



Network Programming
Section Four

Quiz

Essay Question

- 1) What is the Purpose of a Network?
- 2) Compare between proxy and vpn ?
- 3) Explain the main function of the following internetworking devices
 - a) hub
 - b) bridge
 - c) switch
 - d) router
 - e) gateway
 - f) wireless access point
 - g) Modem
- 4) Explain the Different Types of Networks Based on Geographical Area and Host Role ?

5) Given the following URL, identify and extract each of its components: the host name, protocol, port, path, query, and fragment.

- `https://www.example.com:78/enter/to/page?name=example&sort=asc#section2`

- **Protocol:**
- **Host Name:**
- **Port:**
- **Path:**
- **Query:**
- **Fragment:**

6) In Java, the `java.net` package is available for working with HTTP protocols, and the `HttpURLConnection` class is widely used to send HTTP requests and handle HTTP responses. Here is a basic example demonstrating how to make an HTTP GET request in Java.

❖ HTTP Method:

- defines the action the client wants to perform on the resource identified by the Request URI.
- Common HTTP methods include:
 - **GET**: for reading data.
 - **POST**: for sending data to create something new.
 - **PUT**: for updating data.
 - **DELETE**: for removing data.

❖ How to represent http protocol in java program

HTTP POST request in Java using the `HttpURLConnection` class. This example sends a POST request to a JSONPlaceholder API, which is a fake online REST API for testing and prototyping.

```
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
public class HttpPostExample {
    public static void main(String[] args) {
        try {
            // Create a URL object for the JSONPlaceholder API endpoint
            URL url = new URL("https://jsonplaceholder.typicode.com/posts");
            // Open a connection to the URL
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            // Set the HTTP request method to POST
            connection.setRequestMethod("POST");
            // Enable input and output streams for the connection
            connection.setDoInput(true);
            connection.setDoOutput(true);
            // Set the content type to JSON
            connection.setRequestProperty("Content-Type", "application/json");
            // Create the JSON data to send in the request body
            String jsonData = "{\n" +
                "  \"title\": \"Sample Post\",\n" +
                "  \"body\": \"This is a sample post request.\",\n" +
                "  \"userId\": 1\n" +
                "}";
            // Write the JSON data to the output stream
            try (OutputStream outputStream = connection.getOutputStream()) {
                byte[] input = jsonData.getBytes("utf-8");
                outputStream.write(input, 0, input.length);
            }
        }
    }
}
```

```

// Get the response code
int responseCode = connection.getResponseCode();
System.out.println("Response Code: " + responseCode);
// Read the response data
try (BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()))) {
    StringBuilder response = new StringBuilder();
    String line;
    while ((line = reader.readLine()) != null) {
        response.append(line);
    }
    System.out.println("Response Data:");
    System.out.println(response.toString());
}
// Close the connection
connection.disconnect();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```

HTTP PUT request in Java using the **HttpURLConnection** class. In this example, we'll update an existing resource on the JSONPlaceholder API:

```

import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
public class HttpPutExample {
    public static void main(String[] args) {
        try {
            // Create a URL object for the JSONPlaceholder API endpoint
            URL url = new URL("https://jsonplaceholder.typicode.com/posts/1");
            // Open a connection to the URL
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            // Set the HTTP request method to PUT
            connection.setRequestMethod("PUT");
            // Enable input and output streams for the connection
            connection.setDoInput(true);
            connection.setDoOutput(true);
            // Set the content type to JSON
            connection.setRequestProperty("Content-Type", "application/json");
            // Create the JSON data to send in the request body for the update
            String jsonData = "{\n" +

```

```

        "    \"title\": \"Updated Post Title\",\\n\" +
        "    \"body\": \"This is an updated post body.\\n\",\\n\" +
        "    \"userId\": 1,\\n\" +
        "    \"id\": 1\\n\" +
        "  }";

// Write the JSON data to the output stream
try (OutputStream outputStream = connection.getOutputStream()) {
    byte[] input = jsonData.getBytes("utf-8");
    outputStream.write(input, 0, input.length);
}

// Get the response code
int responseCode = connection.getResponseCode();
System.out.println("Response Code: " + responseCode);

// Read the response data
try (BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()))) {
    StringBuilder response = new StringBuilder();
    String line;
    while ((line = reader.readLine()) != null) {
        response.append(line);
    }
    System.out.println("Response Data:");
    System.out.println(response.toString());
}

// Close the connection
connection.disconnect();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```