity(CRImage.IMAGE_TYPE.FUL
LSCREEN,
                                                    (Activity)context);

        AQuery aq = new AQuery(context);
        Bitmap preset = aq.getCachedImage(imageUrl);

        if (preset != null) {
            aq.id(imageView).image(imageUrl, true, true, 0, 0, preset,
AQuery.FADE_IN);
        } else {
            aq.id(imageView).image(imageUrl, true, true, 0, 0);
        }

        //if (position == mSelectedItem) {
        //    // set your color
        //}

        return convertView;
    }
}

package gov.dhs.mobile.efoia.arrayadapters;

```

```

import android.app.Activity;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ImageView;
import android.widget.TextView;

import com.androidquery.AQuery;
import XXXXXXXXXXXX.models.CRImage;
import XXXXXXXXXXXX.models.MobileMenu;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.viewcontrollers.howto.FOIAHowToActivity;

import java.util.HashMap;
import java.util.List;

/**
 * Created by XXXXXXXXXXXX on 5/5/15.
 */
public class HowToViewTableArrayAdapter extends
ArrayAdapter<MobileMenu>
{
    HashMap<MobileMenu, Integer> mIdMap = new HashMap<MobileMenu,
Integer>();
    private List<MobileMenu> menus;
    private Context context;
    public HowToViewTableArrayAdapter(Context context, int resourceId,
List<MobileMenu> objects)
    {
        super(context, resourceId, objects);
        menus = objects;
        this.context = context;
        for (int i = 0; i < objects.size(); ++i) {
            mIdMap.put(objects.get(i), i);
        }
    }

    @Override
    public long getItemId(int position) {
        MobileMenu item = getItem(position);
        return mIdMap.get(item);
    }

    @Override

```



```

public boolean hasStableIds() {
    return true;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent) {
    // Get the data item for this position
    MobileMenu menu = getItem(position);

    // Check if an existing view is being reused, otherwise
    inflate the view
    if (convertView == null) {
        convertView =
LayoutInflater.from(getContext()).inflate(R.layout.how_to_list_item,
parent, false);
    }
    if (FOIAHowToActivity.mSelectedItem == position) {
        convertView.setBackgroundColor(new Color().rgb(64,
185, 229));
    } else {
        convertView.setBackgroundColor(Color.TRANSPARENT);
    }

    // Lookup view for data population
    ImageView imageView = (ImageView)
convertView.findViewById(R.id.how_to_imageview);
    TextView textView = (TextView)
convertView.findViewById(R.id.how_to_title);

    textView.setContentDescription("View "+menu.getTitle());

    Typeface light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");

    textView.setTypeface(light);

    textView.setText(menu.getTitle());

    String imageUrl =
menu.getImage().getCRImage().getImageForDensity(CRImage.IMAGE_TYPE.FUL
LSCREEN,
(Activity)context);

    AQuery aq = new AQuery(context);
    Bitmap preset = aq.getCachedImage(imageUrl);

```

```
        if (preset != null) {
            aq.id(imageView).image(imageUrl, true, true, 0, 0, preset,
AQuery.FADE_IN);
        } else {
            aq.id(imageView).image(imageUrl, true, true, 0, 0);
        }

        //if (position == mSelectedItem) {
        //    // set your color
        //}

        return convertView;
    }
}
```

```
package gov.dhs.mobile.efoia.arrayadapters;
```

```
import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
```

```
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Component;
```

```
import java.util.List;
```

```
/*
| DHS e-FOIA
| *****
| File :ComponentArrayAdapter.java
| Created on 3/17/15
|
| *****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
*/
```

```

    | from XXXXXXXXXXXX.
    |
    *****/

public class ComponentArrayAdapter extends ArrayAdapter<Component>
{
    List<Component> mIdMap;

    private Context context;
    LayoutInflater inflater;

    public ComponentArrayAdapter(Context context, int resourceId,
                                List<Component> objects)
    {
        super(context, resourceId, objects);

        this.context = context;

        mIdMap = objects;

        inflater =
(LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVI
CE);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {

        View row = inflater.inflate(R.layout.spinner_list_item,
parent, false);

        TextView make = (TextView)
row.findViewById(R.id.spinner_item);

        Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");

        make.setTypeface(myTypeFace);
        make.setText(mIdMap.get(position).getTitle());
        return row;
    }

    public View getDropDownView(int position, View convertView,
ViewGroup parent) {

        View row =
inflater.from(getContext()).inflate(R.layout.spinner_list_item,
parent, false);

```

```
        TextView make = (TextView)
row.findViewById(R.id.spinner_item);

        Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
        "nexa_bold.otf");

make.setTextColor(context.getResources().getColor(R.color.black_overla
y));
        make.setTypeface(myTypeFace);
        make.setText(mIdMap.get(position).getTitle());
        return row;
    }
}
```

```
}

package gov.dhs.mobile.efoia.arrayadapters;
```

```
import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.TextView;
```

```
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Country;
```

```
import java.util.List;
```

```
/*
| DHS e-FOIA
| *****
| File :CountryArrayAdapter.java
| Created on 3/17/15
|
| *****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
```

```
| is strictly forbidden unless prior written permission is obtained  
| from XXXXXXXXXXXX.  
|
```

```
*****/
```

```
public class CountryArrayAdapter extends ArrayAdapter<Country>  
{  
    List<Country> mIdMap;  
  
    private Context context;  
  
    public CountryArrayAdapter(Context context, int resourceId,  
                               List<Country> objects)  
    {  
        super(context, resourceId, objects);  
  
        this.context = context;  
        mIdMap = objects;  
    }  
  
    @Override  
    public long getItemId(int position) {  
        return position;  
    }  
  
    @Override  
    public boolean hasStableIds() {  
        return true;  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup  
parent) {  
  
        View row =  
LayoutInflater.from(getContext()).inflate(R.layout.spinner_list_item,  
parent, false);  
  
        TextView make = (TextView)  
row.findViewById(R.id.spinner_item);  
  
        Typeface myTypeFace =  
Typeface.createFromAsset(context.getAssets(),  
"nexa_light.otf");  
  
        make.setTextSize(14);  
        make.setTypeface(myTypeFace);  
        make.setText(mIdMap.get(position).getTitle());  
        return row;  
    }  
}
```

```

    }

    public View getDropDownView(int position, View convertView,
    ViewGroup parent) {

        View row =
        LayoutInflater.from(getContext()).inflate(R.layout.spinner_list_item,
            parent, false);

        TextView make = (TextView)
        row.findViewById(R.id.spinner_item);
        make.setTextSize(14);

        Typeface myTypeFace =
        Typeface.createFromAsset(context.getAssets(),
            "nexa_bold.otf");

        make.setTextColor(context.getResources().getColor(R.color.black_overla
        y));
        make.setTypeface(myTypeFace);
        make.setText(mIdMap.get(position).getTitle());
        return row;
    }
}

```

```

}

```

```

package gov.dhs.mobile.efoia.arrayadapters;

```

```

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.TextView;

```

```

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Exemption;

```

```

import java.util.List;

```

```

/*****

```

```

| DHS e-FOIA

```

```

| ****

```

```

| File :ExemptionArrayAdapter.java

```

```

| Created on 3/17/15

```

```

|

```

```

****

```

```

| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.

```

| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.

*****/

```
public class ExemptionArrayAdapter extends ArrayAdapter<Exemption>
{
    List<Exemption> mIdMap;

    private Context context;
    private Typeface light;
    private Typeface bold;
    private List<Exemption> data;

    public ExemptionArrayAdapter(Context context, int resourceId,
                                List<Exemption> objects)
    {
        super(context, resourceId, objects);

        light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
        bold = Typeface.createFromAsset(context.getAssets(),
"nexa_bold.otf");

        this.context = context;
        mIdMap = objects;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
```

```

parent)
    {
        final ViewHolder viewHolder;

        if (convertView == null)
        {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.privacy_act_list_it
em,
            parent, false);

            viewHolder = new ViewHolder();
            viewHolder.position = position;

            convertView.setTag(viewHolder);
        }
        else
        {
            viewHolder = (ViewHolder) convertView.getTag();
        }

        // Get the data item for this position
        Exemption item = mIdMap.get(position);

        viewHolder.actHeader = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_header);
        viewHolder.actHeader.setTypeface(bold);
        viewHolder.actHeader.setText(item.getTitle());

        viewHolder.actText = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_text);
        viewHolder.actText.setTypeface(light);
        viewHolder.actText.setText(item.getDescription());

        return convertView;
    }
    @Override
    public int getCount() {
        return mIdMap.size();
    }

    @Override
    public Exemption getItem(int position) {
        return mIdMap.get(position);
    }

    static class ViewHolder {
        TextView actHeader;
        TextView actText;
    }

```



```

        int position;
    }
}

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.FeeStructure;

import java.util.List;

/*****
| DHS e-FOIA
| ****
| File :FeeStructureArrayAdapter.java
| Created on 3/17/15
|
| ****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|
| *****/

public class FeeStructureArrayAdapter extends
ArrayAdapter<FeeStructure>
{
    List<FeeStructure> mIdMap;

    private Context context;
    private Typeface light;
    private Typeface bold;

```

```

private List<FeeStructure> data;

public FeeStructureArrayAdapter(Context context, int resourceId,
                                List<FeeStructure> objects)
{
    super(context, resourceId, objects);

    light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
    bold = Typeface.createFromAsset(context.getAssets(),
"nexa_bold.otf");

    this.context = context;
    mIdMap = objects;
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public boolean hasStableIds() {
    return true;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent)
{
    final ViewHolder viewHolder;

    if (convertView == null)
    {
        convertView =
LayoutInflater.from(getContext()).inflate(R.layout.privacy_act_list_it
em,
                                parent, false);

        viewHolder = new ViewHolder();
        viewHolder.position = position;

        convertView.setTag(viewHolder);
    }
    else
    {
        viewHolder = (ViewHolder) convertView.getTag();
    }
}

```

```

        // Get the data item for this position
        FeeStructure item = mIdMap.get(position);

        viewHolder.actHeader = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_header);
        viewHolder.actHeader.setTypeface(bold);
        viewHolder.actHeader.setText(item.getTitle());

        viewHolder.actText = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_text);
        viewHolder.actText.setTypeface(light);
        viewHolder.actText.setText(item.getFee_note());

        return convertView;
    }
    @Override
    public int getCount() {
        return mIdMap.size();
    }

    @Override
    public FeeStructure getItem(int position) {
        return mIdMap.get(position);
    }

    static class ViewHolder {
        TextView actHeader;
        TextView actText;
        int position;
    }
}

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;

import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;

import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.FiveStepsItem;

import java.util.List;

```

```

/*****
| DHS e-FOIA
| ****
| File : EventCalendarTableArrayAdapter.java
| Created on 3/17/15
|
| ****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|
| *****/

```

```

public class FiveStepsTableArrayAdapter extends
ArrayAdapter<FiveStepsItem>
{
    List<FiveStepsItem> mIdMap;

    private Context context;
    private Typeface bold;
    private Typeface light;
    int layoutResourceId;
    private List<FiveStepsItem> data;

    public FiveStepsTableArrayAdapter(Context context, int resourceId,
List<FiveStepsItem> objects)
    {
        super(context, resourceId, objects);

        light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
        bold =
Typeface.createFromAsset(context.getAssets(),"nexa_bold.otf");

        this.context = context;
        mIdMap = objects;
    }

    @Override

```

```

public long getItemId(int position) {
    return position;
}

@Override
public boolean hasStableIds() {
    return true;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent)
{
    final ViewHolder viewHolder;

    if (convertView == null)
    {
        convertView =
LayoutInflater.from(getContext()).inflate(R.layout.five_steps_list_ite
m,
        parent, false);

        viewHolder = new ViewHolder();
        viewHolder.position = position;

        convertView.setTag(viewHolder);
    }
    else
    {
        viewHolder = (ViewHolder) convertView.getTag();
    }

    // Get the data item for this position
    FiveStepsItem item = mMap.get(position);

    viewHolder.stepNumber = (TextView)
convertView.findViewById(R.id.five_steps_line_item_step);
    viewHolder.stepText =
(TextView)convertView.findViewById(R.id.five_steps_line_item_text);

    viewHolder.stepNumber.setTypeface(bold);
    viewHolder.stepText.setTypeface(light);

    viewHolder.stepNumber.setText(""+ (position+1));
    viewHolder.stepText.setText(item.getFoia_step_text());

    return convertView;
}
@Override
public int getCount() {

```

```

        return mMap.size();
    }

    @Override
    public FiveStepsItem getItem(int position) {
        return mMap.get(position);
    }

    static class ViewHolder {
        TextView stepNumber;
        TextView stepText;
        int position;
    }
}

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Limits;

import java.util.List;

/*****
| DHS e-FOIA
| ****
| File :LimitsArrayAdapter.java
| Created on 3/17/15
|
| ****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained

```

```

| from XXXXXXXXXXXX.
|
|*****/

public class LimitsArrayAdapter extends ArrayAdapter<Limits>
{
    List<Limits> mIdMap;

    private Context context;
    private Typeface light;
    private Typeface bold;
    private List<Limits> data;

    public LimitsArrayAdapter(Context context, int resourceId,
                               List<Limits> objects)
    {
        super(context, resourceId, objects);

        light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
        bold = Typeface.createFromAsset(context.getAssets(),
"nexa_bold.otf");

        this.context = context;
        mIdMap = objects;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent)
    {
        final ViewHolder viewHolder;

        if (convertView == null)
        {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.privacy_act_list_it
em,
                parent, false);

```

```

        viewHolder = new ViewHolder();
        viewHolder.position = position;

        convertView.setTag(viewHolder);
    }
    else
    {
        viewHolder = (ViewHolder) convertView.getTag();
    }

    // Get the data item for this position
    Limits item = mMap.get(position);

    viewHolder.actHeader = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_header);
viewHolder.actHeader.setTypeface(bold);
viewHolder.actHeader.setText(item.getTitle());

    viewHolder.actText = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_text);
viewHolder.actText.setTypeface(light);
viewHolder.actText.setText(item.getParagraph());

    return convertView;
}
@Override
public int getCount() {
    return mMap.size();
}

@Override
public Limits getItem(int position) {
    return mMap.get(position);
}

static class ViewHolder {
    TextView actHeader;
    TextView actText;
    int position;
}
}

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;

```



```

import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.PrivacyAct;

import java.util.List;

/*****
| DHS e-FOIA
| *****/
| File :PrivacyActArrayAdapter.java
| Created on 3/17/15
|
*****/
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|
*****/

public class PrivacyActArrayAdapter extends ArrayAdapter<PrivacyAct>
{
    List<PrivacyAct> mIdMap;

    private Context context;
    private Typeface light;
    private Typeface bold;
    private List<PrivacyAct> data;

    public PrivacyActArrayAdapter(Context context, int resourceId,
                                  List<PrivacyAct> objects)
    {
        super(context, resourceId, objects);

        light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
        bold = Typeface.createFromAsset(context.getAssets(),

```

```

"nexa_bold.otf");

        this.context = context;
        mIdMap = objects;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent)
    {
        final ViewHolder viewHolder;

        if (convertView == null)
        {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.privacy_act_list_it
em,
                parent, false);

            viewHolder = new ViewHolder();
            viewHolder.position = position;

            convertView.setTag(viewHolder);
        }
        else
        {
            viewHolder = (ViewHolder) convertView.getTag();
        }

        // Get the data item for this position
        PrivacyAct item = mIdMap.get(position);

        viewHolder.actHeader = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_header);
        viewHolder.actHeader.setTypeface(bold);

        if (!item.getTitle().contains("preamble"))
        {
            viewHolder.actHeader.setText(item.getTitle());
        }
    }

```

```

        else
        {
            viewHolder.actHeader.setText("");
        }

        viewHolder.actText = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_text);
        viewHolder.actText.setTypeface(light);
        viewHolder.actText.setText(item.getRequest_step());

        return convertView;
    }
    @Override
    public int getCount() {
        return mIdMap.size();
    }

    @Override
    public PrivacyAct getItem(int position) {
        return mIdMap.get(position);
    }

    static class ViewHolder {
        TextView actHeader;
        TextView actText;
        int position;
    }
}

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Processing;

import java.util.List;

/*****
| DHS e-FOIA
| *****/

```

```

| File :ProcessingArrayAdapter.java
| Created on 3/17/15
|
|*****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|*****/

public class ProcessingArrayAdapter extends ArrayAdapter<Processing>
{
    List<Processing> mIdMap;

    private Context context;
    private Typeface light;
    private Typeface bold;
    private List<Processing> data;

    public ProcessingArrayAdapter(Context context, int resourceId,
                                  List<Processing> objects)
    {
        super(context, resourceId, objects);

        light = Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");
        bold = Typeface.createFromAsset(context.getAssets(),
"nexa_bold.otf");

        this.context = context;
        mIdMap = objects;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public boolean hasStableIds() {

```

```

        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent)
    {
        final ViewHolder viewHolder;

        if (convertView == null)
        {
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.privacy_act_list_it
em,
                parent, false);

            viewHolder = new ViewHolder();
            viewHolder.position = position;

            convertView.setTag(viewHolder);
        }
        else
        {
            viewHolder = (ViewHolder) convertView.getTag();
        }

        // Get the data item for this position
        Processing item = mMap.get(position);

        viewHolder.actHeader = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_header);
        viewHolder.actHeader.setTypeface(bold);
        viewHolder.actHeader.setText(item.getTitle());

        viewHolder.actText = (TextView)
convertView.findViewById(R.id.privacy_act_list_item_text);
        viewHolder.actText.setTypeface(light);
        viewHolder.actText.setText(item.getProcessingText());

        return convertView;
    }

    @Override
    public int getCount() {
        return mMap.size();
    }

    @Override
    public Processing getItem(int position) {
        return mMap.get(position);
    }
}

```

```

        static class ViewHolder {
            TextView actHeader;
            TextView actText;
            int position;
        }
    }

package gov.dhs.mobile.efoia.arrayadapters;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.State;

import java.util.List;

/*****
 | DHS e-FOIA
 | ****
 | File :StateArrayAdapter.java
 | Created on 3/17/15
 |
 ****
 | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 |
 | NOTICE: All information contained herein is, and remains
 | the property of XXXXXXXXXXXX and its suppliers,
 | if any. The intellectual and technical concepts contained
 | herein are proprietary to XXXXXXXXXXXX
 | and its suppliers and may be covered by U.S. and Foreign Patents,
 | patents in process, and are protected by trade secret or copyright
 | law.
 | Dissemination of this information or reproduction of this material
 | is strictly forbidden unless prior written permission is obtained
 | from XXXXXXXXXXXX.
 |
 *****/

public class StateArrayAdapter extends ArrayAdapter<State>
{
    List<State> mIdMap;

```

```

private Context context;

public StateArrayAdapter(Context context, int resourceId,
                          List<State> objects)
{
    super(context, resourceId, objects);

    this.context = context;
    mIdMap = objects;
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public boolean hasStableIds() {
    return true;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent) {

    View row =
LayoutInflater.from(getContext()).inflate(R.layout.spinner_list_item,
parent, false);

    TextView make = (TextView)
row.findViewById(R.id.spinner_item);
    make.setTextSize(14);
    Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");

    make.setTypeface(myTypeFace);
    make.setText(mIdMap.get(position).getTitle());
    return row;
}

public View getDropDownView(int position, View convertView,
ViewGroup parent) {

    View row =
LayoutInflater.from(getContext()).inflate(R.layout.spinner_list_item,
parent, false);

```

```
        TextView make = (TextView)
row.findViewById(R.id.spinner_item);
        make.setTextSize(14);
        Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
                "nexa_bold.otf");

make.setTextColor(context.getResources().getColor(R.color.black_overla
y));
        make.setTypeface(myTypeFace);
        make.setText(mIdMap.get(position).getTitle());
        return row;
    }

}
```

```
package gov.dhs.mobile.efoia.arrayadapters;
```

```
import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.TextView;
```

```
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Suffix;
```

```
import java.util.List;
```

```
/*
| DHS e-FOIA
| *****
| File :SuffixArrayAdapter.java
| Created on 3/17/15
|
| *****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
*/
```



```

| from XXXXXXXXXXXX.
|
*****/

public class SuffixArrayAdapter extends ArrayAdapter<Suffix>
{
    List<Suffix> mIdMap;

    private Context context;

    public SuffixArrayAdapter(Context context, int resourceId,
                               List<Suffix> objects)
    {
        super(context, resourceId, objects);

        this.context = context;
        mIdMap = objects;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {

        View row =
LayoutInflater.from(getContext()).inflate(R.layout.spinner_list_item,
parent, false);

        TextView make = (TextView)
row.findViewById(R.id.spinner_item);
make.setTextSize(14);
Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
"nexa_light.otf");

        make.setTypeface(myTypeFace);
make.setText(mIdMap.get(position).getTitle());
return row;
    }
}

```


| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.

| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.

*****/

```
public class TitleArrayAdapter extends ArrayAdapter<Title>
{
    List<Title> mIdMap;

    private Context context;
    LayoutInflater inflater;

    public TitleArrayAdapter(Context context, int resourceId,
                             List<Title> objects)
    {
        super(context, resourceId, objects);

        this.context = context;
        mIdMap = objects;

        inflater =
(LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVI
CE);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public boolean hasStableIds() {
        return true;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {

        View row = inflater.inflate(R.layout.spinner_list_item,
parent, false);

        TextView make = (TextView)
row.findViewById(R.id.spinner_item);
```

```

        make.setTextSize(14);
        Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
        "nexa_light.otf");

        make.setTypeface(myTypeFace);
        make.setText(mIdMap.get(position).getTitle());
        return row;
    }

    public View getDropDownView(int position, View convertView,
ViewGroup parent) {

        View row = inflater.inflate(R.layout.spinner_list_item,
        parent, false);

        TextView make = (TextView)
row.findViewById(R.id.spinner_item);
        make.setTextSize(14);
        Typeface myTypeFace =
Typeface.createFromAsset(context.getAssets(),
        "nexa_bold.otf");

make.setTextColor(context.getResources().getColor(R.color.black_overla
y));
        make.setTypeface(myTypeFace);
        make.setText(mIdMap.get(position).getTitle());
        return row;
    }

}

package gov.dhs.mobile.foia.viewcontrollers.submission.step5;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.InputType;
import android.text.Spannable;
import android.text.SpannableStringBuilder;
import android.text.style.ForegroundColorSpan;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;

```

```

import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.datastructures.Submission;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step6.SubmissionStep6A
ctivity;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */
public class SubmissionStep5Activity extends Activity {

    Submission submission;
    EditText feeAmount;
    Button nextButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_submission_step5);

        Typeface bold = Typeface.createFromAsset(getAssets(),
"nexa_bold.otf");
        Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

```

```

Bundle bundle = getIntent().getExtras();
submission = bundle.getParcelable("submission");

nextButton = (Button)findViewById(R.id.nextButton);
TextView title = (TextView)findViewById(R.id.feetitle);
TextView desc = (TextView)findViewById(R.id.feeDesc);
TextView feeText = (TextView)findViewById(R.id.feetext);
    feeAmount = (EditText)findViewById(R.id.feeAmount);
TextView pageTitle =
(TextView)findViewById(R.id.step_5_title);

    title.setTypeface(light);
    desc.setTypeface(light);
    feeText.setTypeface(bold);
    pageTitle.setTypeface(bold);
    feeAmount.setTypeface(light);

attributeText(feeText);

feeAmount.setRawInputType(InputType.TYPE_CLASS_TEXT);
feeAmount.setImeOptions(EditorInfo.IME_ACTION_GO);

feeAmount.setOnEditorActionListener(new
TextView.OnEditorActionListener() {

    @Override
    public boolean onEditorAction(TextView v, int actionId,
                                KeyEvent event) {
        if (event != null && (event.getKeyCode() ==
KeyEvent.KEYCODE_ENTER)) {
            InputMethodManager in = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);

            feeAmount.clearFocus();
            if (validateEntries()) {
                nextButton.setEnabled(true);
            } else {
                nextButton.setEnabled(false);
            }

            in.hideSoftInputFromWindow(feeAmount
                                .getApplicationWindowToken(),
                                InputMethodManager.HIDE_NOT_ALWAYS);
            // Must return true here to consume event
            return true;
        }
        return false;
    }
}

```

```

        });

    }

    public void goBack(View v)
    {
        this.finish();
    }

    public void next(View v)
    {
        if (feeAmount.getText().length() >1) {
            submission.setPayAmount(feeAmount.getText().toString());

            Intent intent = new Intent(SubmissionStep5Activity.this,
SubmissionStep6Activity.class);

            Bundle bundle = new Bundle();
            bundle.putParcelable("submission", submission);

            intent.putExtras(bundle);
            startActivity(intent);
        }
    }

    private void attributeText(Textview view)
    {
        String labelString = view.getText().toString();

        int asteriskPosition = labelString.indexOf("*");

        final SpannableStringBuilder sb = new
SpannableStringBuilder(labelString);

        final ForegroundColorSpan redStyle = new
ForegroundColorSpan(Color.WHITE);

        sb.setSpan(redStyle, asteriskPosition, asteriskPosition+1,
Spannable.SPAN_INCLUSIVE_INCLUSIVE);

        view.setText(sb);
    }
}

```

```

private boolean validateEntries() {
    if (feeAmount.getText().toString().length() > 1 )
    {
        submission.setInitials(feeAmount.getText().toString());
        submission.setDeclaration(true);
        return true;
    }

    return false;
}

}

package gov.dhs.mobile.efoia.viewcontrollers.submission.step3;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.InputType;
import android.text.Spannable;
import android.text.SpannableStringBuilder;
import android.text.style.ForegroundColorSpan;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.ToggleButton;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.datastructures.Submission;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step4.SubmissionStep4A
ctivity;

/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |

```



```

*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */
public class SubmissionStep3Activity extends Activity {

    private Submission submission;
    private ToggleButton recordsYes;
    private ToggleButton recordsNo;
    private ToggleButton perjury;
    private EditText initials;
    private Button nextButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_submission_step3);

        Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");
        Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

        Bundle bundle = getIntent().getExtras();
        submission = bundle.getParcelable("submission");

        recordsYes = (ToggleButton)findViewById(R.id.yesToggle);
        recordsNo= (ToggleButton)findViewById(R.id.noToggle);
        perjury= (ToggleButton)findViewById(R.id.perjuryToggle);
        initials= (EditText)findViewById(R.id.initials);
        nextButton= (Button)findViewById(R.id.next_button);

```

```

    TextView reqLbl = (TextView)findViewById(R.id.persRecsTitle);
    TextView reqtxt = (TextView)findViewById(R.id.persRecsTxt);
    TextView selfTxt = (TextView)findViewById(R.id.selfRecords);
    TextView yesLbl = (TextView)findViewById(R.id.yesLbl);
    TextView noLbl = (TextView)findViewById(R.id.noLbl);
    TextView perStat = (TextView)findViewById(R.id.perjuryTxt);
    TextView perDesc = (TextView)findViewById(R.id.perjuryDesc);
    TextView initsTxt = (TextView)findViewById(R.id.initialsText);
    TextView step3Title =
(TextView)findViewById(R.id.step_3_title);

```

```

    selfTxt.setTypeface(bold);
    initsTxt.setTypeface(bold);
    step3Title.setTypeface(bold);
    perStat.setTypeface(bold);

```

```

    reqLbl.setTypeface(light);
    reqtxt.setTypeface(light);
    perDesc.setTypeface(light);
    initials.setTypeface(light);
    yesLbl.setTypeface(light);
    noLbl.setTypeface(light);

```

```

attributeText(selfTxt);

```

```

submission.setRecordsForSelf(true);

```

```

initials.setRawInputType(InputType.TYPE_CLASS_TEXT);
initials.setImeOptions(EditorInfo.IME_ACTION_GO);

```

```

initials.setOnEditorActionListener(new
TextView.OnEditorActionListener() {

    @Override
    public boolean onEditorAction(TextView v, int actionId,
                                   KeyEvent event) {
        if (event != null && (event.getKeyCode() ==
KeyEvent.KEYCODE_ENTER)) {
            InputMethodManager in = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);

            // NOTE: In the author's example, he uses an
identifier
            // called searchBar. If setting this code on your
EditText
            // then use v.getWindowToken() as a reference to
your
            // EditText is passed into this callback as a

```

TextView

```
        if (validateEntries())
        {
            nextButton.setEnabled(true);
        }

        in.hideSoftInputFromWindow(initials
                                    .getApplicationWindowToken(),
                                    InputMethodManager.HIDE_NOT_ALWAYS);
        // Must return true here to consume event
        return true;
    }
    return false;
}
});

}

private void attributeText(TextView view)
{
    String labelString = view.getText().toString();

    int asteriskPosition = labelString.indexOf("*");

    final SpannableStringBuilder sb = new
SpannableStringBuilder(labelString);

    final ForegroundColorSpan redStyle = new
ForegroundColorSpan(Color.WHITE);

    sb.setSpan(redStyle, asteriskPosition, asteriskPosition+1,
Spannable.SPAN_INCLUSIVE_INCLUSIVE);

    view.setText(sb);

}

public void setSubmissionType(View v)
{
    ToggleButton tgl = (ToggleButton)v;

    if (tgl.getId() == recordsNo.getId())
    {
        initials.setEnabled(false);
        initials.clearFocus();
    }
}
```

```

        perjury.setEnabled(false);
        nextButton.setEnabled(true);
        recordsYes.setChecked(false);
        initials.setText("");
        perjury.setChecked(false);
    }
    else
    {
        initials.setEnabled(true);
        perjury.setEnabled(true);
        nextButton.setEnabled(false);
        recordsNo.setChecked(false);
    }
}

public void next(View v)
{
    if (validateEntries()) {
        Intent intent =
            new Intent(SubmissionStep3Activity.this,
SubmissionStep4Activity.class);

        Bundle bundle = new Bundle();
        bundle.putParcelable("submission", submission);

        intent.putExtras(bundle);
        startActivity(intent);
    }
}

private boolean validateEntries() {
    if (!recordsYes.isChecked()) {
        nextButton.setEnabled(true);
        submission.setRecordsForSelf(false);
        return true;
    }
    else
    {
        if (initials.getText().toString().length() > 1 &&
perjury.isChecked())
        {
            submission.setInitials(initials.getText().toString());
            submission.setDeclaration(true);
            return true;
        }
    }
}
}

```

```
        return false;
    }

    public void goBack(View v)
    {
        this.finish();
    }
}
```

```
package gov.dhs.mobile.efoia.viewcontrollers.submission.step2;
```

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
```

```
import android.text.Spannable;
import android.text.SpannableStringBuilder;
import android.text.style.ForegroundColorSpan;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.TextView;
```

```
import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.arrayadapters.CountryArrayAdapter;
import gov.dhs.mobile.efoia.arrayadapters.StateArrayAdapter;
import gov.dhs.mobile.efoia.arrayadapters.SuffixArrayAdapter;
import gov.dhs.mobile.efoia.arrayadapters.TitleArrayAdapter;
import gov.dhs.mobile.efoia.datastructures.Submission;
import gov.dhs.mobile.efoia.models.Country;
import gov.dhs.mobile.efoia.models.State;
import gov.dhs.mobile.efoia.models.Suffix;
import gov.dhs.mobile.efoia.models.Title;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step3.SubmissionStep3A
ctivity;
```

```

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *| *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 *| *****
 * | <Insert your Javadoc Here>
 * |
 *| *****
 */
public class SubmissionStep2Activity extends Activity {

    private Submission submission;
    private Spinner titleSpinner;
    private Spinner suffixSpinner;
    private Spinner stateSpinner;
    private Spinner countrySpinner;
    private EditText firstName;
    private EditText middleInitial;
    private EditText lastName;
    private EditText emailAddr;
    private EditText addr1;
    private EditText addr2;
    private EditText city;
    private EditText zip;
    private EditText telephone;
    private EditText fax;
    private Button nextButton;
    private Context mContext = (Context)this;

```

```

private List<Title> titleList;
private List<Suffix> suffixList;
private List<State> stateList;
private List<Country> countryList;

private TaskListener<DynamicContentResponse<Title>>
titleTaskListener =
    new TaskListener<DynamicContentResponse<Title>>() {
        @Override
        public void
onTaskSuccess(DynamicContentResponse<Title> result) {
            if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
            {

                titleList = result.getContent();

                titleList.add(0, new Title(0, "None", "",
null));

                TitleArrayAdapter adapter = new
                    TitleArrayAdapter(mContext,
                        R.layout.spinner_list_item,
titleList);

                titleSpinner.setAdapter(adapter);
                titleSpinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
                    @Override
                    public void onItemSelected(AdapterView<?>
parent, View view, int position, long id) {

                        if (position > 0) {

submission.setTitle(titleList.get(position).getTitle());
                            firstName.requestFocus();
                        } else {
                            submission.setTitle("");
                        }
                    }

                    @Override
                    public void
onNothingSelected(AdapterView<?> parent) {

                        }
                    }
                });
            }
        }
    }

```

```

        @Override
        public void onTaskFailure(Exception e) {

        }

        @Override
        public void onTaskCancelled() {

        }

        @Override
        public void onTaskTimeOut() {

        }
    };

    private TaskListener<DynamicContentResponse<Suffix>>
    suffixTaskListener =
        new TaskListener<DynamicContentResponse<Suffix>>() {
            @Override
            public void
onTaskSuccess(DynamicContentResponse<Suffix> result) {
                if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                {

                    suffixList = result.getContent();

                    suffixList.add(0, new Suffix(0, "None", "",
null));

                    SuffixArrayAdapter adapter = new
                    SuffixArrayAdapter(mContext,
                    R.layout.spinner_list_item,
suffixList);

                    suffixSpinner.setAdapter(adapter);
                    suffixSpinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
                        @Override
                        public void onItemSelected(AdapterView<?>
parent, View view, int position, long id) {
                            if (position > 0) {

                                submission.setSuffix(suffixList.get(position).getTitle());
                                emailAddr.requestFocus();
                            } else {
                                submission.setSuffix("");
                            }
                        }
                    });
                }
            }
        };

```



```

        }

        @Override
        public void
onNothingSelected(AdapterView<?> parent) {

            }
        });
    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

private TaskListener<DynamicContentResponse<State>>
stateTaskListener =
    new TaskListener<DynamicContentResponse<State>>() {
        @Override
        public void
onTaskSuccess(DynamicContentResponse<State> result) {
            if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
            {

                stateList = result.getContent();

                stateList.add(0, new State(0L, "None", "",
null, ""));

                StateArrayAdapter adapter = new
                    StateArrayAdapter(mContext,
                    R.layout.spinner_list_item,
stateList);

                stateSpinner.setAdapter(adapter);

```

```

        stateSpinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
parent, View view, int position, long id) {
                if (position > 0) {

submission.setState(stateList.get(position).getCode());
                zip.requestFocus();
                } else {
                    submission.setState("");
                    nextButton.setEnabled(false);
                }

                if (validateEntries())
                {
                    nextButton.setEnabled(true);
                }
                else
                {
                    nextButton.setEnabled(false);
                }
            }

            @Override
            public void
onNothingSelected(AdapterView<?> parent) {

                }
        });
    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

private TaskListener<DynamicContentResponse<Country>>

```



```

        });

    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_submission_step2);

    Bundle bundle = getIntent().getExtras();
    submission = bundle.getParcelable("submission");

    Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");
    Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

    TextView step1Desc =
(TextView)findViewById(R.id.step_1_desc);
    TextView step1Title =
(TextView)findViewById(R.id.step_1_title);
    TextView titleLabel =
(TextView)findViewById(R.id.title_label);
    TextView firstNameLabel
=(TextView)findViewById(R.id.first_name_label);

    final TextView middleInitLabel
=(TextView)findViewById(R.id.middle_initial_label);

```

```

        titleSpinner = (Spinner)findViewById(R.id.title_spinner);
        titleSpinner.setFocusable(true);
        titleSpinner.setFocusableInTouchMode(true);
        firstName      = (EditText)findViewById(R.id.first_name);
        middleInitial  = (EditText)findViewById(R.id.middle_initial);
        TextView lastNameLabel =
(TextView)findViewById(R.id.last_name_label);
        lastName      = (EditText)findViewById(R.id.last_name);
        TextView suffixLabel =
(TextView)findViewById(R.id.suffix_label);
        suffixSpinner = (Spinner)findViewById(R.id.suffix_spinner);
        suffixSpinner.setFocusable(true);
        suffixSpinner.setFocusableInTouchMode(true);
        TextView emailLabel =
(TextView)findViewById(R.id.email_addr_label);
        emailAddr     = (EditText)findViewById(R.id.email_addr);
        TextView centryLabel =
(TextView)findViewById(R.id.centry_label);
        countrySpinner = (Spinner)findViewById(R.id.centry_spinner);
        countrySpinner.setFocusable(true);
        countrySpinner.setFocusableInTouchMode(true);
        TextView addr1Label =
(TextView)findViewById(R.id.addr_1_label);
        addr1          = (EditText)findViewById(R.id.address_1);
        TextView addr2Label =
(TextView)findViewById(R.id.addr_2_label);
        addr2          = (EditText)findViewById(R.id.addr_2);
        TextView cityLabel = (TextView)findViewById(R.id.city_label);
        city           = (EditText)findViewById(R.id.city);
        TextView stateLabel =
(TextView)findViewById(R.id.state_label);
        stateSpinner = (Spinner)findViewById(R.id.state_spinner);
        stateSpinner.setFocusable(true);
        stateSpinner.setFocusableInTouchMode(true);
        TextView zipLabel = (TextView) findViewById(R.id.zip_label);
        zip           = (EditText) findViewById(R.id.zip);
        TextView phoneLabel =
(TextView)findViewById(R.id.phone_label);
        telephone     = (EditText)findViewById(R.id.phone);
        TextView faxLabel = (TextView)findViewById(R.id.fax_label);
        fax           = (EditText)findViewById(R.id.fax);
        nextButton    = (Button)findViewById(R.id.next_button);

        step1Desc.setTypeface(bold);
        step1Title.setTypeface(bold);
        titleLabel.setTypeface(bold);
        firstNameLabel.setTypeface(bold);
        middleInitLabel.setTypeface(bold);
        lastNameLabel.setTypeface(bold);
        suffixLabel.setTypeface(bold);

```

```
emailLabel.setTypeface(bold);
cntryLabel.setTypeface(bold);
addr1Label.setTypeface(bold);
addr2Label.setTypeface(bold);
cityLabel.setTypeface(bold);
stateLabel.setTypeface(bold);
zipLabel.setTypeface(bold);
phoneLabel.setTypeface(bold);
faxLabel.setTypeface(bold);
```

```
firstName.setTypeface(light);
middleInitial.setTypeface(light);
lastName.setTypeface(light);
emailAddr.setTypeface(light);
addr1.setTypeface(light);
addr2.setTypeface(light);
city.setTypeface(light);
zip.setTypeface(light);
telephone.setTypeface(light);
fax.setTypeface(light);
```

```
attributeText(firstNameLabel);
attributeText(lastNameLabel);
attributeText(cityLabel);
attributeText(stateLabel);
attributeText(emailLabel);
attributeText(zipLabel);
attributeText(cntryLabel);
attributeText(addr1Label);
```

```
//start the tasks
```

```
Type suffixType = new
TypeToken<DynamicContentResponse<Suffix>>().getType();
GetDynamicContent<DynamicContentResponse<Suffix>> suffixTask =
    new
DynamicContentTaskBuilder<DynamicContentResponse<Suffix>>()
    .withContext(this)
    .withMethod(HttpGet.METHOD_NAME)
    .withPermanentLink("suffix")
    .withParam("page","0")
    .withTaskListener(suffixTaskListener)
    .withType(suffixType)
    .build();
suffixTask.execute();
```

```
Type stateType = new
TypeToken<DynamicContentResponse<State>>().getType();
GetDynamicContent<DynamicContentResponse<State>> stateTask =
```

```

        new
DynamicContentTaskBuilder<DynamicContentResponse<State>>()
        .withContext(this)
        .withMethod(HttpGet.METHOD_NAME)
        .withPermanentLink("state")
        .withTaskListener(stateTaskListener)
        .withParam("page", "0")
        .withType(stateType)
        .build();
stateTask.execute();

Type countryType = new
TypeToken<DynamicContentResponse<Country>>().getType();
GetDynamicContent<DynamicContentResponse<Country>> countryTask
=
        new
DynamicContentTaskBuilder<DynamicContentResponse<Country>>()
        .withContext(this)
        .withMethod(HttpGet.METHOD_NAME)
        .withPermanentLink("country_codes")
        .withParam("page", "0")
        .withTaskListener(countryTaskListener)
        .withType(countryType)
        .build();
countryTask.execute();

//start the tasks
Type titleType = new
TypeToken<DynamicContentResponse<Title>>().getType();
GetDynamicContent<DynamicContentResponse<Title>> titleTask =
        new
DynamicContentTaskBuilder<DynamicContentResponse<Title>>()
        .withContext(this)
        .withMethod(HttpGet.METHOD_NAME)
        .withPermanentLink("title")
        .withParam("page", "0")
        .withTaskListener(titleTaskListener)
        .withType(titleType)
        .build();
titleTask.execute();

lastName.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {
            suffixSpinner.requestFocus();
            if (validateEntries()) {
                nextButton.setEnabled(true);
            }
        }
    }
});

```

```

        } else {
            nextButton.setEnabled(false);
        }
        return true;
    }
    return false;
}
});

    firstName.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {
            middleInitial.requestFocus();
            if (validateEntries())
            {
                nextButton.setEnabled(true);
            }
            else
            {
                nextButton.setEnabled(false);
            }
            return true;
        }
        return false;
    }
});

    lastName.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {
            suffixSpinner.requestFocus();
            if (validateEntries())
            {
                nextButton.setEnabled(true);
            }
            else
            {
                nextButton.setEnabled(false);
            }
            return true;
        }
        return false;
    }
});

```



```

        emailAddr.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {
            countrySpinner.requestFocus();
            if (validateEntries())
            {
                nextButton.setEnabled(true);
            }
            else
            {
                nextButton.setEnabled(false);
            }
            return true;
        }
        return false;
    }
});

```

```

        addr1.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {
            addr2.requestFocus();
            if (validateEntries())
            {
                nextButton.setEnabled(true);
            }
            else
            {
                nextButton.setEnabled(false);
            }
            return true;
        }
        return false;
    }
});

```

```

        city.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {

```

```

        stateSpinner.requestFocus();
        if (validateEntries())
        {
            nextButton.setEnabled(true);
        }
        else
        {
            nextButton.setEnabled(false);
        }
        return true;
    }
    return false;
});
});

zip.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView view, int actionID,
KeyEvent event) {
        if (actionID == EditorInfo.IME_ACTION_NEXT) {
            telephone.requestFocus();
            if (validateEntries())
            {
                nextButton.setEnabled(true);
            }
            else
            {
                nextButton.setEnabled(false);
            }
            return true;
        }
        return false;
    }
});
});

}

public void next(View v)
{
    if (validateEntries())
    {
        submission.setAddressLine2(addr2.getText().toString());
        submission.setAddressLine1(addr1.getText().toString());
        submission.setZipCode(zip.getText().toString());

        submission.setEmailAddress(emailAddr.getText().toString());
        submission.setFirstName(firstName.getText().toString());
        submission.setLastName(lastName.getText().toString());
        submission.setCity(city.getText().toString());
    }
}

```

```
submission.setMiddleInitial(middleInitial.getText().toString());
    submission.setTelephone(telephone.getText().toString());
    submission.setFax(fax.getText().toString());
```

```
        Intent intent =
            new Intent(SubmissionStep2Activity.this,
SubmissionStep3Activity.class);
```

```
        Bundle bundle = new Bundle();
        bundle.putParcelable("submission", submission);
```

```
        intent.putExtras(bundle);
        startActivity(intent);
```

```
    }
}
```

```
private boolean validateEntries()
{
```

```
    if ((firstName.getText().toString().length() > 1) &&
        (lastName.getText().toString().length() > 1) &&
        (emailAddr.getText().toString().length() > 1) &&
        validateEmail(emailAddr.getText().toString()) &&
        (submission.getCountry().length() > 1) &&
        (addr1.getText().toString().length() > 1) &&
        (submission.getState().toString().length() > 1) &&
        (zip.getText().toString().length() > 1))
```

```
    {
        return true;
    }
```

```
    else
    {
        return false;
    }
}
```

```
}
```

```
private boolean validateEmail(String candidate)
{
```

```
    String emailRegex =
        "(?:[a-z0-9!#$%&'*/=?\\^_`{|}~-]+(?:\\. [a-z0-9!#$%&'*/=?\\^_`{|}~-]+
\\^_`{|}"+
        "~-]+)*|\"(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b
\\x5d-\\\""+
        "x7f]|\\\\\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])*\" )@(?: (?: [a-
z0-9] (?: [a-\"+
        "z0-9-]*[a-z0-9]) ?\\. )+ [a-z0-9] (?: [a-z0-9-]*[a-z0-9]) ?| \\\\ (?:
(?:25[0-5\"+
        ") | 2[0-4] [0-9] | [01]? [0-9] [0-9] ?) \\. ) {3} (?:25[0-5] | 2[0-4] [0-9] |
```

```
[01]?[0-"]+
"9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\
\\x1f\\x21"+
"-\\x5a\\x53-\\x7f]|\\\\\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f]))+\\
\\)");
```

```
    return candidate.matches(emailRegex);
}
```

```
public void goBack(View v)
{
    this.finish();
}
```

```
private void attributeText(TextView view)
{
    String labelString = view.getText().toString();
    int asteriskPosition = labelString.indexOf("*");
    final SpannableStringBuilder sb = new
SpannableStringBuilder(labelString);
    final ForegroundColorSpan redStyle = new
ForegroundColorSpan(Color.WHITE);
    sb.setSpan(redStyle, asteriskPosition, asteriskPosition+1,
Spannable.SPAN_INCLUSIVE_INCLUSIVE);
    view.setText(sb);
}
}
```

```
package gov.dhs.mobile.foia.viewcontrollers.submission.step1;
```

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
```

```

import android.text.Spannable;
import android.text.SpannableStringBuilder;
import android.text.style.ForegroundColorSpan;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.TextView;

import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXXX.CRConstants;
import XXXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.arrayadapters.ComponentArrayAdapter;
import gov.dhs.mobile.efoia.datastructures.Submission;
import gov.dhs.mobile.efoia.models.Component;
import gov.dhs.mobile.efoia.models.SubmissionPageOne;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step2.SubmissionStep2A
ctivity;
import gov.dhs.mobile.efoia.webview.FOIAWebView;

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;

/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material

```

```

* | is strictly forbidden unless prior written permission is obtained
* | from XXXXXXXXXXXXX.
* |
*****
* | <Insert your Javadoc Here>
* |
*****
*/

```

```

public class SubmissionStep1Activity extends Activity {

```

```

    private TextView    step_1_desc;
    private TextView    step_1_title;
    private TextView    step_page_title;
    private TextView    select_component_text;
    private TextView    component;
    private TextView    select_text;
    private Button      choose_secretary_button;
    private Button      select_component_url;
    private Spinner     select_dhs_component;
    private Button      next_button;
    private Submission  submission;
    private SubmissionPageOne pageOneLinks;
    private Context     mContext = (Context)this;
    private List<Component> componentList;
    private boolean     set;

    private TaskListener<DynamicContentResponse<Component>>
    cmsTaskListener =
        new TaskListener<DynamicContentResponse<Component>>() {
            @Override
            public void
    onTaskSuccess(DynamicContentResponse<Component> result) {
                if (result != null && result.getContent() != null
    && result.getContent().size() > 0)
                {

                    componentList = result.getContent();

                    componentList.add(0,new Component(0L,"Select
Component","",null,""));

                    ComponentArrayAdapter adapter = new
                        ComponentArrayAdapter(mContext,
                            R.layout.spinner_list_item,
    componentList);

                    select_dhs_component.setAdapter(adapter);
                }
            }
        }

```

```

select_dhs_component.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?>
parent, View view, int position, long id)
    {

submission.setComponent(componentList.get(position).getValue()+" ");

        if (position >0)
        {
            next_button.setEnabled(true);
        }
        else
        {
            next_button.setEnabled(false);
        }

    }

    @Override
    public void
onNothingSelected(AdapterView<?> parent) {

    }

});

    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }

};

    private TaskListener<DynamicContentResponse<SubmissionPageOne>>
pageOneTaskListener =

```

```

        new
TaskListener<DynamicContentResponse<SubmissionPageOne>>() {
    @Override
    public void
onTaskSuccess(DynamicContentResponse<SubmissionPageOne> result) {
        if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
        {
            pageOneLinks = result.getContent().get(0);

        }
    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_submission_step1);

    step_1_desc =
(TextView)findViewById(R.id.step_1_desc);
    step_1_title =
(TextView)findViewById(R.id.step_1_title);
    step_page_title =
(TextView)findViewById(R.id.step_page_title);
    next_button =
(Button)findViewById(R.id.next_button);
    component =
(TextView)findViewById(R.id.component);
    select_text =
(TextView)findViewById(R.id.select_text);
    choose_secretary_button =
(Button)findViewById(R.id.choose_secretary_button);
    select_component_url =
(Button)findViewById(R.id.select_component_url);
}

```



```

        select_component_text =
(TextView)findViewById(R.id.select_component_text);
        select_dhs_component =
(Spinner)findViewById(R.id.select_dhs_component);

        Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");
        Typeface light =
Typeface.createFromAsset(getAssets(),"nexa_light.otf");

        step_1_desc.setTypeface(light);
        step_1_title.setTypeface(bold);
        step_page_title.setTypeface(bold);
        component.setTypeface(light);
        select_text.setTypeface(bold);
        choose_secretary_button.setTypeface(light);
        select_component_url.setTypeface(light);
        select_component_text.setTypeface(light);

        //Disable the next button by default
        next_button.setEnabled(false);

        //Create a new Submission
        submission = new Submission();

        attributeText(select_text);

        set = false;

        Type adsType = new
TypeToken<DynamicContentResponse<Component>>().getType();
        GetDynamicContent<DynamicContentResponse<Component>>
advertisementsTask =
                new
DynamicContentTaskBuilder<DynamicContentResponse<Component>>()
                .withContext(this)
                .withMethod(HttpGet.METHOD_NAME)
                .withPermanentLink("dhs_component")
                .withParam("page","0")
                .withTaskListener(cmsTaskListener)
                .withType(adsType)
                .build();

        advertisementsTask.execute();

        Type linksType = new
TypeToken<DynamicContentResponse<SubmissionPageOne>>().getType();
        GetDynamicContent<DynamicContentResponse<SubmissionPageOne>>
task =
                new

```

```

DynamicContentTaskBuilder<DynamicContentResponse<SubmissionPageOne>>()
    .withContext(this)
    .withMethod(HttpGet.METHOD_NAME)
    .withPermanentLink("submission_1")
    .withTaskListener(pageOneTaskListener)
    .withType(linksType)
    .build();
task.execute();

ComponentArrayAdapter adapter = new
    ComponentArrayAdapter(this,
        R.layout.spinner_list_item, componentList);

}

public void goBack(View v)
{
    this.finish();
}

public void showSecretary(View v)
{
    Intent webViewIntent = new
Intent(SubmissionStep1Activity.this,FOIAWebView.class);

    Bundle bundle = new Bundle();

    bundle.putString(CRConstants.EXTRAS_KEY_URL,
pageOneLinks.getSecretary());
    bundle.putString(CRConstants.EXTRAS_KEY_TITLE,"Office of the
Secretary");
    webViewIntent.putExtras(bundle);

    startActivity(webViewIntent);
}

public void showComponent(View v)
{
    Intent webViewIntent = new
Intent(SubmissionStep1Activity.this,FOIAWebView.class);

    Bundle bundle = new Bundle();

    bundle.putString(CRConstants.EXTRAS_KEY_URL,
pageOneLinks.getComponent());
    bundle.putString(CRConstants.EXTRAS_KEY_TITLE,"Component");
    webViewIntent.putExtras(bundle);
}

```

```

        startActivity(webViewIntent);
    }

    public void next(View v)
    {
        Intent intent = new Intent(SubmissionStep1Activity.this,
SubmissionStep2Activity.class);

        Bundle bundle= new Bundle();
        bundle.putParcelable("submission", submission);

        intent.putExtras(bundle);
        startActivity(intent);
    }

    private void attributeText(TextView view)
    {
        String labelString = view.getText().toString();

        int asteriskPosition = labelString.indexOf("*");

        final SpannableStringBuilder sb = new
SpannableStringBuilder(labelString);

        final ForegroundColorSpan redStyle = new
ForegroundColorSpan(Color.WHITE);

        sb.setSpan(redStyle, asteriskPosition, asteriskPosition + 1,
Spannable.SPAN_INCLUSIVE_INCLUSIVE);

        view.setText(sb);
    }
}

package gov.dhs.mobile.efoia.viewcontrollers.submission.step6;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;

import android.text.Spannable;
import android.text.SpannableStringBuilder;

```

```

import android.text.style.ForegroundColorSpan;
import android.view.View;

import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.ToggleButton;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.datastructures.Submission;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step7.SubmissionStep7A
ctivity;
/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */
public class SubmissionStep6Activity extends Activity {

    private Submission submission;
    private ToggleButton infoToggle;
    private ToggleButton repToggle;
    private ToggleButton affiliateToggle;
    private ToggleButton privateToggle;
    private Button nextButton;
    private String requestType;

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_submission_step6);

    requestType = "";

    Typeface bold = Typeface.createFromAsset(getAssets(),
"nexa_bold.otf");
    Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

    Bundle bundle = getIntent().getExtras();
    submission = bundle.getParcelable("submission");

    nextButton = (Button)findViewById(R.id.nextButton);

    infoToggle = (ToggleButton)findViewById(R.id.info_toggle);
    repToggle = (ToggleButton)findViewById(R.id.rep_toggle);
    affiliateToggle =
(ToggleButton)findViewById(R.id.affiliate_toggle);
    privateToggle =
(ToggleButton)findViewById(R.id.private_toggle);

    TextView infoText = (TextView)findViewById(R.id.info_text);
    TextView repText = (TextView)findViewById(R.id.rep_text);
    TextView affiliateText =
(TextView)findViewById(R.id.affiliate_text);
    TextView feeDesc = (TextView)findViewById(R.id.feeDesc);
    TextView stepTitle =
(TextView)findViewById(R.id.step_6_title);
    TextView privateText =
(TextView)findViewById(R.id.private_text);

    infoText.setTypeface(light);
    repText.setTypeface(light);
    affiliateText.setTypeface(light);
    privateText.setTypeface(light);

    feeDesc.setTypeface(bold);
    stepTitle.setTypeface(bold);

    attributeText(feeDesc);

    nextButton.setEnabled(false);
}

public void setSubmissionType(View v)
{
    nextButton.setEnabled(true);
}

```

```

ToggleButton tgl = (ToggleButton)v;

if (tgl.getId() == infoToggle.getId())
{
    requestType = "Personal Use ";
    repToggle.setChecked(false);
    privateToggle.setChecked(false);
    affiliateToggle.setChecked(false);
}

else if(tgl.getId() == repToggle.getId())
{
    requestType = "News Media ";
    infoToggle.setChecked(false);
    privateToggle.setChecked(false);
    affiliateToggle.setChecked(false);
}

else if(tgl.getId() == privateToggle.getId())
{
    requestType = "Educational or Scholarly ";
    infoToggle.setChecked(false);
    repToggle.setChecked(false);
    affiliateToggle.setChecked(false);
}

else if(tgl.getId() == affiliateToggle.getId())
{
    requestType = "Private Corporation ";
    infoToggle.setChecked(false);
    repToggle.setChecked(false);
    privateToggle.setChecked(false);
}

}

public void goBack(View v)
{
    this.finish();
}

public void next(View v)
{
    submission.setRequestCategory(requestType);

    Intent intent = new Intent(SubmissionStep6Activity.this,
SubmissionStep7Activity.class);

```

```

        Bundle bundle = new Bundle();
        bundle.putParcelable("submission", submission);

        intent.putExtras(bundle);
        startActivity(intent);
    }

    private void attributeText(TextView view)
    {
        String labelString = view.getText().toString();

        int asteriskPosition = labelString.indexOf("*");

        final SpannableStringBuilder sb = new
SpannableStringBuilder(labelString);

        final ForegroundColorSpan redStyle = new
ForegroundColorSpan(Color.WHITE);

        sb.setSpan(redStyle, asteriskPosition, asteriskPosition + 1,
Spannable.SPAN_INCLUSIVE_INCLUSIVE);

        view.setText(sb);

    }
}

package gov.dhs.mobile.efoia.viewcontrollers.submission.step8;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

import XXXXXXXXXXXX.XXXXXXXXXXXXXConnect;
import gov.dhs.mobile.efoia.MainActivity;
import gov.dhs.mobile.efoia.R;

```

```

import gov.dhs.mobile.foia.datastructures.Submission;
import gov.dhs.mobile.foia.tasks.RetrieveFormIDTask;
import gov.dhs.mobile.foia.tasks.SubmitFOIATask;

import java.util.concurrent.ExecutionException;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *| *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 *| *****
 * | <Insert your Javadoc Here>
 * |
 *| *****
 */
public class SubmissionStep8Activity extends Activity {

    private Submission submission;
    private ToggleButton threatToggle;
    private ToggleButton urgencyToggle;
    private ToggleButton impairmentToggle;
    private ToggleButton harmToggle;
    private EditText expediteText;
    private String expediteSubmissionText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_submission_step8);

        expediteSubmissionText="";

        Typeface bold = Typeface.createFromAsset(getAssets(),

```



```

"nexa_bold.otf");
    Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

    Bundle bundle = getIntent().getExtras();
    submission = bundle.getParcelable("submission");

    TextView pageTitle =
(TextView)findViewById(R.id.step_8_title);
    TextView expediteTitle =
(TextView)findViewById(R.id.expediteTitle);
    TextView expediteDesc =
(TextView)findViewById(R.id.expediteDesc);
    TextView expediteSelectText =
(TextView)findViewById(R.id.expediteSelectText);
    TextView threatText = (TextView)findViewById(R.id.threatText);
    TextView urgencyText =
(TextView)findViewById(R.id.urgencyText);
    TextView impairmentText =
(TextView)findViewById(R.id.impairmentText);
    TextView harmText = (TextView)findViewById(R.id.harmText);
    TextView expediteTextTitle =
(TextView)findViewById(R.id.expediteTextTitle);
    Button submitButton = (Button)findViewById(R.id.submitButton);

    expediteText = (EditText)findViewById(R.id.expediteText);
    threatToggle = (ToggleButton)findViewById(R.id.threatToggle);
    urgencyToggle =
(ToggleButton)findViewById(R.id.urgencyToggle);
    impairmentToggle =
(ToggleButton)findViewById(R.id.impairmentToggle);
    harmToggle = (ToggleButton)findViewById(R.id.harmToggle);

    pageTitle.setTypeface(bold);
    expediteTitle.setTypeface(light);
    expediteDesc.setTypeface(light);
    expediteSelectText.setTypeface(bold);
    threatText.setTypeface(light);
    urgencyText.setTypeface(light);
    impairmentText.setTypeface(light);
    harmText.setTypeface(light);
    expediteTextTitle.setTypeface(bold);
    expediteText.setTypeface(light);

    submitButton.setEnabled(false);

```

```

RetrieveFormIDTask task = new RetrieveFormIDTask();

try
{
    submission.setDrupalFormID(task.execute().get());
    submitButton.setEnabled(true);
}
catch (InterruptedException ie)
{
    Toast.makeText(this, "An Error has Occurred Please Try
Again Later", Toast.LENGTH_SHORT);
}
catch (ExecutionException ee)
{
    Toast.makeText(this, "An Error has Occurred Please Try
Again Later", Toast.LENGTH_SHORT);
}

}

public void goBack(View v)
{
    this.finish();
}

public void setSubmissionType(View v)
{
    ToggleButton tgl = (ToggleButton)v;

    if (tgl.getId() == threatToggle.getId())
    {
        harmToggle.setChecked(false);
        impairmentToggle.setChecked(false);
        urgencyToggle.setChecked(false);

        expediteSubmissionText=
            "Imminent threat to the life or physical safety of
an individual ";
    }
    else if (tgl.getId() == harmToggle.getId())
    {
        threatToggle.setChecked(false);
        impairmentToggle.setChecked(false);
        urgencyToggle.setChecked(false);

        expediteSubmissionText= "A harm to substantial
humanitarian concerns exists ";
    }
}

```

```

else if (tgl.getId() == impairmentToggle.getId())
{
    harmToggle.setChecked(false);
    threatToggle.setChecked(false);
    urgencyToggle.setChecked(false);
    expediteSubmissionText=
        "An impairment of substantial due process rights
of the requester exists ";
}
else
{
    harmToggle.setChecked(false);
    impairmentToggle.setChecked(false);
    impairmentToggle.setChecked(false);
    expediteSubmissionText=
        "Urgency to inform the public concerning actual"+
        " or alleged Federal Government activity
";
}
}

}

public void submitFOIA(View v)
{
    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this, "Submission", "Executed Submission",
"", "");
    //Fill in the data
    SubmitFOIATask task = new SubmitFOIATask();

    submission.setExpediteRequested(expediteSubmissionText);

    submission.setExpediteRequestedDescription(expediteText.getText().toSt
ring());

    try {
        //Boolean success = task.execute(submission).get();
        boolean success = true;
        if (success) {
            AlertDialog.Builder builder = new
AlertDialog.Builder(this);

            builder.setMessage("Your FOIA Request Has Been
Submitted")

                .setTitle("FOIA Submission")
                .setCancelable(false)
                .setPositiveButton("OK", new

```

```

DialogInterface.OnClickListener() {
    public void onClick(DialogInterface
dialog, int id) {
        Intent intent = new
Intent(getApplicationContext(), MainActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    }
});
AlertDialog alert = builder.create();
alert.show();
}
else
{
    AlertDialog.Builder builder = new
AlertDialog.Builder(this);
builder.setMessage("An Error has Occurred Please Try
Again Later")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface
dialog, int id) {
                Intent intent = new
Intent(getApplicationContext(), MainActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
            }
        });
    AlertDialog alert = builder.create();
    alert.show();
}

} catch (Exception e) {
    AlertDialog.Builder builder = new
AlertDialog.Builder(this);
builder.setMessage("An Error has Occurred Please Try Again
Later")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
int id) {
                Intent intent = new
Intent(getApplicationContext(), MainActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
            }
        });
    AlertDialog alert = builder.create();
    alert.show();
}
}

```

```
        }
    });
    AlertDialog alert = builder.create();
    alert.show();
}
}
```

```
package gov.dhs.mobile.efoia.viewcontrollers.submission.step4;
```

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.TextView;
```

```
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.datastructures.Submission;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step5.SubmissionStep5A
ctivity;
```

```
/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
```

```

* |
*****
* | <Insert your Javadoc Here>
* |
*****
*/
public class SubmissionStep4Activity extends Activity {

    private Submission submission;
    private EditText recordsRequest;
    private TextView compDes;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_submission_step4);

        Typeface bold    =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");
        Typeface light   = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

        TextView title    = (TextView)
findViewById(R.id.step_4_title);
        TextView compTitle = (TextView)
findViewById(R.id.componentTitle);
        compDes           = (TextView)
findViewById(R.id.componentDesc);
        TextView request  = (TextView)
findViewById(R.id.recordDesc);

        recordsRequest    = (EditText)
findViewById(R.id.recordRequest);

        Bundle bundle = getIntent().getExtras();
        submission = bundle.getParcelable("submission");

        title.setTypeface(bold);
        compTitle.setTypeface(light);
        compDes.setTypeface(light);

        request.setTypeface(bold);
        recordsRequest.setTypeface(light);
    }

    public void goBack(View v)
    {
        this.finish();
    }
}

```

```
    }

    public void next(View v)
    {
        submission.setRecordsDescription(recordsRequest.getText().toString());

        Intent intent = new Intent(SubmissionStep4Activity.this,
        SubmissionStep5Activity.class);

        Bundle bundle= new Bundle();
        bundle.putParcelable("submission",submission);

        intent.putExtras(bundle);
        startActivity(intent);
    }
}
```

```
package gov.dhs.mobile.efoia.viewcontrollers.submission.step7;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.ToggleButton;

import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.datastructures.Submission;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step8.SubmissionStep8A
ctivity;
/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 * *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
```

```

* | herein are proprietary to XXXXXXXXXXXX
* | and its suppliers and may be covered by U.S. and Foreign Patents,
* | patents in process, and are protected by trade secret or
copyright law.
* | Dissemination of this information or reproduction of this
material
* | is strictly forbidden unless prior written permission is obtained
* | from XXXXXXXXXXXX.
* |
*****
* | <Insert your Javadoc Here>
* |
*****
*/
public class SubmissionStep7Activity extends Activity
{

    private Submission submission;
    private EditText waiverText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_submission_step7);

        Typeface bold = Typeface.createFromAsset(getAssets(),
"nexa_bold.otf");
        Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");

        Bundle bundle = getIntent().getExtras();
        submission = bundle.getParcelable("submission");

        TextView title = (TextView)findViewById(R.id.step_7_title);
        TextView waiverTitle =
(TextView)findViewById(R.id.waivertitle);
        TextView waiverDesc = (TextView)findViewById(R.id.waiverDesc);
        TextView reqWaiverText =
(TextView)findViewById(R.id.requestWaiverText);
        ToggleButton toggleButton =
(ToggleButton)findViewById(R.id.toggleButton);
        TextView requestText =
(TextView)findViewById(R.id.request_text);
        TextView waiverTextTitle =
(TextView)findViewById(R.id.waivertexttitle);
        waiverText = (EditText)findViewById(R.id.waiverText);

        title.setTypeface(bold);
        waiverTitle.setTypeface(light);
        waiverDesc.setTypeface(light);

```



```

        reqWaiverText.setTypeface(bold);
        requestText.setTypeface(light);
        waiverTextTitle.setTypeface(bold);
        waiverText.setTypeface(light);
    }

    public void goBack(View v)
    {
        this.finish();
    }

    public void setSubmissionType(View v)
    {
        ToggleButton tgl = (ToggleButton)v;

        if (tgl.isChecked())
        {
            submission.setRequestFeeWaiver(true);
        }
        submission.setFeeWaiverDescription(waiverText.getText().toString());
    }
    else
    {
        submission.setRequestFeeWaiver(false);
        submission.setFeeWaiverDescription("");
        waiverText.setText("");
    }
}

    public void next(View v)
    {
        Intent intent = new Intent(SubmissionStep7Activity.this,
SubmissionStep8Activity.class);

        Bundle bundle = new Bundle();
        bundle.putParcelable("submission", submission);

        intent.putExtras(bundle);
        startActivity(intent);
    }
}

package gov.dhs.mobile.efoia.viewcontrollers.status;

import android.app.Activity;
import android.app.AlertDialog;

```

```

import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import XXXXXXXXXXXX.XXXXXXXXXXXXXConnect;

import gov.dhs.mobile.foia.MainActivity;
import gov.dhs.mobile.foia.R;
import gov.dhs.mobile.foia.datastructures.CheckStatus;
import gov.dhs.mobile.foia.tasks.CheckStatusTask;
import gov.dhs.mobile.foia.tasks.GetFormIDTask;

import java.util.ConcurrentModificationException;
import java.util.HashMap;
import java.util.concurrent.ExecutionException;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 *****
 * | <Insert your Javadoc Here>
 * |
 *****
 */
public class FOIAStatus extends Activity {

```

```

private EditText reqNumberEdit;
private TextView closedDate;
private TextView delDate;
private TextView status;
private TextView recv;
private TextView reqNumber;
private String formID;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_foiastatus);

    TextView title =
    (TextView)findViewById(R.id.check_title);
    TextView desc =
    (TextView)findViewById(R.id.check_desc);
    TextView enterReqLabel =
    (TextView)findViewById(R.id.enter_req_number_label);
    TextView dateLabel =
    (TextView)findViewById(R.id.est_del_date_label);
    TextView closeDateLbl =
    (TextView)findViewById(R.id.closed_date_label);
    TextView statusLabel =
    (TextView)findViewById(R.id.req_status_label);
    TextView date =
    (TextView)findViewById(R.id.recv_date_label);
    Button statusBtn =
    (Button)findViewById(R.id.check_status_button);
    TextView reqLabel =
    (TextView)findViewById(R.id.request_number_label);
    reqNumberEdit =
    (EditText)findViewById(R.id.request_number_edit_text);
    reqNumber =
    (TextView)findViewById(R.id.request_number);
    recv =
    (TextView)findViewById(R.id.received_date);
    status =
    (TextView)findViewById(R.id.request_status);
    delDate =
    (TextView)findViewById(R.id.delivery_date);
    closedDate =
    (TextView)findViewById(R.id.closed_date);

    Typeface bold =
    Typeface.createFromAsset(getAssets(),"nexa_bold.otf");
    Typeface light =
    Typeface.createFromAsset(getAssets(),"nexa_light.otf");

```

```

title.setTypeface(bold);
desc.setTypeface(bold);
enterReqLabel.setTypeface(bold);
reqNumberEdit.setTypeface(light);
reqLabel.setTypeface(bold);
reqNumber.setTypeface(bold);
date.setTypeface(bold);
recv.setTypeface(bold);
statusLabel.setTypeface(bold);
status.setTypeface(bold);
dateLabel.setTypeface(bold);
delDate.setTypeface(bold);
closeDateLbl.setTypeface(bold);
closedDate.setTypeface(bold);

reqNumberEdit.setRawInputType(InputType.TYPE_CLASS_TEXT);
reqNumberEdit.setImeOptions(EditorInfo.IME_ACTION_GO);

//Request the Page
GetFormIDTask task = new GetFormIDTask();

try{
    formID = task.execute().get();
}
catch (InterruptedException ie)
{

}
catch (ConcurrentModificationException cme)
{

}
catch (ExecutionException ee)
{

}
}

public void goBack(View v)
{
    this.finish();
}

public void onClick(View v)
{

    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    builder.setMessage("Requesting FOIA Status")

```

```

        .setTitle("FOIA Status")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener()
    {
        public void onClick(DialogInterface dialog, int
id)
            {
                dialog.dismiss();
            }
    });

AlertDialog alert = builder.create();
alert.show();

XXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

sdk.trackInsightV2(this, "Check Status", "Executed Check", "",
""");

if (reqNumberEdit.getText().toString().length() > 1) {
    //Fill in the data
    CheckStatus form = new CheckStatus();
    form.setDrupalFormID("formID");

form.setFoia_request_number(reqNumberEdit.getText().toString());

    CheckStatusTask task = new CheckStatusTask();

    try {
        HashMap<String, String> values =
task.execute(form).get();

        closedDate.setText(values.get("Closed Date"));
        recv.setText(values.get("Received Date"));
        delDate.setText(values.get("Estimated Delivery
Date"));
        status.setText(values.get("Request Status"));
        reqNumber.setText(reqNumberEdit.getText().toString());

        alert.dismiss();

    } catch (Exception e) {
        alert.dismiss();

        builder.setMessage("An Error has Occurred Please Try
Again Later")
            .setTitle("FOIA Status")

```



```

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 * | *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 * | *****
 * | <Insert your Javadoc Here>
 * |
 * | *****
 */
public class EventsActivity extends Activity {

    private ListView activityListView;
    private List<EventCalendarItem> eventList;
    public static int mSelectedItem = -1;
    private EventCalendarTableArrayAdapter adapter;

    private TaskListener<DynamicContentResponse<EventCalendarItem>>
cmsTaskListener =
        new
TaskListener<DynamicContentResponse<EventCalendarItem>>()
    {
        @Override
        public void
onTaskSuccess(DynamicContentResponse<EventCalendarItem> result) {
            if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
            {
                eventList = result.getContent();
            }
        }
    }
}

```

```

        adapter = new
EventCalendarTableArrayAdapter(EventsActivity.this,
                                R.layout.event_list_item,eventList);

        activityListView.setAdapter(adapter);
    }
}

@Override
public void onTaskFailure(Exception e) {

}

@Override
public void onTaskCancelled() {

}

@Override
public void onTaskTimeOut() {

}
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_events);

    Bundle bundle = getIntent().getExtras();
    String actionObject = bundle.getString("action_object");

    TextView line1 = (TextView)
findViewById(R.id.event_title_label);
    TextView line2 = (TextView)
findViewById(R.id.calendar_title_label);

    Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");
    Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");

    line1.setTypeface(light);
    line2.setTypeface(bold);

    Type adsType = new
TypeToken<DynamicContentResponse<EventCalendarItem>>().getType();
    GetDynamicContent<DynamicContentResponse<EventCalendarItem>>
advertisementsTask =
        new

```



```

DynamicContentTaskBuilder<DynamicContentResponse<EventCalendarItem>>()
    .withContext(this)
    .withMethod(HttpGet.METHOD_NAME)
    .withPermanentLink(actionObject)
    .withTaskListener(cmsTaskListener)
    .withType(adsType)
    .build();
    advertisementsTask.execute();

    activityListView =
(ListView)findViewById(R.id.event_list_view);

    activityListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
            EventCalendarItem clickedItem =
eventList.get(position);

            mSelectedItem = position;
            adapter.notifyDataSetChanged();

            Intent detailsIntent = new Intent(EventsActivity.this,
EventDetailsActivity.class);

            Bundle bundle = new Bundle();

            bundle.putString("dateTime",
clickedItem.getEvent_date());
            bundle.putString("location",
clickedItem.getLocation());
            bundle.putString("desc", clickedItem.getDetails());
            bundle.putString("title", clickedItem.getTitle());

            detailsIntent.putExtras(bundle);
            startActivity(detailsIntent);

        }
    });
}

public void goBack(View v)
{
    this.finish();
}
}

```

```
package gov.dhs.mobile.foia.viewcontrollers.privacypolicy;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;

import com.google.gson.reflect.TypeToken;

import XXXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.foia.R;
import gov.dhs.mobile.foia.models.PrivacyPolicy;
import
gov.dhs.mobile.foia.viewcontrollers.submission.step1.SubmissionStep1A
ctivity;
import gov.dhs.mobile.foia.MainActivity;
import gov.dhs.mobile.foia.R;
import gov.dhs.mobile.foia.models.PrivacyPolicy;
import org.apache.http.client.methods.HttpGet;
import java.lang.reflect.Type;
import java.util.List;

/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *****
 * | Copyright (c) XXXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
```

```

copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */

```

```

public class PrivacyPolicyActivity extends Activity
{
    private TextView policyTextView;
    private boolean firstLaunch;

    private TaskListener<DynamicContentResponse<PrivacyPolicy>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<PrivacyPolicy>>()
    {
        @Override
        public void
onTaskSuccess(DynamicContentResponse<PrivacyPolicy> result) {
            if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
            {
                List<PrivacyPolicy> policyList =
result.getContent();

                PrivacyPolicy policy = policyList.get(0);

policyTextView.setText(Html.fromHtml(policy.getPolicy()));

            }
        }

        @Override
        public void onTaskFailure(Exception e) {

        }

        @Override
        public void onTaskCancelled() {

        }

        @Override
        public void onTaskTimeOut() {

```

```

        };
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_privacy_policy);

        Bundle bundle = getIntent().getExtras();
        String actionObject = bundle.getString("action_object");
        boolean firstLaunch = bundle.getBoolean("first_launch");

        Button acceptButton = (Button)
        findViewById(R.id.accept_button);
        Button rejectButton = (Button)
        findViewById(R.id.reject_button);
        ImageButton backButton = (ImageButton)
        findViewById(R.id.imageButton2);

        if (!firstLaunch)
        {
            acceptButton.setVisibility(View.GONE);
            rejectButton.setVisibility(View.GONE);
        } else {
            backButton.setVisibility(View.GONE);
        }
        policyTextView =
        (TextView)findViewById(R.id.privacy_policy_textview);
        policyTextView.setMovementMethod(new LinkMovementMethod());

        Typeface font =
        Typeface.createFromAsset(getAssets(),"nexa_light.otf");
        policyTextView.setTypeface(font);

        Type adsType = new
        TypeToken<DynamicContentResponse<PrivacyPolicy>>().getType();
        GetDynamicContent<DynamicContentResponse<PrivacyPolicy>>
        advertisementsTask =
            new
        DynamicContentTaskBuilder<DynamicContentResponse<PrivacyPolicy>>()
            .withContext(this)
            .withMethod(HttpGet.METHOD_NAME)
            .withPermanentLink(actionObject)
            .withTaskListener(cmsTaskListener)
            .withType(adsType)
            .build();
        advertisementsTask.execute();
    }

```



```

import android.widget.TextView;
import android.widget.Toast;

import XXXXXXXXXXXX.XXXXXXXXXXXXXConnect;
import XXXXXXXXXXXX.models.MobileMenu;
import XXXXXXXXXXXX.models.json.response.MobileMenuResponse;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.foia.R;

import gov.dhs.mobile.foia.arrayadapters.HowToViewTableArrayAdapter;
import
gov.dhs.mobile.foia.viewcontrollers.feestructure.FeeStructureActivity
;
import
gov.dhs.mobile.foia.viewcontrollers.fivesteps.FiveStepsActivity;
import gov.dhs.mobile.foia.viewcontrollers.limits.LimitsActivity;
import
gov.dhs.mobile.foia.viewcontrollers.privacyact.PrivacyActActivity;
import
gov.dhs.mobile.foia.viewcontrollers.processing.ProcessingActivity;
import gov.dhs.mobile.foia.tasks.GetSubMenusTask;

import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****/
public class FOIAHowToActivity extends Activity
{

```

```

private ListView activityListView;
private List<MobileMenu> menus;
public static int mSelectedItem=-1;
private HowToViewTableArrayAdapter adapter;

private TaskListener<MobileMenuResponse> mobileMenusTaskCallback =
new TaskListener<MobileMenuResponse>() {
    @Override
    public void onTaskSuccess(MobileMenuResponse result) {
        if (result != null && result.getMobileMenus() != null &&
result.getMobileMenus().size() > 0){

            menus = result.getMobileMenus();

            adapter = new
HowToViewTableArrayAdapter(FOIAHowToActivity.this,
                R.layout.how_to_list_item,menus);

            activityListView.setAdapter(adapter);
        }
    }

    @Override
    public void onTaskFailure(Exception e) {
        Toast.makeText(FOIAHowToActivity.this, "Your token has
been saved", Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_howto);
XXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    Bundle bundle = getIntent().getExtras();
    int menuId = bundle.getInt("menu_id");

```

```

Typeface light = Typeface.createFromAsset(getAssets(),
    "nexa_light.otf");

Typeface bold = Typeface.createFromAsset(getAssets(),
    "nexa_bold.otf");

    new GetSubMenusTask(FOIAHowToActivity.this,
mobileMenusTaskCallback,menuId).execute();

    TextView title = (TextView)findViewById(R.id.how_to_title);
    TextView desc = (TextView)findViewById(R.id.how_to_desc);

    desc.setTypeface(light);
    title.setTypeface(bold);

        activityListView = (ListView)
findViewById(R.id.howto_listview);

        activityListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
                MobileMenu menu = menus.get(position);
                mSelectedItem = position;
                launchSubActivity(menu);
                adapter.notifyDataSetChanged();
            }
        });
    }

private void launchSubActivity(MobileMenu submenu)
{
    Bundle bundle = new Bundle();
    Intent intent=null;
    bundle.putString("action_object", (String)
submenu.getActionObject());

    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this, "How To Menu", submenu.getAction(),
"", "");

    if
(((String)submenu.getActionObject()).equalsIgnoreCase("five_steps"))

```



```

        {
            intent = new Intent(FOIAHowToActivity.this,
FiveStepsActivity.class);
        }
        else if
(((String)submenu.getActionObject()).equalsIgnoreCase("fee_structure")
)
        {
            intent = new Intent(FOIAHowToActivity.this,
FeeStructureActivity.class);
        }
        else if
(((String)submenu.getActionObject()).equalsIgnoreCase("processing"))
        {
            intent = new Intent(FOIAHowToActivity.this,
ProcessingActivity.class);
        }
        else if
(((String)submenu.getActionObject()).equalsIgnoreCase("limits_exemptio
ns"))
        {
            intent = new Intent(FOIAHowToActivity.this,
LimitsActivity.class);
        }
        else
        {
            intent = new Intent(FOIAHowToActivity.this,
PrivacyActActivity.class);
        }

        if (intent != null) {
            intent.putExtras(bundle);
            startActivity(intent);
        }
    }

    public void goBack(View v)
    {
        this.finish();
    }
}

```

```

package gov.dhs.mobile.efoia.viewcontrollers.eventdetails;

```

```

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;

```

```

import android.view.View;
import android.widget.TextView;

import gov.dhs.mobile.efoia.R;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 *****
 * | <Insert your Javadoc Here>
 * |
 *****
 */
public class EventDetailsActivity extends Activity
{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_event_details);

        Bundle bundle = getIntent().getExtras();
        String time = bundle.getString("dateTime");
        String location = bundle.getString("location");
        String desc = bundle.getString("desc");
        String title = bundle.getString("title");

```

```

        Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");
        Typeface bold = Typeface.createFromAsset(getAssets(),
"nexa_bold.otf");

        TextView eventTitle =
(TextView)findViewById(R.id.event_details_event_title);
        TextView eventDate =
(TextView)findViewById(R.id.event_details_event_date);
        TextView eventDesc =
(TextView)findViewById(R.id.event_details_event_desc);
        TextView eventTime =
(TextView)findViewById(R.id.event_details_event_time);
        TextView eventLocation =
(TextView)findViewById(R.id.event_details_event_location);

        eventDesc.setMovementMethod(new LinkMovementMethod());

        eventTitle.setTypeface(bold);
        eventDate.setTypeface(bold);
        eventDesc.setTypeface(light);
        eventTime.setTypeface(light);
        eventLocation.setTypeface(bold);

        eventTitle.setText(title);
        eventDesc.setText(Html.fromHtml(desc));
        eventLocation.setText(location);

a");
        SimpleDateFormat dt = new SimpleDateFormat("MM/dd/yy HH:mm:ss
SimpleDateFormat dt1 = new SimpleDateFormat("MMMM dd, yyyy");
SimpleDateFormat dt2 = new SimpleDateFormat("hh:mm a");
        try {
            Date date = dt.parse(time);

            eventTime.setText(dt2.format(date));
            eventDate.setText(dt1.format(date)+" | ");

        } catch (ParseException e) {
            e.printStackTrace();
        }

    }

    public void goBack(View v)
    {

```

```

        this.finish();
    }
}

package gov.dhs.mobile.efoia.viewcontrollers.exemptions;

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;

import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.arrayadapters.ExemptionArrayAdapter;
import gov.dhs.mobile.efoia.models.Exemption;

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *****
 * | Copyright (c) XXXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXXX.
 * |
 *****
 * | <Insert your Javadoc Here>
 * |

```

```

*****
*/
public class ExemptionsActivity extends Activity {

    private ListView activityListView;
    private List<Exemption> listItems;
    public static int arrayIndex;

    private TaskListener<DynamicContentResponse<Exemption>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<Exemption>>()
        {
            @Override
            public void
onTaskSuccess(DynamicContentResponse<Exemption> result) {
                if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                {
                    listItems = result.getContent();

                    arrayIndex = 0;

                    activityListView.setAdapter(new

ExemptionArrayAdapter(ExemptionsActivity.this,

R.layout.privacy_act_list_item, listItems));
                }

            @Override
            public void onTaskFailure(Exception e) {

            }

            @Override
            public void onTaskCancelled() {

            }

            @Override
            public void onTaskTimeOut() {

            }

        }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_exemptions);
}

```

```

        Bundle bundle = getIntent().getExtras();
        String actionObject = bundle.getString("action_object");

        Type adsType = new
TypeToken<DynamicContentResponse<Exemption>>().getType();
        GetDynamicContent<DynamicContentResponse<Exemption>>
advertisementsTask =
            new
DynamicContentTaskBuilder<DynamicContentResponse<Exemption>>()
                .withContext(this)
                .withMethod(HttpGet.METHOD_NAME)
                .withPermanentLink(actionObject)
                .withTaskListener(cmsTaskListener)
                .withType(adsType)
                .build();
        advertisementsTask.execute();

        activityListView =
(ListView)findViewById(R.id.exemptions_listview);

        Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");

        TextView title = (TextView)
findViewById(R.id.exemptions_title);
        TextView desc = (TextView) findViewById(R.id.exemption_desc);

        title.setTypeface(bold);
        desc.setTypeface(bold);

    }

    public void goBack(View v)
    {
        this.finish();
    }
}

package gov.dhs.mobile.foia.viewcontrollers.feestructure;

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;

```

```

import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.arrayadapters.FeeStructureArrayAdapter;
import gov.dhs.mobile.efoia.models.FeeStructure;

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */
public class FeeStructureActivity extends Activity {

    private ListView activityListView;
    private List<FeeStructure> listItems;
    public static int arrayIndex;

    private TaskListener<DynamicContentResponse<FeeStructure>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<FeeStructure>>()
        {
            @Override

```

```

        public void
onTaskSuccess(DynamicContentResponse<FeeStructure> result) {
            if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
            {
                listItems = result.getContent();

                arrayIndex = 0;

                activityListView.setAdapter(new

FeeStructureArrayAdapter(FeeStructureActivity.this,
R.layout.privacy_act_list_item, listItems));
            }
        }

        @Override
        public void onTaskFailure(Exception e) {

        }

        @Override
        public void onTaskCancelled() {

        }

        @Override
        public void onTaskTimeOut() {

        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fee_structure);

        Bundle bundle = getIntent().getExtras();
        String actionObject = bundle.getString("action_object");

        Type adsType = new
TypeToken<DynamicContentResponse<FeeStructure>>().getType();
        GetDynamicContent<DynamicContentResponse<FeeStructure>>
advertisementsTask =
            new
DynamicContentTaskBuilder<DynamicContentResponse<FeeStructure>>()
                .withContext(this)
                .withMethod(HttpGet.METHOD_NAME)
                .withPermanentLink(actionObject)

```



```

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;

/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 * | *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 * | *****
 * | <Insert your Javadoc Here>
 * |
 * | *****/

public class FiveStepsActivity extends Activity {

    private ListView activityListView;
    private List<FiveStepsItem> eventList;
    public static int arrayIndex;

    private TaskListener<DynamicContentResponse<FiveStepsItem>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<FiveStepsItem>>()
        {
            @Override
            public void
onTaskSuccess(DynamicContentResponse<FiveStepsItem> result) {
                if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                {
                    eventList = result.getContent();

```

```

        arrayIndex = 0;

        activityListView.setAdapter(new
FiveStepsTableArrayAdapter(FiveStepsActivity.this,
R.layout.five_steps_list_item,eventList));
    }
}

@Override
public void onTaskFailure(Exception e) {

}

@Override
public void onTaskCancelled() {

}

@Override
public void onTaskTimeOut() {

}
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_five_steps);

    Bundle bundle = getIntent().getExtras();
    String actionObject = bundle.getString("action_object");

    Type adsType = new
TypeToken<DynamicContentResponse<FiveStepsItem>>().getType();
    GetDynamicContent<DynamicContentResponse<FiveStepsItem>>
advertisementsTask =
        new
DynamicContentTaskBuilder<DynamicContentResponse<FiveStepsItem>>()
            .withContext(this)
            .withMethod(HttpGet.METHOD_NAME)
            .withPermanentLink(actionObject)
            .withTaskListener(cmsTaskListener)
            .withType(adsType)
            .build();
    advertisementsTask.execute();

    activityListView =
(ListView)findViewById(R.id.five_steps_listView);

```

```

        Typeface bold = Typeface.createFromAsset(getAssets(),
"nexa_bold.otf");

        TextView title = (TextView)
findViewById(R.id.five_steps_title);
        TextView desc = (TextView) findViewById(R.id.five_steps_desc);

        title.setTypeface(bold);
        desc.setTypeface(bold);

        activityListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
int position, long id)
            {

            }

        });
    }

    public static int getArrayIndex()
    {
        arrayIndex++;
        Log.e("FiveStepsActivity","Array Index is:"+arrayIndex);
        return arrayIndex;
    }

    public void goBack(View v)
    {
        this.finish();
    }

}

package gov.dhs.mobile.efoia.viewcontrollers.library;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

```

```

import android.widget.TextView;

import com.androidquery.AQuery;
import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXX.CRConstants;
import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXX.models.CRImage;
import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.foia.R;
import gov.dhs.mobile.foia.models.Library;
import gov.dhs.mobile.foia.webview.FOIAWebView;

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;

/*****
| DHS e-FOIA
| ****
| File : FOIALibrary.java
| Created on 3/17/15
|
****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|
*****/

public class FOIALibrary extends Activity {
    private String url;
    private TextView desc;
    private TextView lTitle;
    private Button button;
    private Context mContext;
    private ImageView imageView;

```

```

        private TaskListener<DynamicContentResponse<Library>>
cmsTaskListener =
            new TaskListener<DynamicContentResponse<Library>>() {
                @Override
                public void
onTaskSuccess(DynamicContentResponse<Library> result) {
                    if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                    {
                        List<Library> libraryList =
result.getContent();

                        Library library = libraryList.get(0);
                        url = library.getLibrary_url();
                        desc.setText(library.getDescription());
                        lTitle.setText(library.getTitle());
                        String imageUrl =

library.getImage().getImageForDensity(CRImage.IMAGE_TYPE.SMALL,
                                        (Activity)mContext);

                        AQuery aq = new AQuery(mContext);
                        Bitmap preset = aq.getCachedImage(imageUrl);

                        if (preset != null) {
                            aq.id(imageView).image(imageUrl, true,
true, 0, 0, preset, AQuery.FADE_IN);
                        } else {
                            aq.id(imageView).image(imageUrl, true,
true, 0, 0);
                        }
                    }
                }
            }

        @Override
        public void onTaskFailure(Exception e) {

        }

        @Override
        public void onTaskCancelled() {

        }

        @Override
        public void onTaskTimeOut() {

        }
    };

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_foialibrary);

    mContext = (Context)this;

    Bundle bundle = getIntent().getExtras();
    String actionObject = bundle.getString("action_object");

    Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");
    Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");

    desc = (TextView)findViewById(R.id.library_desc);
    lTitle = (TextView)findViewById(R.id.library_title);
    button = (Button)findViewById(R.id.button);
    imageView = (ImageView)findViewById(R.id.library_img);

    desc.setTypeface(light);
    button.setTypeface(bold);
    lTitle.setTypeface(bold);

    Type adsType = new
TypeToken<DynamicContentResponse<Library>>().getType();
    GetDynamicContent<DynamicContentResponse<Library>>
advertisementsTask =
        new
DynamicContentTaskBuilder<DynamicContentResponse<Library>>()
            .withContext(this)
            .withMethod(HttpGet.METHOD_NAME)
            .withPermanentLink(actionObject)
            .withTaskListener(cmsTaskListener)
            .withType(adsType)
            .build();
    advertisementsTask.execute();
}

public void onClick(View view)
{
    Intent webViewIntent = new
Intent(FOIALibrary.this,FOIAWebView.class);

    Bundle bundle = new Bundle();

    bundle.putString(CRConstants.EXTRAS_KEY_URL, url);

```

```
        bundle.putString(CRConstants.EXTRAS_KEY_TITLE,"FOIA Library");
        webViewIntent.putExtras(bundle);

        startActivity(webViewIntent);
    }

    public void goBack(View v)
    {
        this.finish();
    }
}
```

```
package gov.dhs.mobile.efoia.viewcontrollers.limits;
```

```
import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;
```

```
import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.arrayadapters.LimitsArrayAdapter;
import gov.dhs.mobile.efoia.models.Limits;
```

```
import org.apache.http.client.methods.HttpGet;
```

```
import java.lang.reflect.Type;
import java.util.List;
```

```
/**
```

```
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
```

```
*****
```

```
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
```



```

copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
*****
 * | <Insert your Javadoc Here>
 * |
*****
 */
public class LimitsActivity extends Activity {

    private ListView activityListView;
    private List<Limits> listItems;
    public static int arrayIndex;

    private TaskListener<DynamicContentResponse<Limits>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<Limits>>()
        {
            @Override
            public void
onTaskSuccess(DynamicContentResponse<Limits> result) {
                if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                {
                    listItems = result.getContent();

                    arrayIndex = 0;

                    activityListView.setAdapter(new

LimitsArrayAdapter(LimitsActivity.this,
R.layout.privacy_act_list_item, listItems));
                }
            }

            @Override
            public void onTaskFailure(Exception e) {

            }

            @Override
            public void onTaskCancelled() {

            }

            @Override

```

```

        public void onTaskTimeOut() {
            }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_limits);

        Bundle bundle = getIntent().getExtras();
        String actionObject = bundle.getString("action_object");

        Type adsType = new TypeToken<DynamicContentResponse<Limits>>()
        {}.getType();
        GetDynamicContent<DynamicContentResponse<Limits>>
        advertisementsTask =
            new
        DynamicContentTaskBuilder<DynamicContentResponse<Limits>>()
            .withContext(this)
            .withMethod(HttpGet.METHOD_NAME)
            .withPermanentLink(actionObject)
            .withTaskListener(cmsTaskListener)
            .withType(adsType)
            .build();
        advertisementsTask.execute();

        activityListView =
        (ListView)findViewById(R.id.limits_listview);

        Typeface bold =
        Typeface.createFromAsset(getAssets(),"nexa_bold.otf");

        TextView title = (TextView) findViewById(R.id.limits_title);
        TextView desc = (TextView) findViewById(R.id.limits_desc);

        title.setTypeface(bold);
        desc.setTypeface(bold);

    }

    public void goBack(View v)
    {
        this.finish();
    }
}

package gov.dhs.mobile.efoia.viewcontrollers.privacyact;

```

```

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;

import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.arrayadapters.PrivacyActArrayAdapter;
import gov.dhs.mobile.efoia.models.PrivacyAct;

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 * *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 * *****
 * | <Insert your Javadoc Here>
 * |
 * *****
 */
public class PrivacyActActivity extends Activity {

```

```

private ListView activityListView;
private List<PrivacyAct> listItems;
public static int arrayIndex;

private TaskListener<DynamicContentResponse<PrivacyAct>>
cmsTaskListener =
    new TaskListener<DynamicContentResponse<PrivacyAct>>()
    {
        @Override
        public void
onTaskSuccess(DynamicContentResponse<PrivacyAct> result) {
            if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
            {
                listItems = result.getContent();

                arrayIndex = 0;

                activityListView.setAdapter(new
PrivacyActArrayAdapter(PrivacyActActivity.this,
R.layout.privacy_act_list_item, listItems));
            }

            @Override
            public void onTaskFailure(Exception e) {

            }

            @Override
            public void onTaskCancelled() {

            }

            @Override
            public void onTaskTimeOut() {

            }
        }
    };

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_privacy_act);

    Bundle bundle = getIntent().getExtras();
    String actionObject = bundle.getString("action_object");

```

```

        Type adsType = new
TypeToken<DynamicContentResponse<PrivacyAct>>() {}.getType();
        GetDynamicContent<DynamicContentResponse<PrivacyAct>>
advertisementsTask =
            new
DynamicContentTaskBuilder<DynamicContentResponse<PrivacyAct>>()
                .withContext(this)
                .withMethod(HttpGet.METHOD_NAME)
                .withPermanentLink(actionObject)
                .withTaskListener(cmsTaskListener)
                .withType(adsType)
                .build();
        advertisementsTask.execute();

        activityListView =
(ListView)findViewById(R.id.privacy_act_listview);

        Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");

        TextView title = (TextView)
findViewById(R.id.privacy_act_title);
        TextView desc = (TextView)
findViewById(R.id.privacy_act_desc);

        title.setTypeface(bold);
        desc.setTypeface(bold);

    }

    public static int getArrayIndex()
    {
        arrayIndex++;
        Log.e("FiveStepsActivity", "Array Index is:" + arrayIndex);
        return arrayIndex;
    }

    public void goBack(View v)
    {
        this.finish();
    }
}

package gov.dhs.mobile.efoia.viewcontrollers.privacystatement;

import android.app.Activity;

```

```
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.View;
import android.widget.TextView;
import gov.dhs.mobile.efoia.R;
import
gov.dhs.mobile.efoia.viewcontrollers.submission.step1.SubmissionStep1A
ctivity;
```

```
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 *****
 * | <Insert your Javadoc Here>
 * |
 *****
 */
```

```
public class PrivacyStatementActivity extends Activity
{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_privacy_statement);

TextView policyTextView;
Bundle bundle = getIntent().getExtras();

policyTextView =
(TextView)findViewById(R.id.privacy_policy_textview);
policyTextView.setMovementMethod(new LinkMovementMethod());

Typeface font =
Typeface.createFromAsset(getAssets(),"nexa_light.otf");
policyTextView.setTypeface(font);

String textString = "<!DOCTYPE html PUBLIC "-//W3C//DTD HTML
4.01//EN" \"http://www.w3.org/TR/html4/strict.dtd\">\n" +
    "<html>\n" +
    "<head>\n" +
    " <meta http-equiv=\"Content-Type\" content=\"text/
html; charset=utf-8\">\n" +
    " <meta http-equiv=\"Content-Style-Type\" content=
\"text/css\">\n" +
    " <title></title>\n" +
    "</head>\n" +
    "<body>\n" +
    "<p class=\"p1\">Privacy Act Statement</p>\n" +
    "<p class=\"p1\">eFOIA Mobile Application</p>\n" +
    "<p class=\"p2\"><span class=\"Apple-converted-space
\"> </span></p>\n" +
    "<p class=\"p3\">Authorities: 5 U.S.C. § 552, 5 U.S.C.
§ 552a, and 44 U.S.C. § 3101 authorize the collection of this
information.</p>\n" +
    "<p class=\"p2\"><span class=\"Apple-converted-space
\"> </span></p>\n" +
    "<p class=\"p3\">Purpose:<span class=\"Apple-
converted-space\"> </span>DHS will use this information to locate
applicable records and to respond to requests made under the Freedom
of Information Act (5 U.S.C. § 552) and Privacy Act of 1974 (5 U.S.C.
§ 552a).</p>\n" +
    "<p class=\"p2\"><span class=\"Apple-converted-space
\"> </span></p>\n" +
    "<p class=\"p3\">Routine Uses:<span class=\"Apple-
converted-space\"> </span>This information may be used by and
disclosed to DHS personnel, contractors, and/or other agents who need
the information to assist in activities related to the processing of
your Freedom of Information Act and/or Privacy Act request.<span
class=\"Apple-converted-space\"> </span>Additionally, DHS may use the
information, as necessary and authorized by the routine uses published

```

in the DHS/ALL-001 - Department of Homeland Security (DHS) Freedom of Information Act (FOIA) and Privacy Act (PA) Record System February 4, 2014, 79 FR 6609.</p>\n" +

"<p class=\"p2\"> </p>\n" +

"<p class=\"p3\">Disclosure: Furnishing this information is voluntary; however, failure to provide the information requested may delay or prevent DHS from processing your Freedom of Information Act or Privacy Act Request.</p>\n" +

"</body>\n" +

"</html>";

policyTextView.setText(Html.fromHtml(textString));

}

public void goBack(View v)

{

 this.finish();

}

public void goNext(View v)

{

 Intent intent = new Intent(PrivacyStatementActivity.this, SubmissionStep1Activity.class);

 startActivity(intent);

}

}

package gov.dhs.mobile.foia.viewcontrollers.processing;

import android.app.Activity;

import android.graphics.Typeface;

import android.os.Bundle;

import android.view.View;

import android.widget.ListView;

import android.widget.TextView;

import com.google.gson.reflect.TypeToken;

import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;

import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;

import XXXXXXXXXXXX.network.GetDynamicContent;

import XXXXXXXXXXXX.network.request.TaskListener;

import gov.dhs.mobile.foia.R;

import gov.dhs.mobile.foia.arrayadapters.ProcessingArrayAdapter;

import gov.dhs.mobile.foia.models.Processing;

import org.apache.http.client.methods.HttpGet;


```

import java.lang.reflect.Type;
import java.util.List;
/**
 * *****
 * | DHS e-FOIA
 * | *****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
 * *****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXX.
 * |
 * *****
 * | <Insert your Javadoc Here>
 * |
 * *****
 */
public class ProcessingActivity extends Activity {

    private ListView activityListView;
    private List<Processing> listItems;
    public static int arrayIndex;

    private TaskListener<DynamicContentResponse<Processing>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<Processing>>()
        {
            @Override
            public void
onTaskSuccess(DynamicContentResponse<Processing> result) {
                if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                {
                    listItems = result.getContent();

                    arrayIndex = 0;

                    activityListView.setAdapter(new

```

```

ProcessingArrayAdapter(ProcessingActivity.this,
R.layout.privacy_act_list_item, listItems));
        }
    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_processing);

    Bundle bundle = getIntent().getExtras();
    String actionObject = bundle.getString("action_object");

    Type adsType = new
TypeToken<DynamicContentResponse<Processing>>().getType();
    GetDynamicContent<DynamicContentResponse<Processing>>
advertisementsTask =
        new
DynamicContentTaskBuilder<DynamicContentResponse<Processing>>()
            .withContext(this)
            .withMethod(HttpGet.METHOD_NAME)
            .withPermanentLink(actionObject)
            .withTaskListener(cmsTaskListener)
            .withType(adsType)
            .build();
    advertisementsTask.execute();

    activityListView =
(ListView)findViewById(R.id.procesing_listview);

    Typeface bold =
Typeface.createFromAsset(getAssets(), "nexa_bold.otf");

```

```

        Typeface light =
Typeface.createFromAsset(getAssets(),"nexa_light.otf");

        TextView title = (TextView)
findViewById(R.id.processing_title);
        TextView desc = (TextView) findViewById(R.id.processing_desc);

        title.setTypeface(bold);
        desc.setTypeface(bold);

    }

    public void goBack(View v)
    {
        this.finish();
    }
}

package gov.dhs.mobile.efoia.viewcontrollers.statutes;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.androidquery.AQuery;
import com.google.gson.reflect.TypeToken;
import XXXXXXXXXXXX.CRConstants;
import XXXXXXXXXXXX.builders.DynamicContentTaskBuilder;
import XXXXXXXXXXXX.models.CRImage;
import XXXXXXXXXXXX.models.json.response.DynamicContentResponse;
import XXXXXXXXXXXX.network.GetDynamicContent;
import XXXXXXXXXXXX.network.request.TaskListener;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.models.Statutes;
import gov.dhs.mobile.efoia.webview.FOIAWebView;

import org.apache.http.client.methods.HttpGet;

import java.lang.reflect.Type;
import java.util.List;

```

```

/*****
| DHS e-FOIA
| ****
| File : FOIAStatutes.java
| Created on 3/17/15
|
| ****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|
| *****/

```

```

public class FOIAStatutes extends Activity {
    private String url;
    private TextView desc;
    private TextView sTitle;
    private Button button;
    private Context mContext;
    private ImageView imageView;

    private TaskListener<DynamicContentResponse<Statutes>>
cmsTaskListener =
        new TaskListener<DynamicContentResponse<Statutes>>() {
            @Override
            public void
onTaskSuccess(DynamicContentResponse<Statutes> result) {
                if (result != null && result.getContent() != null
&& result.getContent().size() > 0)
                {
                    List<Statutes> statuteList =
result.getContent();

                    Statutes statute = statuteList.get(0);

                    url = statute.getStatutes_url();
                    desc.setText(statute.getDescription());
                    sTitle.setText(statute.getTitle());

                    String imageUrl =

```

```

statute.getImage().getImageForDensity(CRImage.IMAGE_TYPE.SMALL,
                                       (Activity)mContext);

        AQuery aq = new AQuery(mContext);
        Bitmap preset = aq.getCachedImage(imageUrl);

        if (preset != null) {
            aq.id(imageView).image(imageUrl, true,
true, 0, 0, preset, AQuery.FADE_IN);
        } else {
            aq.id(imageView).image(imageUrl, true,
true, 0, 0);
        }
    }

    @Override
    public void onTaskFailure(Exception e) {

    }

    @Override
    public void onTaskCancelled() {

    }

    @Override
    public void onTaskTimeOut() {

    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_foiastatutes);

    mContext = (Context)this;

    Bundle bundle = getIntent().getExtras();
    String actionObject = bundle.getString("action_object");

    Typeface light = Typeface.createFromAsset(getAssets(),
"nexa_light.otf");
    Typeface bold =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");

    desc = (TextView)findViewById(R.id.statutes_desc);

```

```

        sTitle = (TextView)findViewById(R.id.statutes_title);
        button = (Button)findViewById(R.id.statutes_button);
        imageView = (ImageView)findViewById(R.id.statutes_img);

        desc.setTypeface(light);
        button.setTypeface(bold);
        sTitle.setTypeface(bold);

        Type adsType = new
TypeToken<DynamicContentResponse<Statutes>>() {}.getType();
        GetDynamicContent<DynamicContentResponse<Statutes>>
advertisementsTask =
            new
DynamicContentTaskBuilder<DynamicContentResponse<Statutes>>()
                .withContext(this)
                .withMethod(HttpGet.METHOD_NAME)
                .withPermanentLink(actionObject)
                .withTaskListener(cmsTaskListener)
                .withType(adsType)
                .build();
        advertisementsTask.execute();
    }

    public void onClick(View view)
    {
        Intent webViewIntent = new
Intent(FOIAStatutes.this,FOIAWebView.class);

        Bundle bundle = new Bundle();

        bundle.putString(CRConstants.EXTRAS_KEY_URL, url);
        bundle.putString(CRConstants.EXTRAS_KEY_TITLE,"FOIA
Statutes");
        webViewIntent.putExtras(bundle);

        startActivity(webViewIntent);
    }

    public void goBack(View v)
    {
        this.finish();
    }
}

package gov.dhs.mobile.efoia;

import android.app.Activity;
import android.content.Context;
import android.content.SharedPreferences;

```

```

import android.os.Bundle;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.content.Intent;

import android.widget.ListView;
import android.widget.RelativeLayout;
import android.widget.Toast;

import java.util.List;
import java.util.Random;

import XXXXXXXXXXXX.XXXXXXXXXXXXXConnect;
import XXXXXXXXXXXX.models.Advertisement;
import XXXXXXXXXXXX.models.CRImage;
import XXXXXXXXXXXX.models.json.response.AdvertisementResponse;
import XXXXXXXXXXXX.network.request.TaskListener;
import XXXXXXXXXXXX.models.MobileMenu;
import XXXXXXXXXXXX.models.json.response.MobileMenuResponse;
import gov.dhs.mobile.efoia.R;
import gov.dhs.mobile.efoia.tasks.AdvertisementTask;
import gov.dhs.mobile.efoia.tasks.GetBackgroundImageTask;
import gov.dhs.mobile.efoia.viewcontrollers.events.EventsActivity;
import
gov.dhs.mobile.efoia.viewcontrollers.exemptions.ExemptionsActivity;
import gov.dhs.mobile.efoia.viewcontrollers.howto.FOIAHowToActivity;
import gov.dhs.mobile.efoia.viewcontrollers.library.FOIALibrary;
import
gov.dhs.mobile.efoia.viewcontrollers.privacypolicy.PrivacyPolicyActivi
ty;
import
gov.dhs.mobile.efoia.viewcontrollers.privacystatement.PrivacyStatement
Activity;
import gov.dhs.mobile.efoia.viewcontrollers.status.FOIAStatus;
import gov.dhs.mobile.efoia.viewcontrollers.statutes.FOIAStatutes;
import gov.dhs.mobile.efoia.tasks.GetMobileMenusTask;
import gov.dhs.mobile.efoia.arrayadapters.HomeViewTableArrayAdapter;

/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains

```

```

* | the property of XXXXXXXXXXXX and its suppliers,
* | if any. The intellectual and technical concepts contained
* | herein are proprietary to XXXXXXXXXXXX
* | and its suppliers and may be covered by U.S. and Foreign Patents,
* | patents in process, and are protected by trade secret or
copyright law.
* | Dissemination of this information or reproduction of this
material
* | is strictly forbidden unless prior written permission is obtained
* | from XXXXXXXXXXXX.
* |
*****
* | <Insert your Javadoc Here>
* |
*****
*/
public class MainActivity extends Activity {

    private ListView activityListView;
    private List<MobileMenu> menus;
    private List<Advertisement> advertisementList;
    private Context mContext = this;
    private GestureDetector gestureDetector;
    private int currentIndex;
    private RelativeLayout layout;
    private HomeViewTableArrayAdapter adapter;
    public static final String PREF_FILE_NAME = "eFOIA-Prefs";
    public static final String POLICY_ACCEPTED = "accepted";
    public static int mSelectedItem=-1;

    private TaskListener<AdvertisementResponse>
advertisementTaskListener = new TaskListener<AdvertisementResponse>()
{
    @Override
    public void onTaskSuccess(AdvertisementResponse result) {
        if (result != null && result.getAdvertisements() != null
&& result.getAdvertisements().size() > 0) {
            advertisementList = result.getAdvertisements();

            gestureDetector = new GestureDetector(mContext, new
GestureListener());

            layout.setOnTouchListener(new View.OnTouchListener() {
                @Override
                public boolean onTouch(View view, MotionEvent
motionEvent) {
                    return
gestureDetector.onTouchEvent(motionEvent);
                }
            });
        }
    }
};
}

```



```

        });

        Random rand = new Random();

        rand.setSeed(advertisementList.size());

        currentIndex =
        rand.nextInt(advertisementList.size());

        Advertisement ad =
        advertisementList.get(currentIndex);

        String imageUrl =

        ad.getImage().getCRIImage().getImageForDensity(CRIImage.IMAGE_TYPE.FULLS
        CREEN,

                                (Activity) mContext);

        try
        {
            layout.setBackground(new
        GetBackgroundImageTask().execute(imageUrl).get());
        }
        catch (Exception e)
        {
        }
    }
}

@Override
public void onTaskFailure(Exception e) {

}

@Override
public void onTaskCancelled() {

}

@Override
public void onTaskTimeOut() {

}

};

private TaskListener<MobileMenuResponse> mobileMenusTaskCallback =
new TaskListener<MobileMenuResponse>() {
    @Override
    public void onTaskSuccess(MobileMenuResponse result) {

```

```
        if (result != null && result.getMobileMenus() != null &&
result.getMobileMenus().size() > 0) {
```

```
            menus = result.getMobileMenus();
            adapter = new
HomeViewTableAdapter(MainActivity.this,
                R.layout.home_table_list_item, menus);
            activityListView.setAdapter(adapter);
```

```
            SharedPreferences prefs =
getSharedPreferences(PREF_FILE_NAME,MODE_PRIVATE);
```

```
            boolean accepted =
prefs.getBoolean(POLICY_ACCEPTED,false);
```

```
            if (!accepted)
            {
                Intent intent = new Intent(MainActivity.this,
PrivacyPolicyActivity.class);
                Bundle bundle = new Bundle();
                bundle.putString("action_object",
"privacy_policy");
                bundle.putBoolean("first_launch",true);
                intent.putExtras(bundle);
                startActivity(intent);
            }
        }
    }
```

```
        @Override
        public void onTaskFailure(Exception e) {
            Toast.makeText(MainActivity.this, "Error: " +
e.getLocalizedMessage(), Toast.LENGTH_SHORT).show();
        }
```

```
        @Override
        public void onTaskCancelled() {

        }
```

```
        @Override
        public void onTaskTimeOut() {

        }
```

```
    };
```

```
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    new GetMobileMenusTask(MainActivity.this,
mobileMenusTaskCallback).execute();
    new AdvertisementTask(MainActivity.this,
advertisementTaskListener).execute();

    activityListView = (ListView)
findViewById(R.id.activity_main_lv);
    layout = (RelativeLayout) ((Activity)
mContext).findViewById(R.id.relativeLayout);

    activityListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {

            MobileMenu menu = menus.get(position);

            mSelectedItem = position;
            adapter.notifyDataSetChanged();

            if (menu.getAction().equalsIgnoreCase("subcategory"))
{
                launchSubMenu(menu, menu.getTitle());

            } else if
(menu.getAction().equalsIgnoreCase("dynamic_contents")) {
                launchContentScreen((String)
menu.getActionObject());
            } else if
(menu.getAction().equalsIgnoreCase("tagged_contents")) {
                //Surveys use tagged contents
            } else if
(menu.getAction().equalsIgnoreCase("no_action")) {
                launchAction(menu);
            }
        }
    });
}
}

```

```

private void launchContentScreen(String action) {
    Intent intent = null;

    Bundle bundle = new Bundle();

    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this,"Main Menu",action,"","");

    if (action.equalsIgnoreCase("privacy_policy")) {
        intent = new Intent(MainActivity.this,
PrivacyPolicyActivity.class);
    } else if (action.equalsIgnoreCase("library")) {
        intent = new Intent(MainActivity.this, FOIALibrary.class);

    } else if (action.equalsIgnoreCase("statutes")) {
        intent = new Intent(MainActivity.this,
FOIAStatutes.class);
    } else if (action.equalsIgnoreCase("events")) {
        intent = new Intent(MainActivity.this,
EventsActivity.class);

    } else if (action.equalsIgnoreCase("exemptions_exclusions")) {
        intent = new Intent(MainActivity.this,
ExemptionsActivity.class);
    } else if (action.equalsIgnoreCase("foia_status")) {
        intent = new Intent(MainActivity.this, FOIAStatus.class);
    }

    if (intent != null) {
        bundle.putString("action_object", action);
        intent.putExtras(bundle);
        startActivity(intent);
    }
}

private void launchSubMenu(MobileMenu menu, String title)
{
    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this,"Main Menu",title,"","");

    if (title.equalsIgnoreCase("How to Submit a FOIA Request")) {
        Intent howtoIntent = new Intent(MainActivity.this,
FOIAHowToActivity.class);
        Bundle bundle = new Bundle();

        bundle.putInt("menu_id", (int) menu.getId());
        howtoIntent.putExtras(bundle);
    }
}

```

```

        startActivity(howtoIntent);
    }
}

private void launchAction(MobileMenu menu)
{
    XXXXXXXXXXXXXXXConnect sdk = XXXXXXXXXXXXXXXConnect.getInstance();

    sdk.trackInsightV2(this,"Main Menu",menu.getAction(),"","");

    //Submission starts with a privacy policy
    if (menu.getAction().equalsIgnoreCase("no_action") &&
        menu.getActionObject().equals("privacy_policy")) {
        Intent intent = new Intent(MainActivity.this,
PrivacyStatementActivity.class);

        startActivity(intent);
    }
}

final class GestureListener extends
GestureDetector.SimpleOnGestureListener {

    private static final int SWIPE_THRESHOLD = 100;
    private static final int SWIPE_VELOCITY_THRESHOLD = 100;

    @Override
    public boolean onDown(MotionEvent e) {
        return true;
    }

    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float
velocityX, float velocityY) {
        boolean result = false;
        try {
            float diffY = e2.getY() - e1.getY();
            float diffX = e2.getX() - e1.getX();
            if (Math.abs(diffX) > Math.abs(diffY)) {
                if (Math.abs(diffX) > SWIPE_THRESHOLD &&
Math.abs(velocityX) > SWIPE_VELOCITY_THRESHOLD) {
                    if (diffX > 0) {
                        onSwipeRight();
                    } else {
                        onSwipeLeft();
                    }
                }
            }
        }
    }
}

```

```

        }
        result = true;
    } else if (Math.abs(diffY) > SWIPE_THRESHOLD &&
Math.abs(velocityY) > SWIPE_VELOCITY_THRESHOLD) {
        if (diffY > 0) {
            onSwipeBottom();
        } else {
            onSwipeTop();
        }
    }
    result = true;
} catch (Exception exception) {
    exception.printStackTrace();
}
return result;
}

public void onSwipeRight() {
    currentIndex--;

    if (currentIndex-- < 0) {
        currentIndex = advertisementList.size() - 1;
    }

    Advertisement ad =
advertisementList.get(currentIndex--);

    String imageUrl =

ad.getImage().getCRIImage().getImageForDensity(CRIImage.IMAGE_TYPE.FULLS
CREEN,

                (Activity) mContext);

    try
    {
        layout.setBackground(new
GetBackgroundImageTask().execute(imageUrl).get());
    }
    catch (Exception e)
    {

    }

}
}

```

```

        public void onSwipeLeft() {
            currentImageIndex++;

            if (currentImageIndex == advertisementList.size()) {
                currentImageIndex = 0;
            }

            Advertisement ad =
advertisementList.get(currentImageIndex);

            String imageUrl =

ad.getImage().getCRIImage().getImageForDensity(CRIImage.IMAGE_TYPE.FULLS
CREEN,
                                                    (Activity) mContext);

            try
            {
                layout.setBackground(new
GetBackgroundImageTask().execute(imageUrl).get());
            }
            catch (Exception e)
            {

            }

        }

        public void onSwipeTop() {

        }

        public void onSwipeBottom() {

        }

    }

}
package gov.dhs.mobile.efoia.datastructures;

/*****
| DHS e-FOIA
| *****/
|
| *****/
| Copyright (c) XXXXXXXXXXXX [2012] – 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains

```

| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.

| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.

*****/

```
public class CheckStatus {

    private String foia_request_number;
    private String drupalFormID;

    public CheckStatus()
    {
        foia_request_number="";
        drupalFormID="";
    }

    public String serializeSubmission(String boundary)
    {
        StringBuilder body = new StringBuilder();

        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"foia_request_number\"\r\n\r\n"+foia_request_number);

        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=\"op\"\r\n\r
\nCheck Status");

        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"form_build_id\"\r\n\r\n"+drupalFormID);

        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=\"form_id
\"\r\n\r\n\"custom_foia_request_status_form");

        body.append("\r\n--"+boundary+"\r\n");

        return body.toString();
    }
}
```



```

    public String getFoia_request_number() {
        return foia_request_number;
    }

    public void setFoia_request_number(String foia_request_number) {
        this.foia_request_number = foia_request_number;
    }

    public String getDrupalFormID() {
        return drupalFormID;
    }

    public void setDrupalFormID(String drupalFormID) {
        this.drupalFormID = drupalFormID;
    }
}

package gov.dhs.mobile.efoia.datastructures;

import android.os.Parcel;
import android.os.Parcelable;

/*****
| DHS e-FOIA
|
|*****
| Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
|
| NOTICE: All information contained herein is, and remains
| the property of XXXXXXXXXXXX and its suppliers,
| if any. The intellectual and technical concepts contained
| herein are proprietary to XXXXXXXXXXXX
| and its suppliers and may be covered by U.S. and Foreign Patents,
| patents in process, and are protected by trade secret or copyright
law.
| Dissemination of this information or reproduction of this material
| is strictly forbidden unless prior written permission is obtained
| from XXXXXXXXXXXX.
|
|*****/
public class Submission implements Parcelable {

    private String component;
    private String title;
    private String firstName;
    private String middleInitial;
    private String lastName;
    private String suffix;
    private String emailAddress;
    private String country;

```

```
private String addressLine1;
private String addressLine2;
private String state;
private String city;
private String zipCode;
private String telephone;
private String fax;
private Boolean recordsForSelf;
private Boolean declaration;
private String initials;
private String recordsDescription;
private String payAmount;
private String requestCategory;
private Boolean requestFeeWaiver;
private String feeWaiverDescription;
private String expediteRequested;
private String expediteRequestedDescription;
private String drupalFormID;
```

```
public Submission()
{
    component="";
    title="";
    firstName="";
    middleInitial="";
    lastName="";
    suffix="";
    emailAddress="";
    country="";
    addressLine1="";
    addressLine2="";
    state="";
    city="";
    zipCode="";
    telephone="";
    fax="";
    recordsForSelf=false;
    declaration=false;
    initials="";
    recordsDescription="";
    payAmount="";
    requestCategory="";
    requestFeeWaiver=false;
    feeWaiverDescription="";
    expediteRequested="";
    expediteRequestedDescription="";
    drupalFormID="";
}
```

```
public String getComponent() {
```

```
        return component;
    }

    public void setComponent(String component) {
        this.component = component;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getMiddleInitial() {
        return middleInitial;
    }

    public void setMiddleInitial(String middleInitial) {
        this.middleInitial = middleInitial;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getSuffix() {
        return suffix;
    }

    public void setSuffix(String suffix) {
        this.suffix = suffix;
    }

    public String getEmailAddress() {
        return emailAddress;
    }
}
```

```
public void setEmailAddress(String emailAddress) {
    this.emailAddress = emailAddress;
}

public String getCountry() {
    return country;
}

public void setCountry(String country) {
    this.country = country;
}

public String getAddressLine1() {
    return addressLine1;
}

public void setAddressLine1(String addressLine1) {
    this.addressLine1 = addressLine1;
}

public String getAddressLine2() {
    return addressLine2;
}

public void setAddressLine2(String addressLine2) {
    this.addressLine2 = addressLine2;
}

public String getState() {
    return state;
}

public void setState(String state) {
    this.state = state;
}

public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
}

public String getZipCode() {
    return zipCode;
}

public void setZipCode(String zipCode) {
```

```
    this.zipCode = zipCode;
}

public String getTelephone() {
    return telephone;
}

public void setTelephone(String telephone) {
    this.telephone = telephone;
}

public String getFax() {
    return fax;
}

public void setFax(String fax) {
    this.fax = fax;
}

public boolean getRecordsForSelf() {
    return recordsForSelf;
}

public void setRecordsForSelf(boolean recordsForSelf) {
    this.recordsForSelf = recordsForSelf;
}

public boolean getDeclaration() {
    return declaration;
}

public void setDeclaration(boolean declaration) {
    this.declaration = declaration;
}

public String getInitials() {
    return initials;
}

public void setInitials(String initials) {
    this.initials = initials;
}

public String getRecordsDescription() {
    return recordsDescription;
}

public void setRecordsDescription(String recordsDescription) {
    this.recordsDescription = recordsDescription;
}
```

```
public String getPayAmount() {
    return payAmount;
}

public void setPayAmount(String payAmount) {
    this.payAmount = payAmount;
}

public String getRequestCategory() {
    return requestCategory;
}

public void setRequestCategory(String requestCategory) {
    this.requestCategory = requestCategory;
}

public boolean getRequestFeeWaiver() {
    return requestFeeWaiver;
}

public void setRequestFeeWaiver(boolean requestFeeWaiver) {
    this.requestFeeWaiver = requestFeeWaiver;
}

public String getFeeWaiverDescription() {
    return feeWaiverDescription;
}

public void setFeeWaiverDescription(String feeWaiverDescription) {
    this.feeWaiverDescription = feeWaiverDescription;
}

public String getExpediteRequested() {
    return expediteRequested;
}

public void setExpediteRequested(String expediteRequested) {
    this.expediteRequested = expediteRequested;
}

public String getExpediteRequestedDescription() {
    return expediteRequestedDescription;
}

public void setExpediteRequestedDescription(String
expediteRequestedDescription) {
    this.expediteRequestedDescription =
expediteRequestedDescription;
}
```

```

public String getDrupalFormID() {
    return drupalFormID;
}

public void setDrupalFormID(String drupalFormID) {
    this.drupalFormID = drupalFormID;
}

public String serializeSubmission(String boundary)
{
    StringBuilder body = new StringBuilder();

    body.append("\r\n--"+boundary+"\r\n");

    body.append("Content-Disposition: form-data; name=
\"submitted[dhs_component]\"\\r\\n\\r\\n"+component);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[title]\"\\r\\n\\r\\n"+title);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[first_name]\"\\r\\n\\r\\n"+firstName);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[middle_initial]\"\\r\\n\\r\\n"+middleInitial);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[last_name]\"\\r\\n\\r\\n"+lastName);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[suffix]\"\\r\\n\\r\\n"+suffix);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[email_address]\"\\r\\n\\r\\n"+emailAddress);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[country]\"\\r\\n\\r\\n"+country);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[address_1]\"\\r\\n\\r\\n"+addressLine1);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[address_2]\"\\r\\n\\r\\n"+addressLine2);
    body.append("\r\n--"+boundary+"\r\n");
    body.append("Content-Disposition: form-data; name=
\"submitted[city]\"\\r\\n\\r\\n"+city);
    body.append("\r\n--"+boundary+"\r\n");
}

```

```

        body.append("Content-Disposition: form-data; name=
\"submitted[state]\"\\r\\n\\r\\n"+state);
        body.append("\\r\\n--"+boundary+"\\r\\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[zip_code]\"\\r\\n\\r\\n"+zipCode);
        body.append("\\r\\n--"+boundary+"\\r\\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[telephone_number]\"\\r\\n\\r\\n"+telephone);
        body.append("\\r\\n--"+boundary+"\\r\\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[fax_number]\"\\r\\n\\r\\n"+fax);
        body.append("\\r\\n--"+boundary+"\\r\\n");

        String boolString ="No ";
        if (recordsForSelf)
        {
            boolString = "Yes ";
        }

        body.append("Content-Disposition: form-data; name=
\"submitted[records_request_self]\"\\r\\n\\r\\n"+boolString);
        body.append("\\r\\n--"+boundary+"\\r\\n");

        boolString ="";
        if (declaration)
        {
            boolString = "Yes ";
        }

        body.append("Content-Disposition: form-data; name=
\"submitted[perjury_statement]\"\\r\\n\\r\\n"+boolString);
        body.append("\\r\\n--"+boundary+"\\r\\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[electronic_signature]\"\\r\\n\\r\\n"+initials);
        body.append("\\r\\n--"+boundary+"\\r\\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[amount_willing_to_pay]\"\\r\\n\\r\\n"+payAmount);
        body.append("\\r\\n--"+boundary+"\\r\\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[requester_category]\"\\r\\n\\r\\n"+requestCategory);
        body.append("\\r\\n--"+boundary+"\\r\\n");

        boolString ="";
        if (declaration)
        {
            boolString = "fee waiver request ";
        }

        body.append("Content-Disposition: form-data; name=
\"submitted[fee_waiver_request]\"\\r\\n\\r\\n"+boolString);

```



```

        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[fee_waiver_explanation]\"\r\n\r\n"+feeWaiverDescription);
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[expedite_selection]\"\r\n\r\n"+expediteRequested);
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"submitted[additional_expedite_information]\"\r\n\r\n
\n"+expediteRequestedDescription);
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"details[sid]\"");
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"details[page_num]\"\r\n\r\n1");
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"details[page_count]\"\r\n\r\n1");
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=
\"details[finished]\"\r\n\r\n0");
        body.append("\r\n--" + boundary + "\r\n");
        body.append("Content-Disposition: form-data; name=
\"form_build_id\"\r\n\r\n"+drupalFormID);
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=\"form_id
\r\n\r\nwebform_client_form_11359");
        body.append("\r\n--"+boundary+"\r\n");
        body.append("Content-Disposition: form-data; name=\"op\"\r\n\r\n
\nSubmit FOIA Request");
        body.append("\r\n--"+boundary+"\r\n");

```

```

    }
    return body.toString();
}

```

```

@Override
public int describeContents() {
    return 8675309;
}

```

```

@Override
public void writeToParcel(Parcel dest, int flags)
{
    dest.writeString(component);
    dest.writeString(title);
    dest.writeString(firstName);
    dest.writeString(middleInitial);
    dest.writeString(lastName);
}

```

```

dest.writeString(suffix);
dest.writeString(emailAddress);
dest.writeString(country);
dest.writeString(addressLine1);
dest.writeString(addressLine2);
dest.writeString(state);
dest.writeString(city);
dest.writeString(zipCode);
dest.writeString(telephone);
dest.writeString(fax);
dest.writeByte((byte) (recordsForSelf ? 1 : 0));
dest.writeByte((byte) (declaration ? 1 : 0));
dest.writeString(initials);
dest.writeString(recordsDescription);
dest.writeString(payAmount);
dest.writeString(requestCategory);
dest.writeByte((byte) (requestFeeWaiver ? 1 : 0));
dest.writeString(feeWaiverDescription);
dest.writeString(expediteRequested);
dest.writeString(expediteRequestedDescription);
dest.writeString(drupalFormID);
}

```

```

public static final Parcelable.Creator<Submission> CREATOR = new
Parcelable.Creator<Submission>() {
    @Override
    public Submission createFromParcel(Parcel source) {
        Submission submission = new Submission();

        submission.setComponent(source.readString());
        submission.setTitle(source.readString());
        submission.setFirstName(source.readString());
        submission.setMiddleInitial(source.readString());
        submission.setLastName(source.readString());
        submission.setSuffix(source.readString());
        submission.setEmailAddress(source.readString());
        submission.setCountry(source.readString());
        submission.setAddressLine1(source.readString());
        submission.setAddressLine2(source.readString());
        submission.setState(source.readString());
        submission.setCity(source.readString());
        submission.setZipCode(source.readString());
        submission.setTelephone(source.readString());
        submission.setFax(source.readString());
        submission.setRecordsForSelf(source.readByte() != 0);
        submission.setDeclaration(source.readByte() != 0);
        submission.setInitials(source.readString());
        submission.setRecordsDescription(source.readString());
        submission.setPayAmount(source.readString());
        submission.setRequestCategory(source.readString());
    }
}

```

```

        submission.setRequestFeeWaiver(source.readByte() != 0);
        submission.setFeeWaiverDescription(source.readString());
        submission.setExpediteRequested(source.readString());
        submission.setExpediteRequested(source.readString());
        submission.setDrupalFormID(source.readString());

        return submission;
    }

    @Override
    public Submission[] newArray(int size) {
        return new Submission[size];
    }
};

}

package gov.dhs.mobile.efoia;

import android.os.Bundle;
import com.crashlytics.android.Crashlytics;
import XXXXXXXXXXXXXXXX.ui.activities.CRSplashActivity;
import gov.dhs.mobile.efoia.BuildConfig;
import gov.dhs.mobile.efoia.R;

import io.fabric.sdk.android.Fabric;

/**
 * *****
 * | EventsFargo
 * | *****
 * | Created byXXXXXXXXXXon 5/5/15.
 * |
 *****
 * | Copyright (c) XXXXXXXXXXXXX [2012] - 2015. All Rights Reserved.
 * |
 * | NOTICE: All information contained herein is, and remains
 * | the property of XXXXXXXXXXXXX and its suppliers,
 * | if any. The intellectual and technical concepts contained
 * | herein are proprietary to XXXXXXXXXXXXX
 * | and its suppliers and may be covered by U.S. and Foreign Patents,
 * | patents in process, and are protected by trade secret or
copyright law.
 * | Dissemination of this information or reproduction of this
material
 * | is strictly forbidden unless prior written permission is obtained
 * | from XXXXXXXXXXXXX.
 * |
 *****
 * | <Insert your Javadoc Here>

```

```

* |
*****
*/
public class HeroActivity extends CRSplashActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Fabric.with(this, new Crashlytics());
    }
    /*
    protected String getGCMKey() {
        return BuildConfig.GCM_KEY;
    }
    */
    protected String getServerUrl() {
        return BuildConfig.CR_URL;
    }

    @Override
    protected Class<?> getMainClassActivity() {
        return MainActivity.class;
    }

    protected int getHeroScreenResource() {
        return R.drawable.hero;
    }
}

package gov.dhs.mobile.efoia.webview;

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;

import XXXXXXXXXXXX.CRConstants;
import gov.dhs.mobile.efoia.R;
/**
 * *****
 * | DHS e-FOIA
 * |*****
 * | Created byXXXXXXXXXXon 5/6/15.
 * |
*****
 * | Copyright (c) XXXXXXXXXXXX [2012] - 2015. All Rights Reserved.

```

```

* |
* | NOTICE: All information contained herein is, and remains
* | the property of XXXXXXXXXXXXX and its suppliers,
* | if any. The intellectual and technical concepts contained
* | herein are proprietary to XXXXXXXXXXXXX
* | and its suppliers and may be covered by U.S. and Foreign Patents,
* | patents in process, and are protected by trade secret or
copyright law.
* | Dissemination of this information or reproduction of this
material
* | is strictly forbidden unless prior written permission is obtained
* | from XXXXXXXXXXXXX.
* |
*****
* | <Insert your Javadoc Here>
* |
*****
*/
public class FOIAWebView extends Activity
{
    private String url;
    private String title;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_webview);

        WebView webView = (WebView)findViewById(R.id.webView);
        TextView titleView =
(TextView)findViewById(R.id.webview_title);

        webView.setWebViewClient(new WebViewClient());
        webView.getSettings().setBuiltInZoomControls(true);

        Typeface font =
Typeface.createFromAsset(getAssets(),"nexa_bold.otf");
        titleView.setTypeface(font);

        Bundle bundle = getIntent().getExtras();
        url = bundle.getString(CRConstants.EXTRAS_KEY_URL);
        title = bundle.getString(CRConstants.EXTRAS_KEY_TITLE);

        webView.getSettings().setUseWideViewPort(true);
        webView.getSettings().setLoadWithOverviewMode(true);

        titleView.setText(title);
        webView.loadUrl(url);
    }
}

```

```
public void goBack(View v)
{
    this.finish();
}
}
```