CPSC475/575
Networking using http

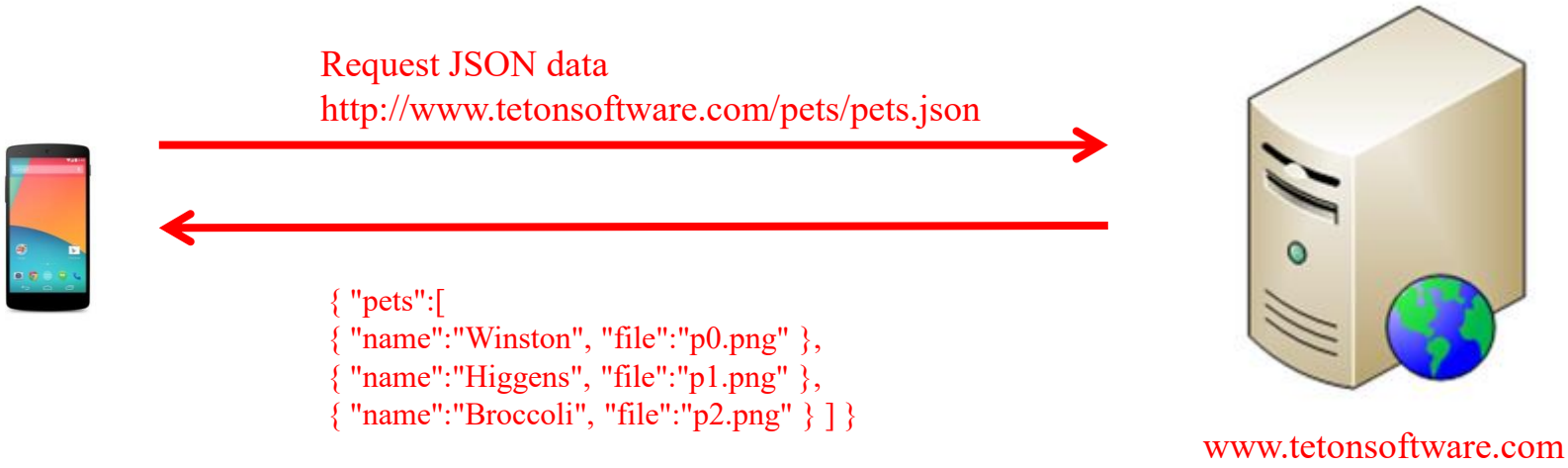
Topics

- Networking overview
- HttpURLConnection
- Why all the http based networking?
- Webservices – data transport mechanism
  - XML
  - JSON
- JSON
  - Introduction
  - How to use it
  - Troubleshooting
- How to query Connectivity

Networking choices

- **Many ways to communicate with a server**
  - Socket class
    - Lets you do general-purpose network programming
      - Same as with desktop Java programming
  - HttpURLConnection
    - Simplifies connections to HTTP servers
      - Same as with desktop Java programming
  - ~~HttpClient~~ Removed as of 6.0 – don't use anymore
    - ~~Simplest way to download entire content of a URL~~
      - ~~Not standard in Java SE, but standard in Android~~
  - JSONObject
    - Simplifies creation and parsing of JSON data
      - Not standard in Java SE, but standard in Android

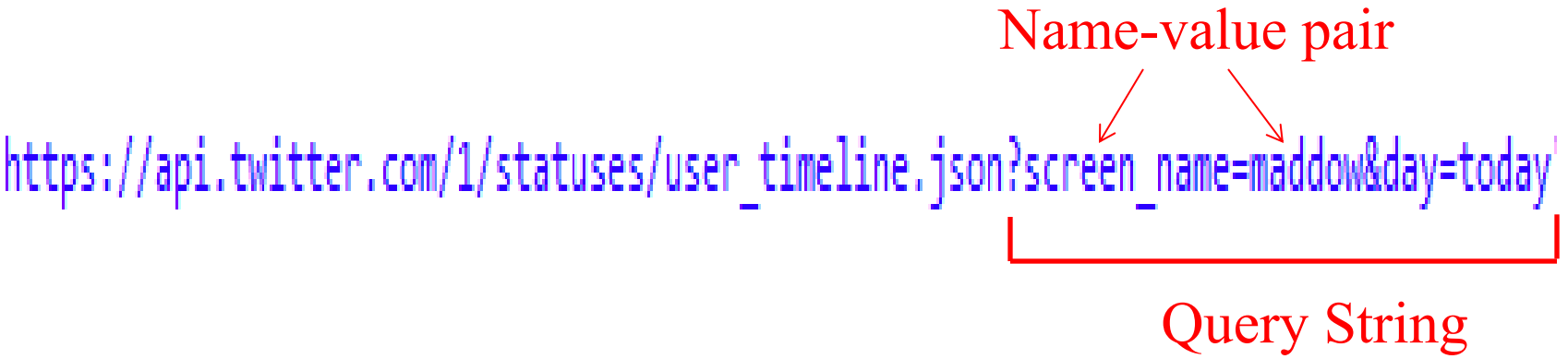
Why all the bother with web based networking?



- **Webservices - applications that run over the web, meant to be machine consumed, not intended for browser display.**
- **Much of the worlds data available through webservices**
- **Connect via http or https**
- **They mostly use XML and JSON for serialization and transport**

http GET- format of request

- Can send data to server as part of URL
- Data encoded in URL as name value pairs
- Known as a http query string
- first character after URL is “?”
- Each name-value pair separated by &



http POST- format of request

- **Sends data to server, form of name-value pairs**
- **No ? mark**
- **Each pair separated by &**
- **URLEncode for POST transactions**
  - slightly more complicated than GET
  - Use when you have a lot of data to send (like an image upload)

http GET vs POST summary

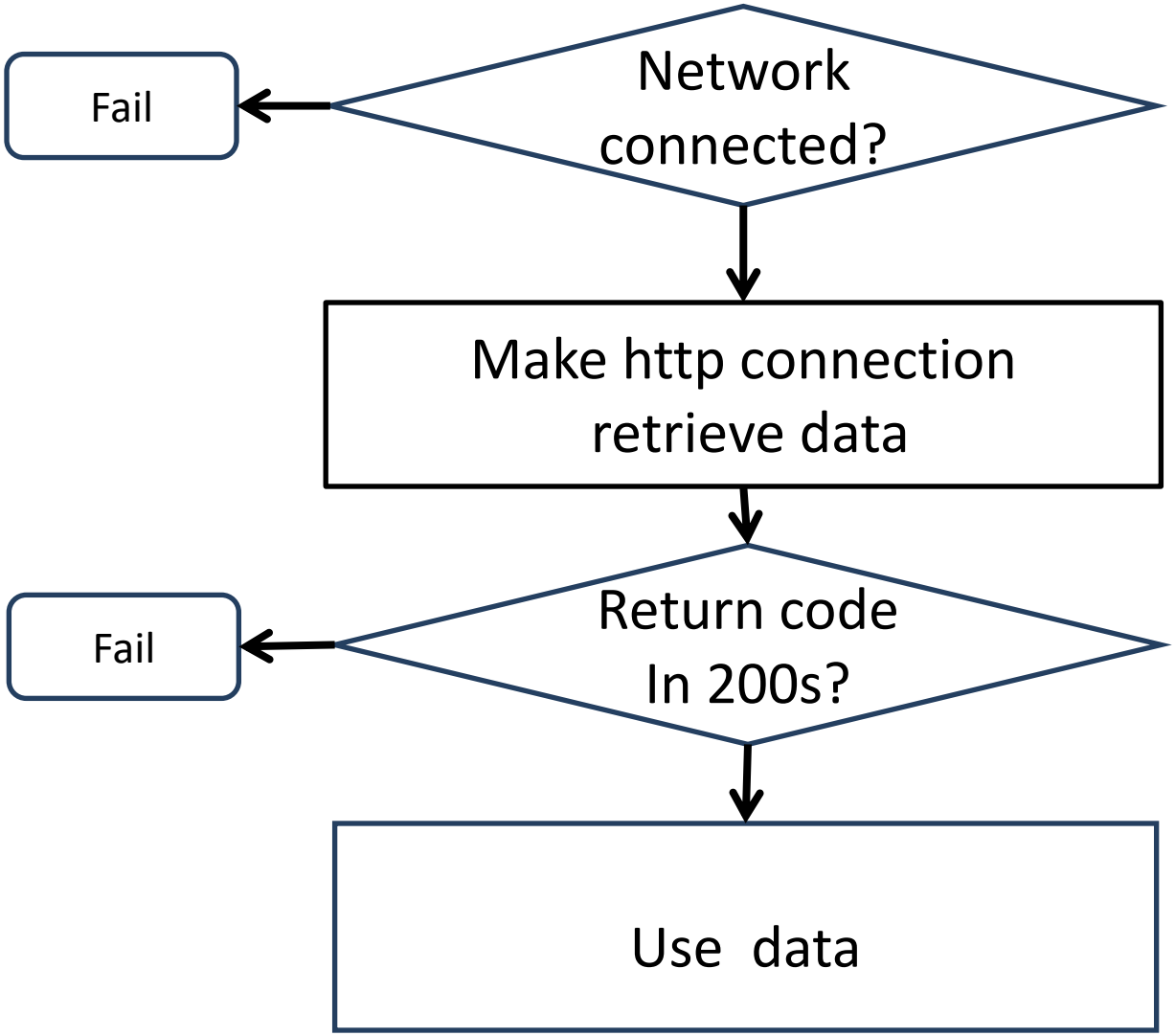
	GET	POST
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

Size matters

How you handle images and encrypted data

Chart from http://www.w3schools.com/tags/ref_httpmethods.asp

Software procedure for network calls



Network Connected?

If not you cannot do http, so you must verify your connection as the first order of business

Are you connected?

- **Network isn't always there**
- **Check before attempting network call**
- **See isNetworkReachable() isWifiReachable()**
- **Notify user?**
  - Kind of Lazy, you want to ease their burden not put your burden on them
  - Maybe wait a while and try again?

Are you connected?

- **Can also lose connection after initial success**
- **Check to see if you have connectivity whenever a network request fails.**

Demonstration

NetworkCheck

Http in Android

HttpURLConnection

Perform all network calls on a separate Thread!

Manifest File Permissions AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.coreservlets.networking"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.INTERNET" />
   <application
    ...
</manifest>
```

URL and HttpURLConnection: Overview

- **URL**
  - The URL class can parse a URL and extract its components (protocol, host, port, URI, etc.)
  - Use openConnection to get a stream to the URL
- **HttpURLConnection**
  - If the URL's protocol is HTTP, cast the result of openConnection to HttpURLConnection
    - Lets you read the response codes
      - 200, 404, etc.
    - Lets you read data returned from stream

XML and JSON (webservice ‘language’)

XML and JSON

- **Both character or string based transport**
- **XML – most common format**
  - General purpose with validation
  - Can do everything but a bit more complex than JSON
- **JSON – also very common**
  - Easier for humans to read than XML
  - Smaller in size (data objects verses equivalent XML)
  - Faster to parse than XML
  - Generally easier to use, therefore we will focus on it in this course

Search ‘JSON verses XML’ for in depth pro/con discussions

JSON – where used

- **Yahoo APIs**
  - Search, travel, answers
    - http://developer.yahoo.com/
- **Twitter APIs**
  - https://dev.twitter.com/
- **GeoNames**
  - http://www.geonames.org/export/web-services.html
- **Flickr**
  - http://www.flickr.com/services/api/
- **Thousands of others**
  - See list here
    - http://www.programmableweb.com/apis/directory/1?format=JSON

JSON – Syntax Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Entire JSON code is wrapped in {}

```
{
"employees": [
{ "firstName":"John" , "lastName":"Doe" },
{ "firstName":"Anna" , "lastName":"Smith" },
{ "firstName":"Peter" , "lastName":"Jones" }
]
}
```

Original content located at http://www.w3schools.com/json/default.asp

JSON – Arrays

- JSON arrays are written inside square brackets.
- An array can contain multiple objects:

```
{
"employees": [
{ "firstName":"John" , "lastName":"Doe" },
{ "firstName":"Anna" , "lastName":"Smith" },
{ "firstName":"Peter" , "lastName":"Jones" }
]
}
```

Original content located at http://www.w3schools.com/json/default.asp

JSON – Values

- A number (integer or floating point)
- A string (in double quotes)
- A Boolean (true or false)
- An array (in square brackets)
- An object (in curly brackets)
- null

Original content located at http://www.w3schools.com/json/default.asp

JSON and Android

- Android has built in classes (JSONObject, JSONArray) that will both build and parse strings representing JSON
- Downside- have to know what is in data

```
try {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    tvfirstname.setText(jsonObject.getString("firstname"));
    tvlastname.setText(jsonObject.getString("lastname"));
} catch (JSONException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

JSONObject and Android
Extracting Data From JSONObject

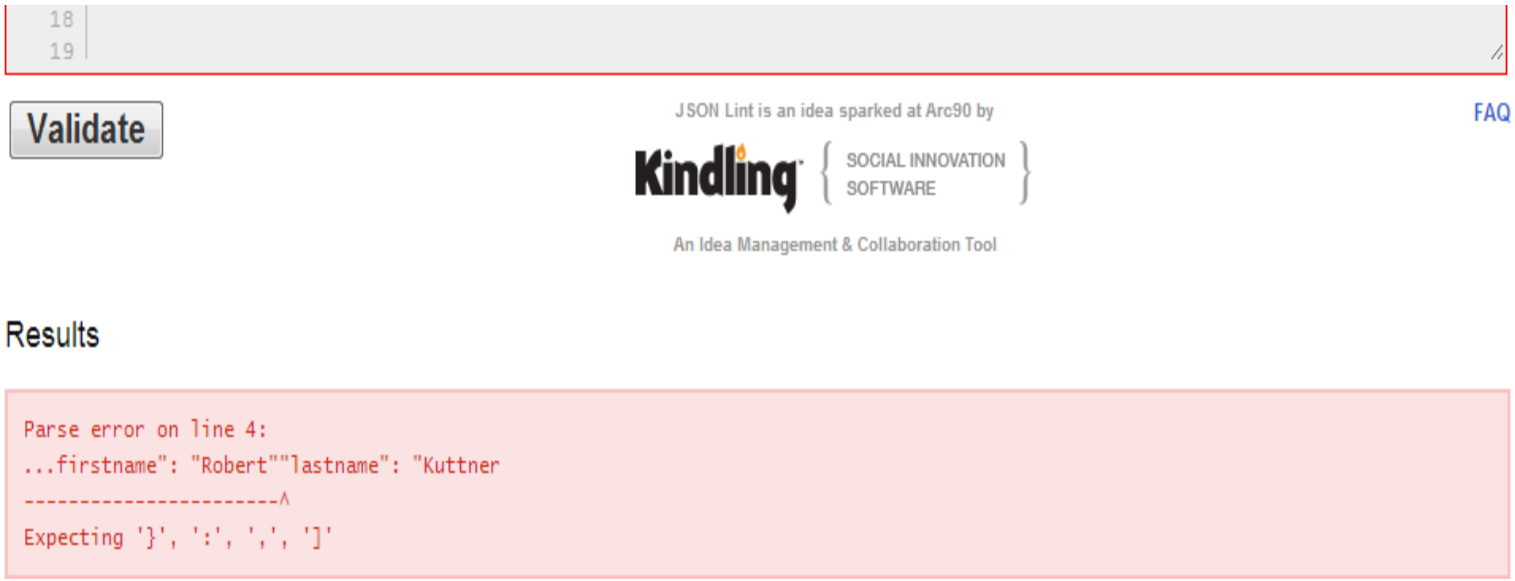
- **Accessors**
  - get(propertyName)
    - Returns Object associated with name
  - getString(propertyName)
    - Returns String associated with name. Works for any type (if Object, uses its toString method)
  - getDouble(propertyName)
    - Returns double associated with name. Throws JSONException if value cannot be converted to double.
  - get*Blah*(propertyName)
    - getInt, getBoolean, etc. Similar in spirit to getDouble.
  - getJSONArray(propertyName)
    - Returns JSONArray (not native Java array!) associated with name

JSONs not working. Is it my JSON script or my Java code?

- Test your JSON. Paste into JSON validator.
- Ex. http://jsonlint.com/
- Paste code, click validate button

JSONs not working. Is it my JSON script or my Java code?

- I appear to be missing a ,



More Reading

- **JSON Tutorial (generic lots of others)**
  - http://www.w3schools.com/json/default.asp

- **Json is used a LOT in the world, and also in Projects 3 and 4**

Demonstration 8_ParseJSON

ParseJSONActivity. processJSON