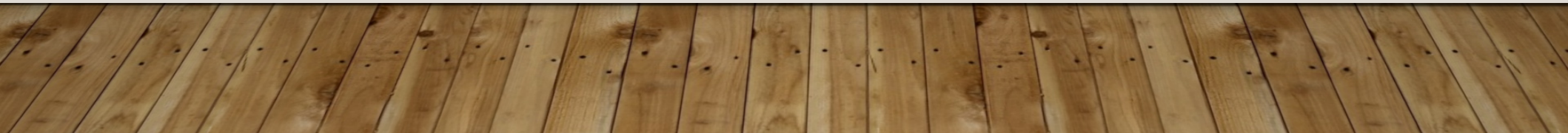
Introduction to: Network Communication in Android



TOPICS

- Introduction
- Http Protocol
  - Http request
  - http response
- Downloading data from remote server
- Example

Introduction Networking API

- Connecting devices over the Internet and exchanging Data
- Network API in Android provide an accessed to remote servers over the HTTP protocol.
- The data exchange use

  RESTful service that is based on JSON /XML objects

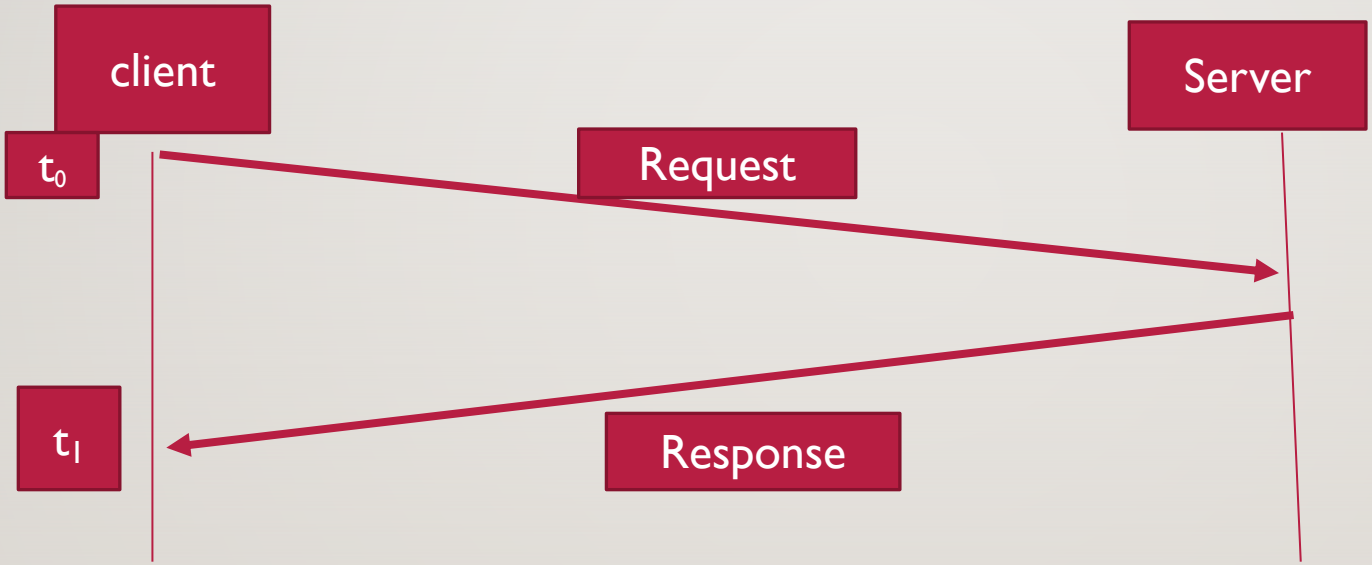
HTTP PROTOCOL

- protocol that is widely used to communicate (web)
- perform various network operation
  - Downloading webpages
  - Downloading/uploading files and images

THE HYPERTEXT TRANSFER PROTOCOL (HTTP)

- is an application-level protocol
- TCP/IP based communication
- **HTTP is connectionless and stateless:**
- client-server architecture that exchange stateless request/response

CLIENT - SERVER COMMUNICATION



HTTP REQUEST MAIN METHODS

- **GET :**

  to retrieve information from server

- **POST**

  to send data to the server

- The Protocol support other methods such as Delete , Put (we will focus on GET and POST)

STEPS FOR DOWNLOADING DATA

Step 3 and 4 must be done in background thread

- 1- add permission to manifest file
- 2- Check if the device is connected
- 3- get a connection to remote server(url)
- 4- download the data
- 5- process the in coming response and display the data

ADDING PERMISSIONS

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

```
ConnectivityManager connMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
if (networkInfo != null && networkInfo.isConnected()) {
        // connected   downloadData()
} else {
      // not connected
}
```

```
private String downloadData (String myurl) throws IOException {
    InputStream is = null;

    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection
        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();

          // process and display the result

         String result = processResponse(is);

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

Getting connection and downloading data

Must be done in a background

```
private String processResponse(InputStream is) throws Exception{
        InputStreamReader isr = new InputStreamReader(is);
        BufferedReader br = new BufferedReader(isr);
        String line =null;
        StringBuilder sb = new StringBuilder();
        while((line = br.readLine()) != null){
            Log.d("response ", line);
            sb = sb.append(line);
        }
        String res = sb.toString();
        return res;
    }
```

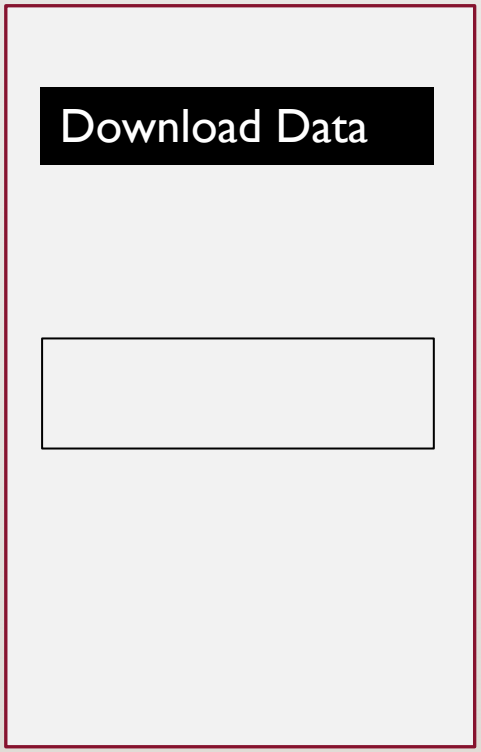
```
private void task(String url ) {

new Thread(new Runnable() {
    @Override
    public void run() {
        String res =    downloadData(url); // Run Task in a Background Thread
        runOnUiThread(new Runnable() {
            @Override
            public void run() {   // update the UI
                TextView textview = findViewById(R.id.textView);
                textview.setText(res);
            }
        });
    }
}).start();

}
```

;

TRIGGER DOWNLOAD ON CLICK

```
public void action(View v){
String urlStr
=“https://jsonplaceholder.typicode.com/todos”
   task(urlStr);
}
```

EXAMPLE



ADDING PROGRESS BAR

- Visual indicator for progress
- Add Progress Bar to improve user experience

REFERENCES

- https://developer.android.com/training/basics/network-ops/connecting.html
- **https://developer.android.com/training/basics/network-ops**