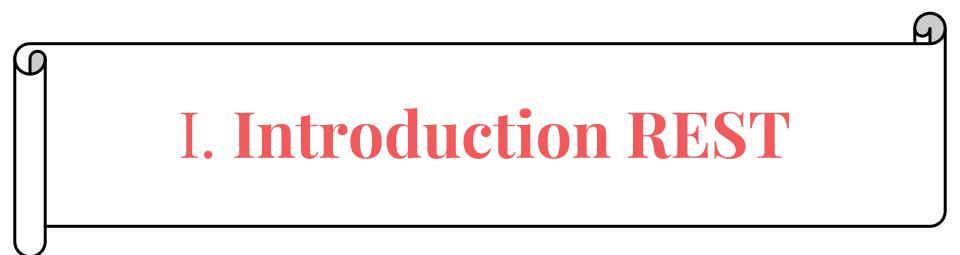
Android Advance Lesson 7 JSON, HttpUrlConnection



Outline

- I. Introduction REST
- II. Introduction HttpURLConnection
- III. Introduction JSON



1. What is REST

- **REST** stands for **R**epresentational **S**tate **T**ransfer. (It is sometimes spelled "ReST") It relies on a stateless, client-server, cacheable communications protocol -- and in virtually all case, the HTTP protocol is used
- REST is an architecture style for designing networked applications, rather than using complex mechanisms such as CORBA, RPC or SOAP to connect between machines
- REST uses HTTP for all four CRUD operations
- REST is not a "standard". There will never be a W3C recommendation for REST

1. What is REST

- REST as Lightweight Web Services:
 - Much like Web Service, a REST service is
 - Platform-independent
 - Language-independent
 - Sandards-based (ron on top of HTTP)
 - Can easily be used in the presence of firewalls
 - Like Web Services, REST offers no built-in Security features, encryption, session management, QoS guarantees, etc. But also as with Web Services, these can be added by building on top of HTTP
 - For security, username/password token are often used
 - For encryption, REST can be used on top of HTTPS
 - ...

II. Introduction HttpURLConnection

1 What is HttpURLConnection

- Defined in java.net package
- Is an abstract subclass of URLConnection that provides some additional methods specific to HTTP Protocol
- URLConnection Object that returned by an http URL will be instance of java.net.HtttpURLConnection

2. How to use HttpURLConnection

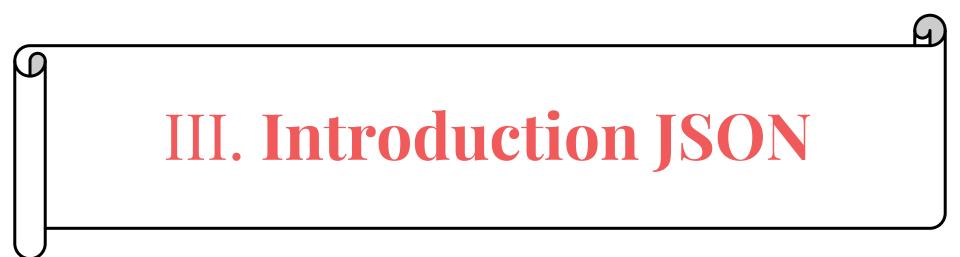
- Obtain a new HttpURLConnection
 - by calling URL.openConnection() and casting the result to HttpURLConnection.
- Prepare the request (URI)
- Optionally upload a request body.
- Read the response.
 - Response headers typically include metadata such as the response body's content type and length, modified dates and session cookies. The response body may be read from the stream returned by getInputStream(). If the response has no body, that method returns an empty stream.
- Disconnect.
 - Once the response body has been read, the HttpURLConnection should be closed by calling disconnect(). Disconnecting releases the resources held by a connection so they may be closed or reused.

2. How to use HttpURLConnection

```
public void connectInternet(String link) throws IOException {
    URL url = new URL(link);
    HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
    InputStream in = new BufferedInputStream(httpURLConnection.getInputStream());
    String data = "";
    int charactor;
    while ((charactor = in.read()) != -1) {
        data += charactor;
    }
    httpURLConnection.disconnect();
}
```

3. Some public method

abstract void	disconnect()
	Indicates that other requests to the server are unlikely in the near future.
void	setRequestMethod(String method)
	Set the method for the URL request, one of:
	GET, POST, HEAD, OPTIONS, PUT, DELETE, TRACE
	are legal, subject to protocol restrictions.
int	getResponseCode()
	Gets the status code from an HTTP response message.
String	getResponseMessage()
	Gets the HTTP response message, if any, returned along with the response code from a server.
InputStream	getInputStream(int contentLength)
	an input stream that reads from this open connection.



1. What is JSON

- JSON (JavaScript Object Notation) is a lightweight data- interchange format.
- JSON is a syntax for storing and exchanging data.
- JSON is text, written with JavaScript object notation.

2. Why JSON

- Because JSON is lightweight, easy to understand
- Easy for humans to read and write
- Easy for machines to parse and generate

```
{"menu": {
    "id": "file",
    "value": "File",
    "popup": {
        "menuitem": [
            {"value": "New", "onclick": "CreateNewDoc()"},
            {"value": "Open", "onclick": "OpenDoc()"},
            {"value": "Close", "onclick": "CloseDoc()"}
        ]
    }
}
```

3. JSON Structures

- JSON is built on two structures
 - A collection of name/ value pairs
 - In various languages, this is realized as an object, record, dictionary, hash table, keyed list or associative
 - An ordered list of values
 - In most language, this is realized as an array, vector, list or sequence

4. Android JSONObject

- A modifiable set of name/value mappings.
- Names are unique, non-null strings. Values may be any mix of JSONObjects,
 JSONArrays, Strings, Booleans, Integers, Longs, Doubles or NULL.
- Values may not be null, NaNs, infinities, or of any type not listed here.

Object	get(String name)
	Returns the value mapped by name, or throws if no such mapping exists.
Iterator <string></string>	keys()
	Returns an iterator of the String names in this object.
Object	opt(String name)
	Returns the value mapped by name, or null if no such mapping exists.
JSONArray	optJSONArray(String name)
	Returns the value mapped by name if it exists and is a JSONArray, or null otherwise.

5. Android JSONArray

- A dense indexed sequence of values
- Values may be any mix of JSONObjects, other JSONArrays, Strings, Booleans, Integers, Longs, Doubles, null or NULL. Values may not be NaNs, infinities, or of any type not listed here.

Object	get(int index)
	Returns the value at index.
JSONArray	getJSONArray(int index)
	Returns the value at index if it exists and is a JSONArray.
Object	opt(int index)
	Returns the value at index, or null if the array has no value at index.
JSONObject	optJSONObject(int index)
	Returns the value at index if it exists and is a JSONObject.



Q&A

Sample

Project search user github:

Link: https://github.com/daolq3012/SearchUserGithubAsyncTask



Exercise

Apply knowledge above get data from <u>github api</u> display in a list