Enterprise development in practice.

Part 2. Understanding of http protocol (Hyper Text Transfer Protocol)

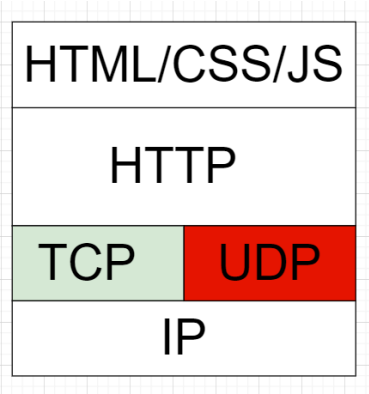
The key purpose of http protocol is to arrange communication between user (rather web browser) and server. The content of http protocol is html,css (as part of rendered ui) and js (javascript) or logic to be executed on frontend side.

Requirements of http protocol is described in RFC 2616. Main structure of http protocol is:

- URL or Uniform Resource Locator. For example: http://www.google.com
- Method Name. For example: GET or PUT, there are more exotic: HEAD, OPTIONS
- Request headers. For example: Keep-Alive: timeout=5, max=1000
- Request body. For example body in json format : "'id':'12'"

Protocol can be implemented on top of any protocol but last one has to be reliable (should have delivery acknowledgement). For example TCP can be used as base protocol meanwhile UDP – can't.

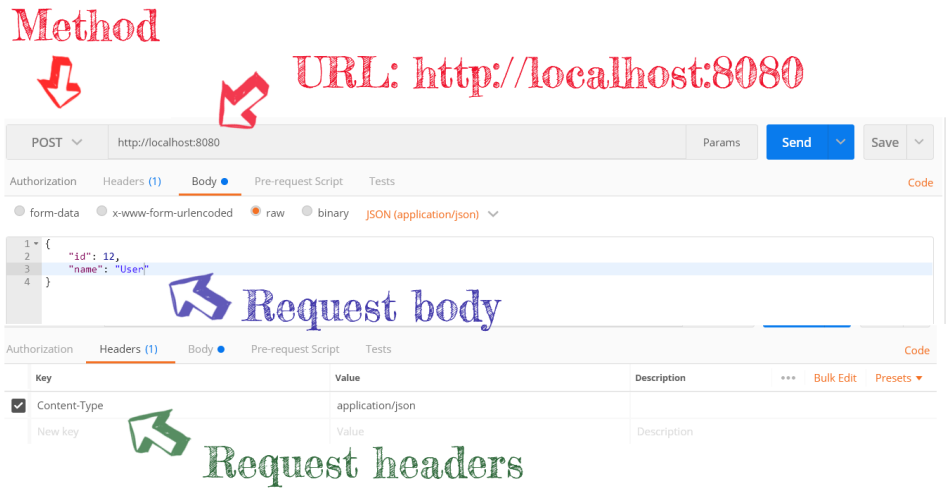
So if we take a look into layers, we can easlisy see that HTTP protocol belongs to application layer:



Making http request using postman tool.

There are many way to perform http request eg – open browser with specific url. In our example we will use postman plugin. We perform simple

request with next parameters:



Implementing simple TCP server using Java

In order to receive that request let's create a simple Java TCP server that listen specific port and print all incoming requests. Open `TCPServerWithHttpRequest` class and start the application. (change port according the port in postman)

```
public class TCPServerWithHttpRequest {
    public static void main(String[] args) throws Exception {
        ServerSocket serverSocket = new ServerSocket( port: 8282);
        Socket clientSocket = serverSocket.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
        String line;
        while (!(line = in.readLine()).isEmpty() ) {
            System.out.println(line);
        }
        in.close();
        clientSocket.close();
        serverSocket.close();
    }
}
```

Once postman request is performed you can see output in console:

So you can see that there is no magic and all http servers are nothing but tcp server that process http request according it's format.

Sending http response back to browser.

The main purpose of the browser is to render UI and execute JavaScript instructions. Now let's send request from chrome to our TCP server and return some simple response. Chrome will render received http response.

Open TCPServerWithHttpResponse class and start the server. From chrome tab perform localhost:8282 request. As result you will see rendered html that shows you youtube video.