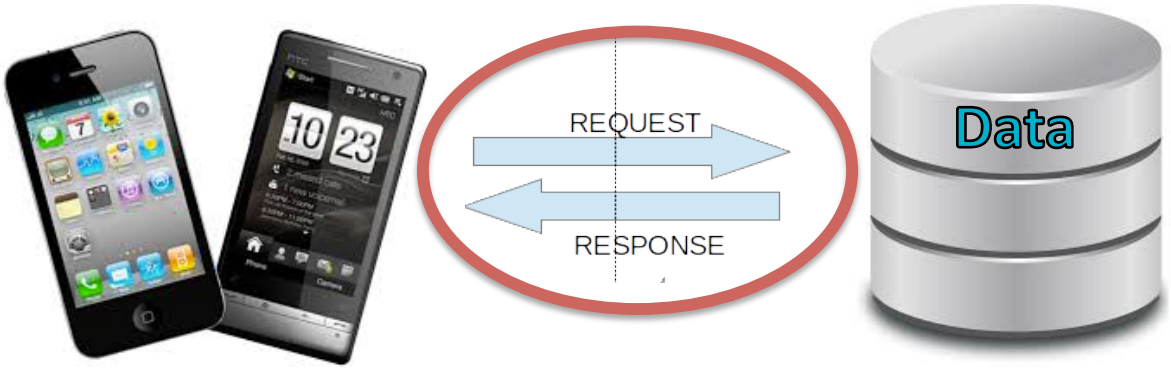
Meet-up 6: Talking to Your Backend

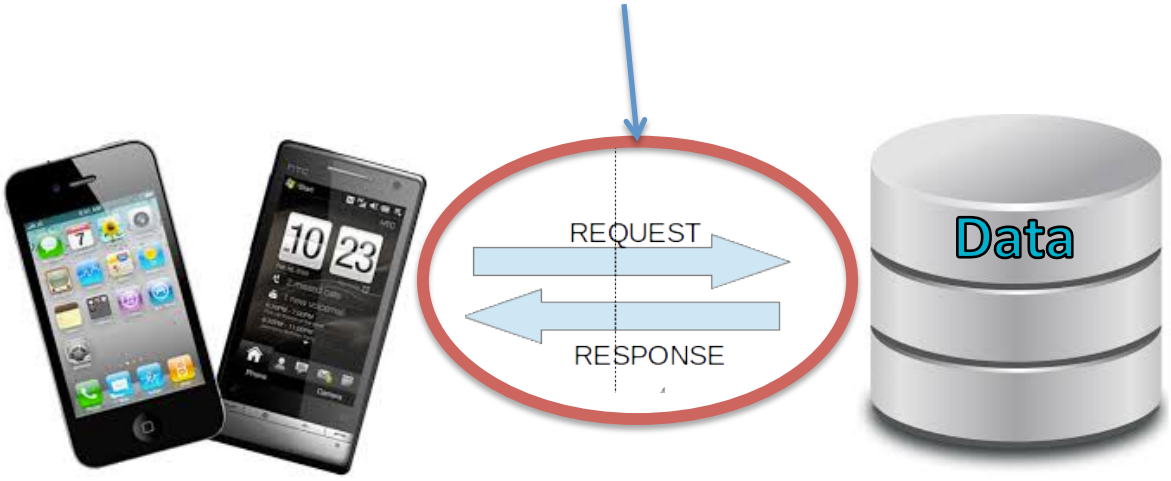
The Big Picture

The Big Picture

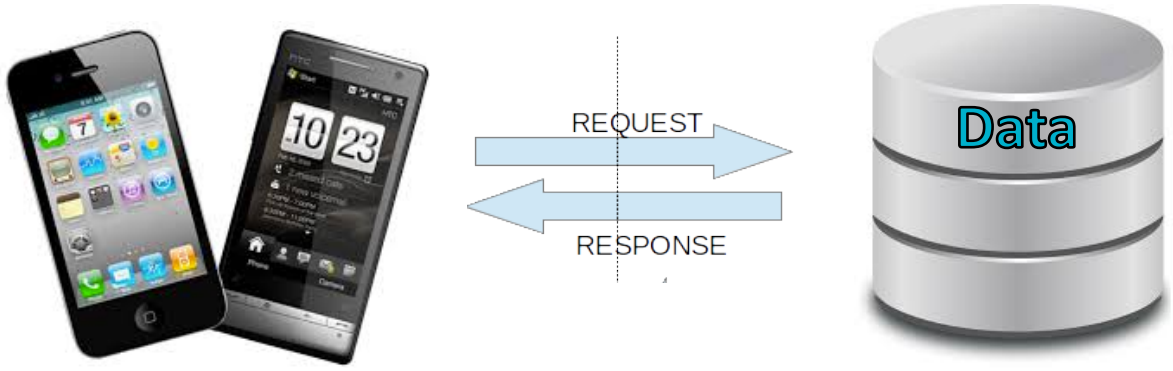
The Big Picture



RESTful API

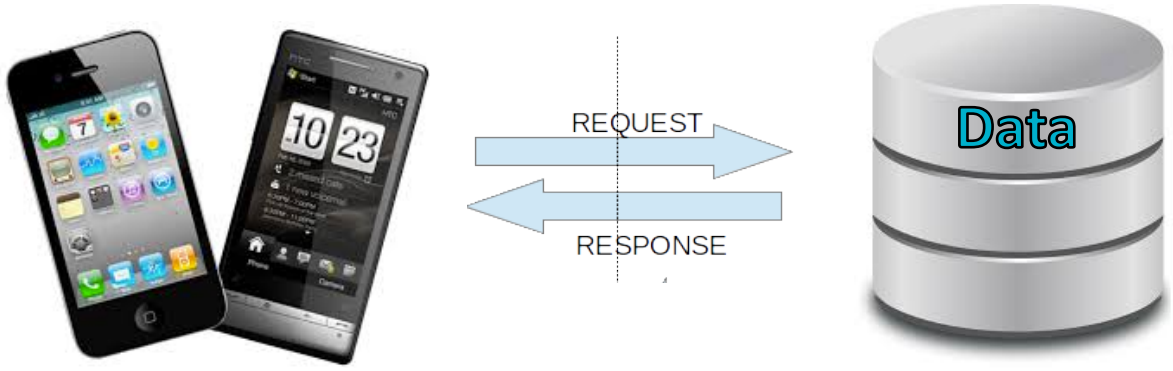


```
GET graph.facebook.com
        /<user_id>/posts
```



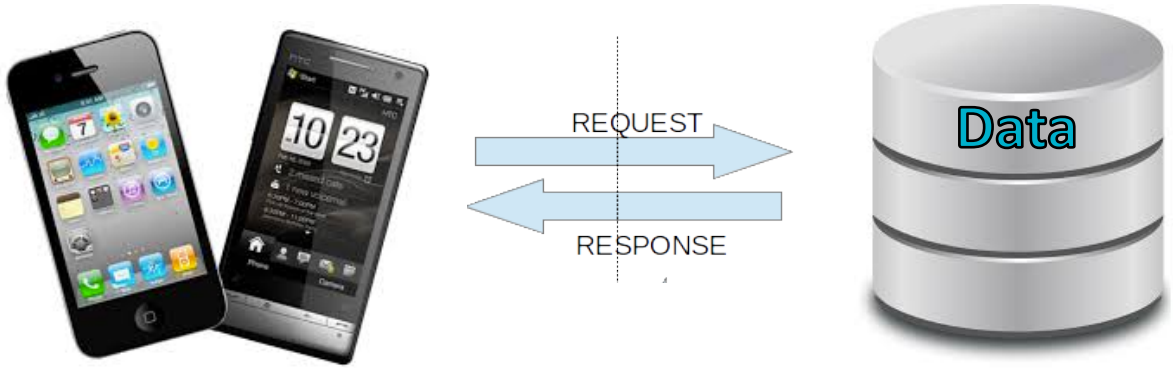
Facebook queries database:

```
user = User.objects.get(id = 1)
posts = user.post_set.all()
```



Facebook responds with posts JSON:

```
[{ post: { id: 1, text: 'Wesley is
the best...' }},
{ post: { id: 5, text: 'The students
who bribe me with the best Malaysian
food get As' }}]
```



REST verbs

GET	Retrieve data
POST	Create new data
PUT	Update existing data
DELETE	Delete data

Connecting Android over a RESTful API

First, your app needs the right permissions

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Source:
http://developer.android.com/training/basics/network-ops/connecting.html

Check the network connection

```
public void myClickHandler(View view) {
    ...
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected()) {
        // fetch data
    } else {
        // display error
    }
    ...
}
```

GETing data

Download in a separate thread using AsyncTask

```
new DownloadWebpageTask().execute(stringUrl);
```

```
private class DownloadWebpageTask extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {

        // params comes from the execute() call: params[0] is the url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
        textView.setText(result);
    }
}
```

Download away

```
private String downloadUrl(String myurl) throws IOException {
    InputStream is = null;
    // Only display the first 500 characters of the retrieved
    // web page content.
    int len = 500;

    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();
        int response = conn.getResponseCode();
        Log.d(DEBUG_TAG, "The response is: " + response);
        is = conn.getInputStream();

        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;

    // Makes sure that the InputStream is closed after the app is
    // finished using it.
    } finally {
        if (is != null) {
            is.close();
        }
    }
}
```

Parse JSON

```
JSONObject jObject = new JSONObject(result);
```

Or `JSONArray(result)`
if you're receiving an array

POSTing data

Create JSON String

```
import java.util.ArrayList;
import java.util.HashMap;
import org.json.JSONArray;
import org.json.JSONObject;

ArrayList<Integer> collection = new ArrayList<Integer>();
collection.add(Integer.valueOf(1));
collection.add(Integer.valueOf(2));

JSONArray jsonArray = new JSONArray(collection);
String jsonString = jsonArray.toString();
# jsonString is the string "[1, 2]"

HashMap<String, Object> map = new ArrayList<String, Object>();
map.put("key1", "string");
map.put("key2", Double.valueOf(2.5));

JSONObject jsonObject = new JSONObject(map);
jsonString = jsonObject.toString();
# jsonString is the string "{\"key1\": \"string\", \"key2\": 2.5}"
```

POST away

```
HttpURLConnection c;
OutputStream os;
byte[] data = jsonData.getBytes();
try {
    URL url = new URL("http://www.example.com/example_upload_site");
    c = url.openConnection();
    c.setDoOutput(true);
    c.setFixedLengthStreamingMode(data.length);
    os = c.getOutputStream();
} catch (MalformedURLException e) {
    // If the URL is not a valid URL.
} catch (IOException e) {
    // If an error prevented opening or writing to the stream.
}
try {
    os.write(data);
} catch (IOException e) {
    // If an error prevented opening or writing to the stream.
} finally {
    // This clause ensures that disconnect() is always run last,
    // even after an exception is caught.  You should wrap
    // reader.readLine() like this too.
    c.disconnect();
}
```

Remember:

POST in a separate thread using `AsyncTask`

Today's Assignment

Answer a survey for us

Answer a survey for us

Simple, right?

Problem: We have a backend but no frontend

Problem: We have a backend but no frontend

So write one for us

Questions located at:

`http://gsl-poll-app.herokuapp.com/surveys/1/`

POST responses to:

`http://gsl-poll-app.herokuapp.com/surveys/1/`

Format:

```
{
    "response": {
        "first_name": "Wesley",
        "last_name": "Graybill",
        "question_responses": [
            {
                "question": 1,
                "response_text": "Why do you have to make our lives so difficult?"
            },
            {
                "question": 2,
                "response_text": "Stop asking so many questions"
            }
        ]
    }
}
```