

Objectives

1

- Describe JSON
- Look at 2 different HTTP client packages
- Look at 2 different JSON packages
- Use a web API to get some data and then process that data for display in an app

Web APIs

2

- Often return data in the JSON or XML format
- www.programmableweb.com
- Example:
 - OMDb API
 - Simple, open movie data API
 - www.omdbapi.com

Using a Web API

3

- Spring Framework for Android
 - Previous class example
 - projects.spring.io/spring-android
 - Rest Client for Android
 - ✦ spring.io/guides/gs/consuming-rest-android
- HTTP client library in **java.net** package
 - Fairly low-level handling
 - ✦ java.net.HttpURLConnection
 - ✦ java.net. HttpURLConnection
 - ✦ java.net.URL
 - ✦ java.net.URLEncoder

java.net http package

4

- **Instantiate a URL class, encoding any strings as needed**

```
String url_params = URLEncoder.encode(params[0]);  
URL url = new URL("http://www.omdbapi.com/?t=" + url_params );
```

- **Instantiate an http connection class, and set its parameters as needed**

```
HttpURLConnection conn =  
    (HttpURLConnection) url.openConnection();  
conn.setDoOutput(true);  
conn.setRequestMethod("GET");  
conn.setRequestProperty("Accept", "application/json");
```

- **Read the response from the connection, using either**

- public [InputStream](#) getInputStream() throws [IOException](#)
- public [Object](#) getContent() throws [IOException](#)

JSON

5

- Javascript Object Notation
- Simple data-interchange format
 - Human-readable
 - Easy to parse
- Structures:
 - Arrays
 - Collections of key-value pairs
- www.json.org

JSON packages

6

- org.json.*
 - www.json.org/java
- Google gson
 - com.google.gson.*
 - github.com/google/gson

org.json main classes

7

- **JSONObject**

- `String getString(String key)`
- `int getInt(String key)`
- `JSONObject getJSONObject(String key)`
- `JSONArray getJSONArray(String key)`
- **And so on**

- **JSONArray**

- `int length()`
- `int getInt(int index)`
- `JSONObject getJSONObject(int index)`
- **And so on**