BCS3283-Mobile Application Development

Chapter 9
ANDROID, PHP, MYSQL

Editor
Dr. Mohammed Falah Mohammed

Faculty of Computer Systems & Software Engineering
falah@ump.edu.my

ANDROID, PHP, MYSQL

- Aims

  To learn about the PHP and MYSQL based android concepts and how to connect to MYSQL database in Android Studio.

- Expected Outcomes
  - Understanding the concept of PHP and MYSQL database
  - What is JSON?
  - What is **AsyncTask**?
- References
  - https://www.simplifiedcoding.net
  - http://www.compiletimerror.com/
  - http://android.toolib.net/
  - https://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/
  - https://www.survivingwithandroid.com/2016/07/android-http-library.html

Android HTTP library

- The HTTP library uses to Handle the following activities:
    1. managing the HTTP connection to remote server.
    2. JSON data.
    3. Images.

Note: while there are many HTTP libraries, users must choose the suitable one according to their application requirements and needs.

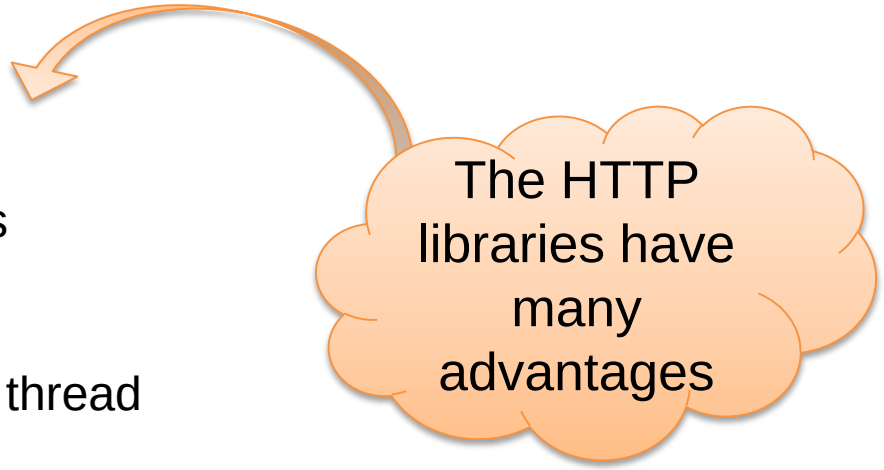
- There are two main libraries at the level of HTTP handling:
    1. Volley
    2. OkHTTP

- While there are other libraries use for more specific tasks, i.e., handling image over HTTP, handling JSON data and extra.

Android HTTP library

Why it is preferable to use the Android HTTP library alternatives?

1. Efficiency
2. Parallel requests
3. Caching system
4. Non-blocking UI thread
5. HTTP/2 support



- The Android HTTP library is based on HttpURLConnection.

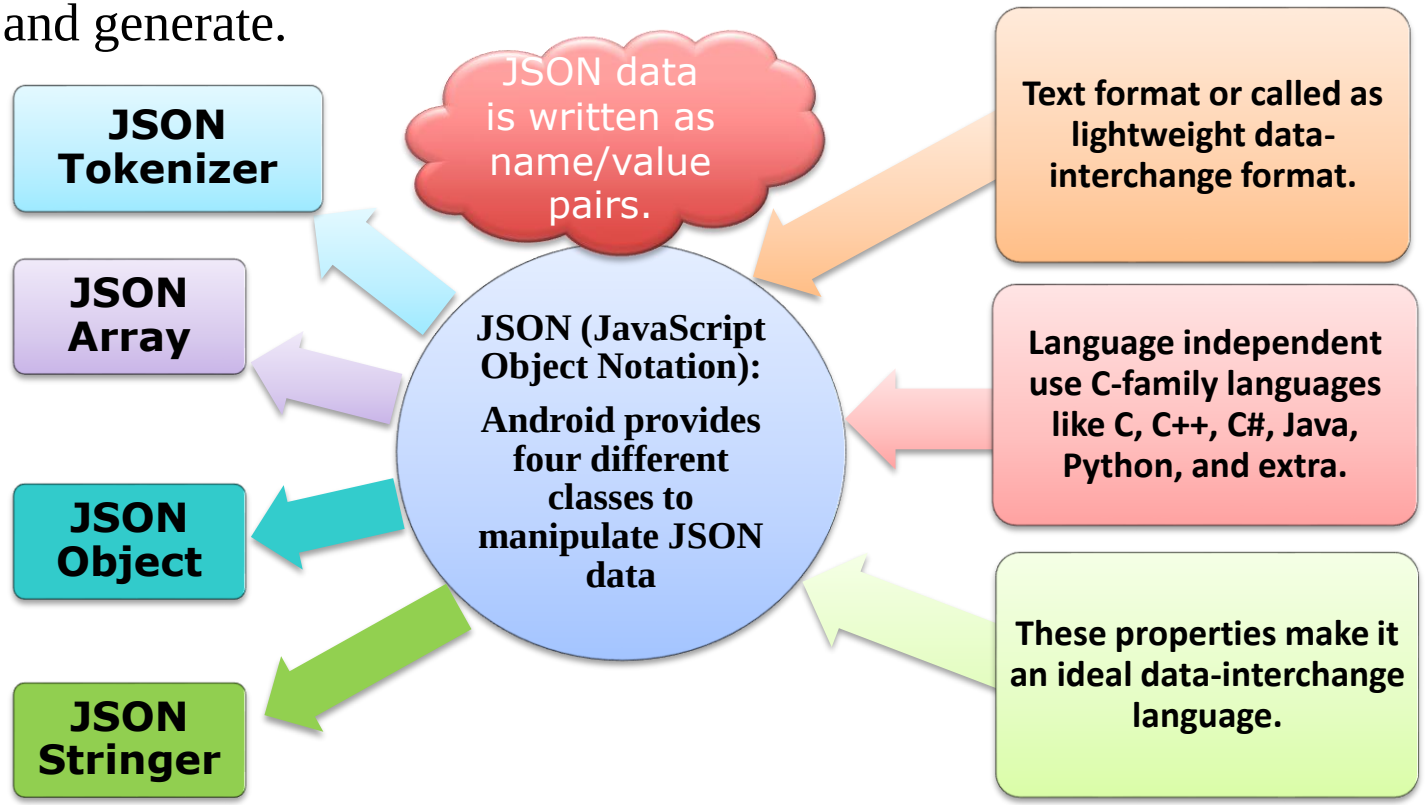
HttpURLConnection

- To send and receive information over the web, the URL Connection for HTTP is used. To do so:
    1. Obtain a new HttpURLConnection.
    2. Prepare the request.
    3. Optionally upload a request body.
    4. Read the response.
    5. And finally, disconnect.

- Data can only be as text when exchanging it between a browser and a server.
- While JSON is text, so we can convert any JavaScript object into a JSON format and send it to a server and vice versa, hence, simplify working with the data as JavaScript objects.

JSON

JSON is an independent data exchange format and considers as the best alternative for XML. It uses to send/receive data from/to server in a straightforward manner for programmers to read/write and for machines to parse and generate.



JSON Code Example

Sending Data Code:

```
<!DOCTYPE html>
<html>
<body>

<h2>Convert a JavaScript object into a JSON string, and send it to the server.
</h2>

<script>
var myObj = { name: "John", age: 31, city: "New York" };
var myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
</script>

</body>
</html>
```

Output is :

John from New York is 31

Source: https://www.w3schools.com/js/js_json_intro.asp

JSON Code Example

Receiving Data Code:

```
<!DOCTYPE html>
<html>
<body>

<h2>Convert a string written in JSON format, into a JavaScript object.</h2>

<p id="demo"></p>

<script>
var myJSON = '{"name":"John", "age":31, "city":"New York"}';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
</script>

</body>
</html>
```

Output is :

Convert a string written in JSON format, into a JavaScript object.

John

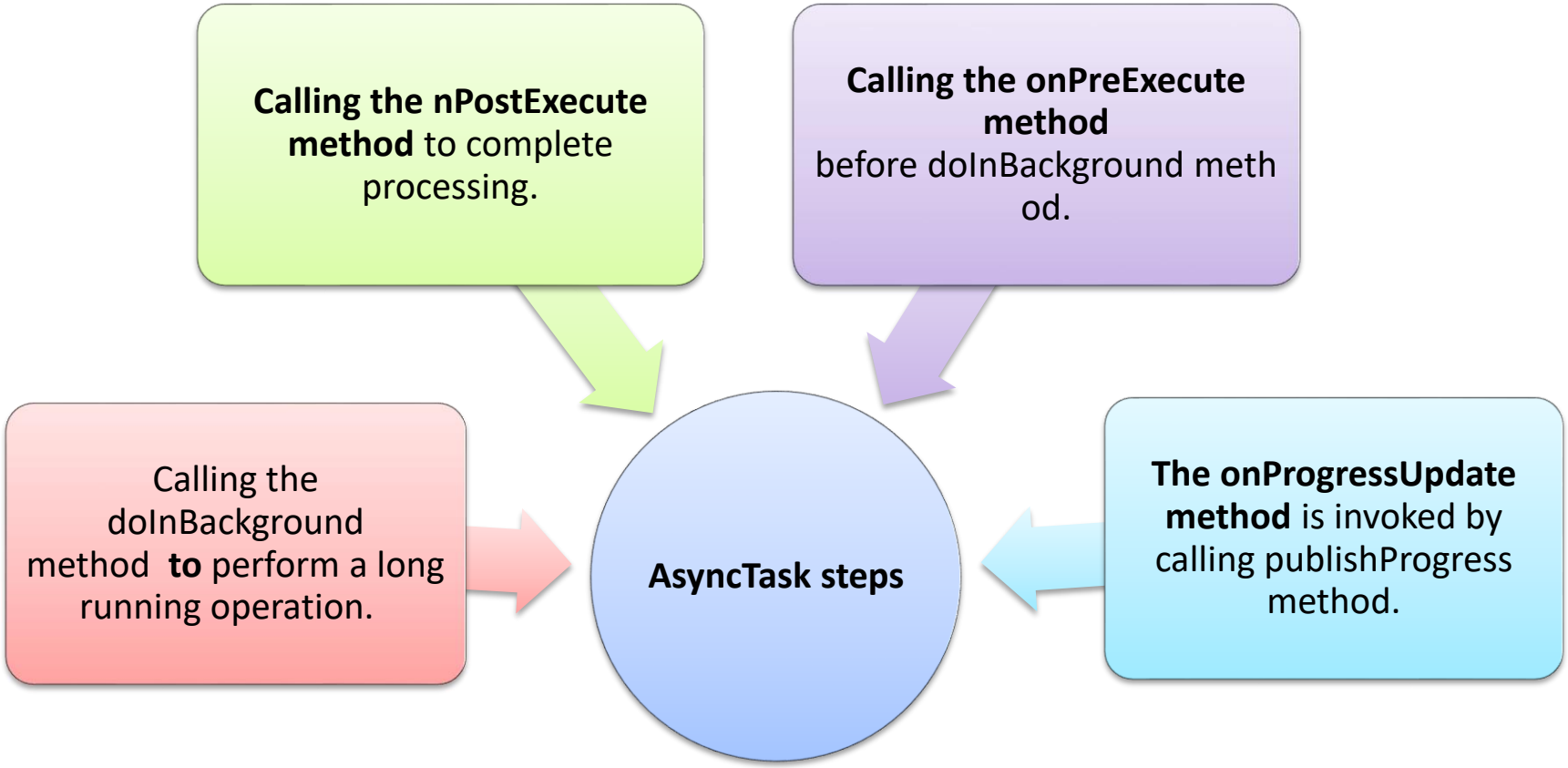
Source: https://www.w3schools.com/js/js_json_intro.asp

AsyncTask

- The AsyncTask class uses with HttpUrlConnection to support thread.
- AsyncTask is an abstract class which Android provide it to enable programmers to use the UI thread properly.
- AsyncTasc will be created when a user launches the android application.
- To overcome the awaited case and to avoid the long-running operation on UI thread, there is a need to create a new thread and implementing the run method (UI remains responsive).

Note: Android UI toolkit is not a thread-safe, because Android is performing as a single thread model. For that reason, it is better to use the AsyncTask for a task that takes a few seconds of operation.

AsyncTask



***Note** that invoking cancel (boolean) method led to cancel the task.*

Connect Android with PHP, MySQL

To connect Android with PHP, MySQL we need to do the following:

1. Installing and Running WAMP Server
   - Download & Install WAMP server.
   - Test your server by opening http://localhost/.
2. Creating and Running PHP Project
   - Create a new project folder inside (**www** folder) and place all your project files inside it.
   - Create a folder called ”**android_connect”**, then create a new PHP file “**test.php”** and try out a simple PHP code to test the connection.

```
<?php
    echo "Welcome, I am connecting Android to PHP, MySQL";
?>
```

Connect Android with PHP, MySQL

3. Create MySQL Database and Tables

- Name the database as **“androidhive”**, while the table as **“products”**

```
CREATE TABLE products(
pid int(11) primary key auto_increment,
name varchar(100) not null,
price decimal(10,2) not null,
description text,
created_at timestamp default now(),
updated_at timestamp
);
```

4. Connecting to MySQL database using PHP

Create a PHP class to connect to MySQL database to open a and close connection to database. Two files are required for that as follows:

- **db_config.php** – will have database connection variables
- **db_connect.php** – a class file to connect to database

Connect Android with PHP, MySQL

db_config.php

```
<?php
define('DB_USER', "root"); // db user
define('DB_PASSWORD', ""); // db password (mention your db password here)
define('DB_DATABASE', "androidhive"); // database name
define('DB_SERVER', "localhost"); // db server
?>
```

db_connect.php

```
<?php
class DB_CONNECT {
    function __construct() {
            $this->connect();    }
    function__destruct() {
            $this->close();    }
    function connect() {
            require_once __DIR__ . '/db_config.php';
            $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or die(mysql_error());
            $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or die(mysql_error());
             return $con;  }
    function close() {
             mysql_close(); }
}
?>
```

Connect Android with PHP, MySQL

- Whenever you want to connect to MySQL database and do some operations, use the db_connect.php class like this:

```
$db = new DB_CONNECT(); //creating class object(To open database connection)
```

5. Basic MySQL CRUD Operations using PHP
   - That helps you to cover the basic **CRUD (Create, Read, Update, Delete)** operations on MySQL database using PHP.

6. Creating Android Application
   - Don't forget to add internet permission (if necessary).

More details are available on:

Source: https://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/