**Editing The Manifest**

<**activity**

android:name="ah.hathi.simpleweather.WeatherActivity"

android:label="@string/app\_name"

android:screenOrientation="portrait"

>

<**intent-filter**>

<**action** android:name="android.intent.action.MAIN" />

<**category** android:name="android.intent.category.LAUNCHER" />

</**intent-filter**>

</**activity**>

<**uses-permission** android:name="android.permission.INTERNET"/> // Permission to access internet

**Activity’s Layout**

<**FrameLayout** xmlns:android="<http://schemas.android.com/apk/res/android>"

xmlns:tools="<http://schemas.android.com/tools>"

android:id="@+id/container"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="ah.hathi.simpleweather.WeatherActivity"

tools:ignore="MergeRootFrame"

android:background="#FF0099CC" />

**Strings.xml**

<?**xml** version="1.0" encoding="utf-8"?><**resources**>

<**string** name="app\_name">Simple Weather</**string**>

<**string** name="change\_city">Şehir Değiştir</**string**>

<**string** name="open\_weather\_maps\_app\_id">Weather App</**string**>

<**string** name="weather\_sunny">&#xf00d;</**string**>

<**string** name="weather\_clear\_night">&#xf02e;</**string**>

<**string** name="weather\_foggy">&#xf014;</**string**>

<**string** name="weather\_cloudy">&#xf013;</**string**>

<**string** name="weather\_rainy">&#xf019;</**string**>

<**string** name="weather\_snowy">&#xf01b;</**string**>

<**string** name="weather\_thunder">&#xf01e;</**string**>

<**string** name="weather\_drizzle">&#xf01c;</**string**>

<**string** name="place\_not\_found">Üzgünüz.Hava durumu

bulunamadı</**string**> </**resources**>

**Adding a menu item**

<**menu** xmlns:android="<http://schemas.android.com/apk/res/android>"

xmlns:app="<http://schemas.android.com/apk/res-auto>"

xmlns:tools="<http://schemas.android.com/tools>"

tools:context="ah.hathi.simpleweather.WeatherActivity" >

<**item**

android:id="@+id/change\_city"

android:orderInCategory="1"

android:title="@string/change\_city"

app:showAsAction="never"/

</**menu**>

**Fetch Data From OpenWeatherApp**

{

"base": "cmc stations",

"clouds": {

"all": 90 },

"cod": 200,

"coord": {

"lat": -35.28,

"lon": 149.13 },

"dt": 1404390600,

"id": 2172517,

"main": {

"humidity": 100,

"pressure": 1023,

"temp": -1,

"temp\_max": -1,

"temp\_min": -1 },

"name": "Canberra", "sys": {

"country": "AU",

"message": 0.313,

"sunrise": 1404335563,

"sunset": 1404370965 },

"weather": [

{ "description": "overcast clouds",

"icon": "04n",

"id": 804,

"main": "Clouds } ], "wind": { "deg": 305.004, "speed": 1.07

}}

**Storing the City**

**package** ah.hathi.simpleweather;

**import** android.app.Activity;

**import** android.content.SharedPreferences;

**public** **class** CityPreference {

SharedPreferences prefs;

**public** CityPreference(Activity activity){

prefs = activity.getPreferences(Activity.MODE\_PRIVATE);

}

String getCity(){

**return** prefs.getString("city", "Istanbul, TR");

}

**void** setCity(String city){

prefs.edit().putString("city", city).commit();

}}

**Fragment**

**public** **class** WeatherFragment **extends** Fragment {

Typeface weatherFont;

TextView cityField;

TextView updatedField;

TextView detailsField;

TextView currentTemperatureField;

TextView weatherIcon;

Handler handler;

**public** WeatherFragment(){

handler = **new** Handler();

}

@Override

**public** View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View rootView = inflater.inflate(R.layout.fragment\_weather, container, **false**);

cityField = (TextView)rootView.findViewById(R.id.city\_field);

updatedField = (TextView)rootView.findViewById(R.id.updated\_field);

detailsField = (TextView)rootView.findViewById(R.id.details\_field);

currentTemperatureField = (TextView)rootView.findViewById(R.id.current\_temperature\_field);

weatherIcon = (TextView)rootView.findViewById(R.id.weather\_icon);

weatherIcon.setTypeface(weatherFont);

**return** rootView;

}}

**Updating Weather Data**

**private** **void** updateWeatherData(**final** String city){

**new** Thread(){

**public** **void** run(){

**final** JSONObject json = RemoteFetch.getJSON(getActivity(), city);

**if**(json == **null**){

handler.post(**new** Runnable(){

**public** **void** run(){

Toast.makeText(getActivity(),

getActivity().getString(R.string.place\_not\_found),

Toast.LENGTH\_LONG).show(); });} **else** {handler.post(**new** Runnable(){ **public** **void** run(){renderWeather(json); } }); }}}.start();

}

**RenderWeather**

**private** **void** renderWeather(JSONObject json){

**try** {

cityField.setText(json.getString("name").toUpperCase(Locale.US) +

", " +

json.getJSONObject("sys").getString("country"));

JSONObject details = json.getJSONArray("weather").getJSONObject(0);

JSONObject main = json.getJSONObject("main");

detailsField.setText(

details.getString("description").toUpperCase(Locale.US) +

"\n" + "Humidity: " + main.getString("humidity") + "%" +

"\n" + "Pressure: " + main.getString("pressure") + " hPa");

currentTemperatureField.setText(

String.format("%.2f", main.getDouble("temp"))+ " ℃");

DateFormat df = DateFormat.getDateTimeInstance();

String updatedOn = df.format(**new** Date(json.getLong("dt")\*1000));

updatedField.setText("Last update: " + updatedOn);

setWeatherIcon(details.getInt("id"),

json.getJSONObject("sys").getLong("sunrise") \* 1000,

json.getJSONObject("sys").getLong("sunset") \* 1000);

}**catch**(Exception e){

Log.e("SimpleWeather", "One or more fields not found in the JSON data");

}

}

**Weather Icon**

**private** **void** setWeatherIcon(**int** actualId, **long** sunrise, **long** sunset){

**int** id = actualId / 100;

String icon = "";

**if**(actualId == 800){**long** currentTime = **new** Date().getTime();

**if**(currentTime>=sunrise && currentTime<sunset) {

icon = getActivity().getString(R.string.weather\_sunny);

} **else** {

icon = getActivity().getString(R.string.weather\_clear\_night); } } **else** {

**switch**(id) {

**case** 2 : icon = getActivity().getString(R.string.weather\_thunder); **break**;

**case** 3 : icon = getActivity().getString(R.string.weather\_drizzle); **break**;

**case** 7 : icon = getActivity().getString(R.string.weather\_foggy); **break**;

**case** 8 : icon = getActivity().getString(R.string.weather\_cloudy); **break**;

**case** 6 : icon = getActivity().getString(R.string.weather\_snowy); **break**;

**case** 5 : icon = getActivity().getString(R.string.weather\_rainy); **break**;

}

}

weatherIcon.setText(icon);

}

**Changing City’s Name**

**public** **void** changeCity(String city){

updateWeatherData(city);}

**ChangeCity Method**

@Override

**public** **boolean** onOptionsItemSelected(MenuItem item) {

**if**(item.getItemId() == R.id.change\_city){

showInputDialog() }

**return** **false**;

}**private** **void** showInputDialog(){

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

builder.setTitle("Change city");

**final** EditText input = **new** EditText(**this**);

input.setInputType(InputType.TYPE\_CLASS\_TEXT);

builder.setView(input);

builder.setPositiveButton("Go", **new** DialogInterface.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

changeCity(input.getText().toString());

}

});

builder.show();

}**public** **void** changeCity(String city){

WeatherFragment wf = (WeatherFragment)getSupportFragmentManager()

.findFragmentById(R.id.container);

wf.changeCity(city);

**new** CityPreference(**this**).setCity(city);

}