ALGONQUIN COLLEGE

CST2335 GRAPHICAL INTERFACE PROGRAMMING

Week 10

Getting data from servers

JSON

- JSON stands for **J**ava**S**cript **O**bject **N**otation.
- Objects are surrounded by curly braces { }, and inside are name/value pairs separated by a colon (:)

```
{
    "firstName" : "Eric",
    "lastName" : "Torunski"
}
```

JSON

- The first element (key) is always a string.
- The value however can be of different data types namely: **string**, **integer**, **long**, **double**, **boolean** and **array**

```
{
    "firstName" : "Eric",
    "lastName" : "Torunski"
}
```

Functions to retrieve JSON data types

- {"firstName": "Eric"} - use getString("firstName")
- {"age": 20} - use **getInt**("age")
- {"age": 20} - use **getLong**(“age")
- {"height": 5.5} - use **getDouble**("height")
- {"active": false} - use **getBoolean**(“active")
- {"team": [{"name": "Harry Kane"}, {"name" : "Jordan Henderson"}]} - use **getJSONArray**("team")

Connecting to a Web Server

- One of the problems with connecting to a web server is that it can take a lot of time to get data back.
- If you do this from the GUI code, it can make your GUI freeze until the data has finished downloading.
- Hence, network connections must be done on a background thread, meaning that the computation is done on a second processor within your phone.

Connecting to a Web Server

- There are several libraries available to help with network communications namely:
- Volley
- OkHttp
- Retrofit

We will be using Volley in this course but the others work in similar ways.

Setting up Volley

- You will be sending data over the internet, which might cost the user money from their data plan, so you have to ask for Internet permissions in the AndroidManifest.xml. Add this line to the manifest
  <uses-permission
  android:name="android.permission.INTERNET"/>

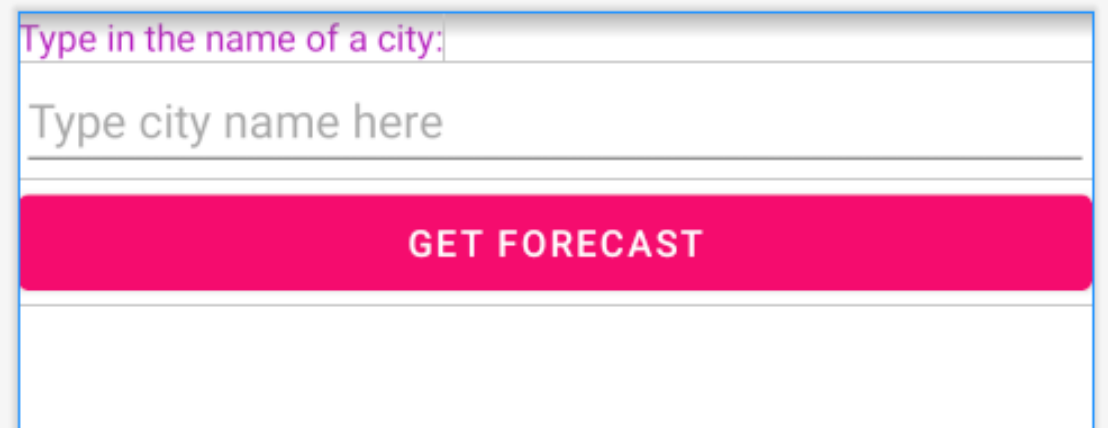
Setting up Volley - cont'd

- Change the "app/build.gradle" file to include the Volley libraries.
- Add this to the dependencies{} section:

```
implementation 'com.android.volley:volley:1.2.1'
```

Interface design

- Modify the layout in activity_main.xml to have an EditText and a Button like this (change **ConstraintLayout** to **LinearLayout**):



Interface design

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Type in the name of a city:" />

    <EditText
        android:hint="Type city name here"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/cityTextField"/>

    <Button
        android:id="@+id/forecastButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Get Forecast"/>

</LinearLayout>
```

Set onClick listener on Button

```
protected String cityName;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActivityMainBinding binding = ActivityMainBinding.inflate( getLayoutInflater() );
        setContentView(binding.getRoot());

        binding.forecastButton.setOnClickListener(click -> {
            cityName = binding.cityTextField.getText().toString();

        });
    }
```

Button code thus far…

```
binding.forecastButton.setOnClickListener(click -> {
        cityName = binding.cityTextField.getText().toString();
        String stringURL = "";

        //this goes in the button click handler:
        JsonObjectRequest request = JsonObjectRequest(Request.Method.GET, stringURL,
 null, (response) -> { },
                (error) -> { });
        queue.add(request);

});
```

stringURL variable

- The only thing missing now is the stringURL variable. In this class, we will be using a free weather server at this address:

https://api.openweathermap.org/data/2.5/weather?q={**city name**}&appid={**API key**}

You can generate your free API key from here:
https://openweathermap.org/price

stringURL variable

- By default, the server returns Fahrenheit, so to change it to celsius, you would add: &units=metric

https://api.openweathermap.org/data/2.5/weather?q=TORONTO&appid=7e943c97096a9784391a981c4d878b22**&units=metric**

Button code thus far…

```
binding.forecastButton.setOnClickListener(click -> {
    cityName = binding.cityTextField.getText().toString();
    String stringURL = null;
    try {
        stringURL = new StringBuilder()
            .append("https://api.openweathermap.org/data/2.5/weather?q=")
            .append(URLEncoder.encode(cityName, "UTF-8"))
           .append("&appid=7e943c97096a9784391a981c4d878b22&units=metric").toString();
           } catch(UnsupportedEncodingException e) { e.printStackTrace(); }

           //this goes in the button click handler (JsonObjectRequest):
           …
}); // binding.forecastButton.setOnClickListener
```

JsonObjectRequest

```
binding.forecastButton.setOnClickListener(click -> {
    …
    //this goes in the button click handler (JsonObjectRequest):
    …
    (response) -> {
        try {
     JSONObject coord = response.getJSONObject("coord");
     JSONArray weatherArray = response.getJSONArray("weather");
            JSONObject position0 = weatherArray.getJSONObject(0);
     String description = position0.getString(“description");
     …
            } //
     }
});// binding.forecastButton.setOnClickListener
```

JsonObjectRequest - cont'd

```
binding.forecastButton.setOnClickListener(click -> {
    …
    //this goes in the button click handler (JsonObjectRequest):
    …cont'd
        String iconName = position0.getString("icon");
        JSONObject mainObject = response.getJSONObject("main");
        double current = mainObject.getDouble("temp");
        double min = mainObject.getDouble("temp_min");
        double max = mainObject.getDouble("temp_max");
        int humidity = mainObject.getInt("humidity");
  }
    }
}); // binding.forecastButton.setOnClickListener
```

JsonObjectRequest - cont'd

```
//declare the next line outside the onCreate method
Bitmap image;
…
(response) -> {
…
  try {
      String pathname = getFilesDir() + "/" + iconName + ".png";
            File file = new File(pathname);
            if(file.exists()) {
               image = BitmapFactory.decodeFile(pathname);
            }
            else {/*. ImageRequest goes here */}
   }
}
```

JsonObjectRequest - cont'd

```
else {
   ImageRequest imgReq = new ImageRequest("https://openweathermap.org/img/img/w" +
iconName + ".png", new Response.Listener<Bitmap>(){
@Override
public void onResponse(Bitmap bitmap) {
  try{
      image= bitmap;
      image.compress(Bitmap.CompressFormat.PNG, 100,
MainActivity.this.openFileOutput(iconName + ".png", Activity.MODE_PRIVATE));
binding.icon.setImageBitmap(image);
  } catch(Exception e) { e.printStackTrace(); }
} // end of onResponse
```

JsonObjectRequest - cont'd

```
else {
    ...
}, 1024, 1024, ImageView.ScaleType.CENTER, null, (error) -> {
      Toast.makeText(MainActivity.this, ""+error, Toast.LENGTH_SHORT).show();
    });
     queue.add(imgReq);
} // end of else
```

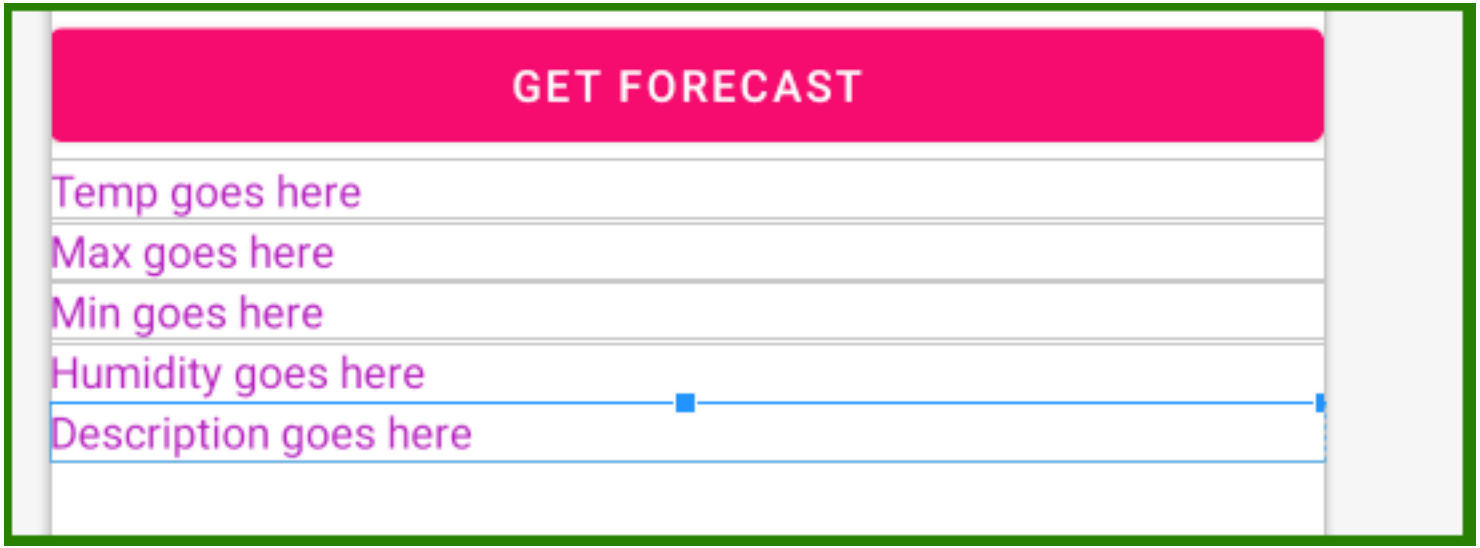
JsonObjectRequest - cont'd

```
} // end of try
            catch(JSONException e) {
                        e.printStackTrace();
            }
  },
  (error) -> { });
  queue.add(request);

}); // binding.forecastButton.setOnClickListener
```

Add TextView to view returned data

- Add these TextViews to your activity_main.xml, somewhere below the “Get Forecast” button:

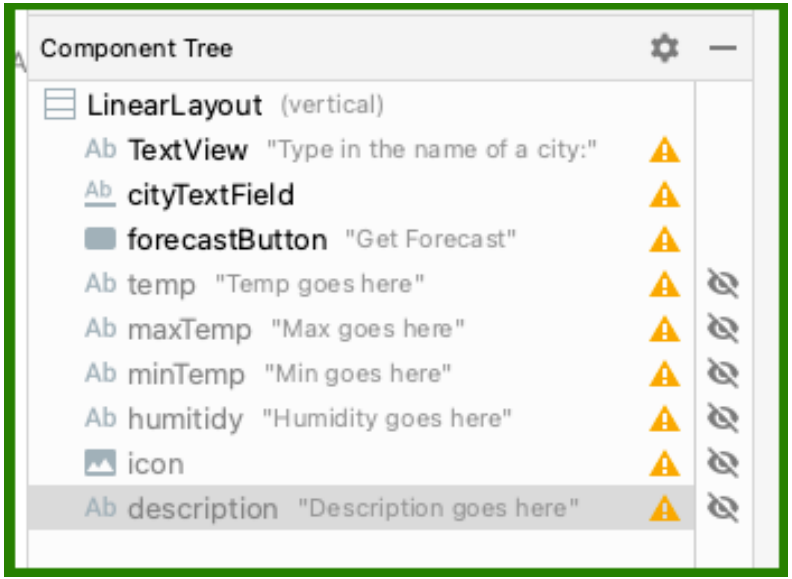


Design

```
18
19      <Button
20          android:id="@+id/forecastButton"
21          android:layout_width="match_parent"
22          android:layout_height="wrap_content"
23          android:text="Get Forecast"/>
24
25      <TextView
26          android:id="@+id/temp"
27          android:layout_width="match_parent"
28          android:layout_height="wrap_content"
29          android:text="Temp goes here" />
30
31      <TextView
32          android:id="@+id/maxTemp"
33          android:layout_width="match_parent"
34          android:layout_height="wrap_content"
35          android:text="Max goes here" />
36
37      <TextView
38          android:id="@+id/minTemp"
39          android:layout_width="match_parent"
40          android:layout_height="wrap_content"
41          android:text="Min goes here" />
42
43      <TextView
44          android:id="@+id/humitidy"
45          android:layout_width="match_parent"
46          android:layout_height="wrap_content"
47          android:text="Humidity goes here" />
48
49      <ImageView
50          android:id="@+id/icon"
51          android:layout_width="match_parent"
52          android:layout_height="wrap_content" />
53
54      <TextView
55          android:id="@+id/description"
56          android:layout_width="match_parent"
57          android:layout_height="wrap_content"
58          android:text="Description goes here" />
59
```

Hide the TextViews

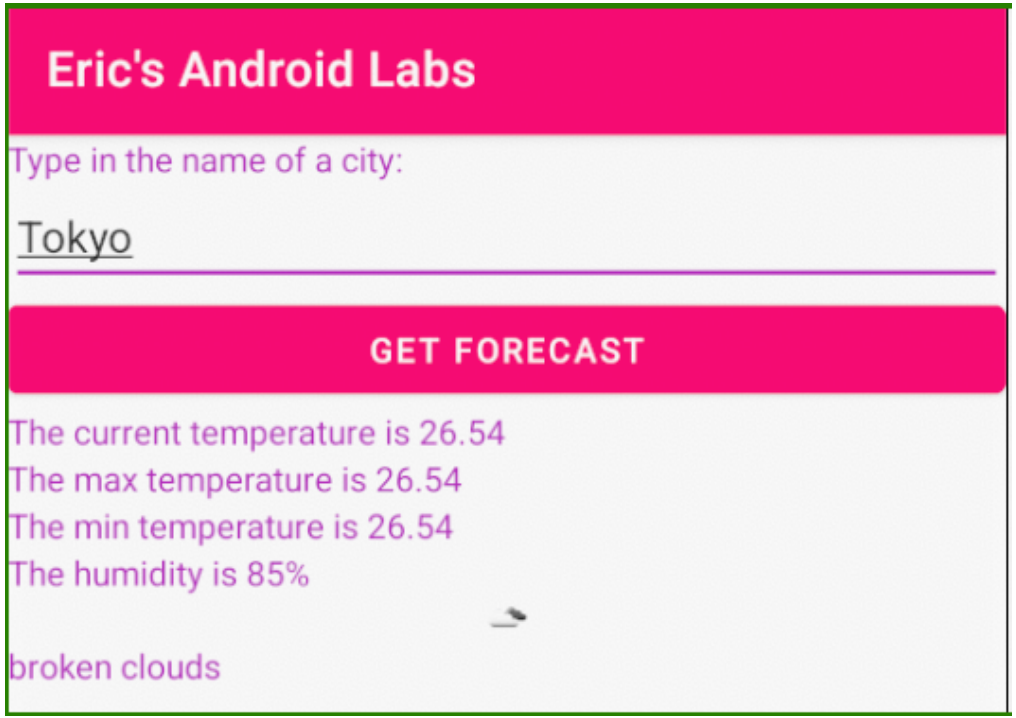
- Add these TextViews to your activity_main.xml, somewhere below the “Get Forecast” button:



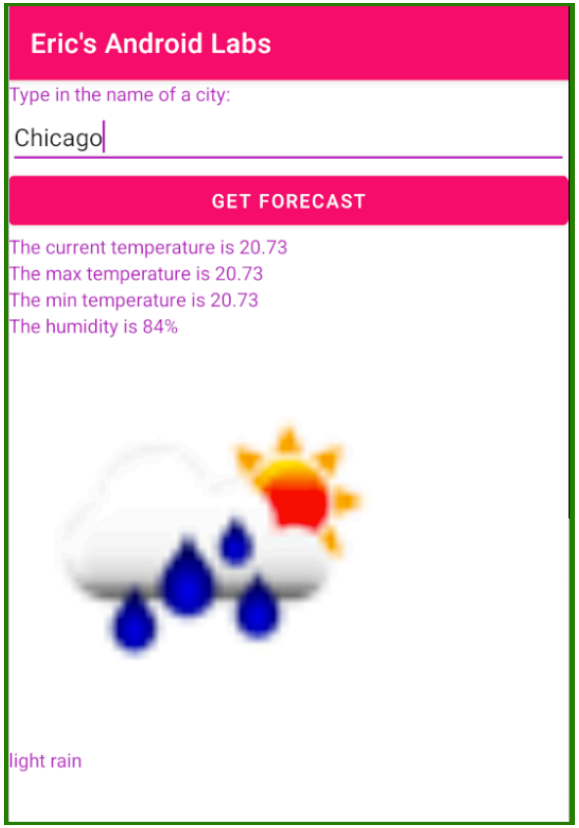
GUI Update

```
runOnUiThread(() -> {
        binding.temp.setText("The current temperature is " + current);
        binding.temp.setVisibility(View.VISIBLE);
        binding.minTemp.setText("The min temperature is " + min);
        binding.minTemp.setVisibility(View.VISIBLE);
        binding.maxTemp.setText("The max temperature is " + max);
        binding.maxTemp.setVisibility(View.VISIBLE);
        binding.humidity.setText("The humidity is " + humidity + "%");
        binding.humidity.setVisibility(View.VISIBLE);
        binding.icon.setImageBitmap(image);
        binding.icon.setVisibility(View.VISIBLE);
        binding.description.setText(description);
        binding.description.setVisibility(View.VISIBLE);
});
```

Final View


Eric's Android Labs
Type in the name of a city:
Tokyo
GET FORECAST
The current temperature is 26.54
The max temperature is 26.54
The min temperature is 26.54
The humidity is 85%
broken clouds

Final View


Eric's Android Labs
Type in the name of a city:
Chicago
GET FORECAST
The current temperature is 20.73
The max temperature is 20.73
The min temperature is 20.73
The humidity is 84%
light rain