Network Programming

Section four

- **Examples for http requests using different methods:**

  - Using DELETE method

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class HttpDeleteRequest {
    public static void main(String[] args) {
        try {
            String apiUrl = "https://jsonplaceholder.typicode.com/posts/1"; // Example DELETE API URL
            URL url = new URL(apiUrl);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("DELETE");
            int responseCode = connection.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP_NO_CONTENT) {
                System.out.println("DELETE request successful. Resource deleted.");
            } else if (responseCode == HttpURLConnection.HTTP_NOT_FOUND) {
                System.out.println("DELETE request failed. Resource not found.");
            } else {
                System.out.println("DELETE request failed. Response Code: " + responseCode);
                // If you want to read the error response, you can use similar logic as in previous examples.
                // BufferedReader in = new BufferedReader(new InputStreamReader(connection.getErrorStream()));
                // String inputLine;
                // StringBuilder response = new StringBuilder();
                // while ((inputLine = in.readLine()) != null) {
                //     response.append(inputLine);
                // }
                // in.close();
                // System.out.println("Error Response: " + response.toString());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- Using GET method

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class HttpPlainTextRequest {
    public static void main(String[] args) {
        try {
            String apiUrl = "https://example.com/text"; // Example URL for plain text response
            URL url = new URL(apiUrl);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            int responseCode = connection.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP_OK) {
                BufferedReader in = new BufferedReader(new
     InputStreamReader(connection.getInputStream()));
                String inputLine;
                StringBuilder response = new StringBuilder();

                while ((inputLine = in.readLine()) != null) {
                    response.append(inputLine);
                }
                in.close();

                System.out.println("Plain Text Response:");
                System.out.println(response.toString());
            } else {
                System.out.println("GET request failed. Response Code: " + responseCode);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
     }
```

Here is an example Java program that sends an HTTP OPTIONS request to a server:

```
import java.net.*;
import java.io.*;

public class HttpExample {
    public static void main(String[] args) throws Exception {
        URL url = new URL(apiUrl);
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("OPTIONS");
        BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()));
        String inputLine;
        while ((inputLine = in.readLine()) != null) {
            System.out.println(inputLine);
        }
        in.close();
    }
}
```

This program sends an OPTIONS request to `apiUrl` and prints the response to the console. You can modify this program to send a GET or POST request by changing the `setRequestMethod` method to `"GET"` or `"POST"` and writing the request body to the output stream returned by `con.getOutputStream()`.

- Using OPTIONS method

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;
import java.util.Map;

public class HttpOptionsRequest {
    public static void main(String[] args) {
        try {
            String apiUrl = "https://jsonplaceholder.typicode.com"; // Example API URL
            URL url = new URL(apiUrl);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("OPTIONS");

            // Optional: Set request headers if needed
            // connection.setRequestProperty("HeaderName", "HeaderValue");

            int responseCode = connection.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP_OK) {
                Map<String, List<String>> headers = connection.getHeaderFields();
                System.out.println("Options request successful. Response Headers:");
                for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
                    System.out.println(entry.getKey() + ": " + entry.getValue());
                }
            } else {
                System.out.println("Options request failed. Response Code: " + responseCode);
                // If you want to read the error response, you can use similar logic as in previous examples.
                // BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getErrorStream()));
                // String inputLine;
                // StringBuilder response = new StringBuilder();
                // while ((inputLine = in.readLine()) != null) {
                //     response.append(inputLine);
                // }
                // in.close();
                // System.out.println("Error Response: " + response.toString());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- **Getting the MAC address: The following method returns a string containing the MAC address for a NetworkInterface instance.**

```
public static String getMacIdentifier(NetworkInterface network){
    StringBuilder identifier = new StringBuilder();
    try{
        byte[] macBuffer =network.getHardwareAddress();
        if(macBuffer !=null){
            for(int i=0 ;i<macBuffer.length;i++){
              identifier.append(String.format("%02x%s",macBuffer[i],(i<macBuffer.length-1)?":":""));
            }
        }
        } catch (SocketException e) {

            e.printStackTrace();
        }
    return identifier.toString();
    }


   public static void main(String[] args) {
        try {
            Enumeration<NetworkInterface> networkInterfaces =
NetworkInterface.getNetworkInterfaces();
            while (networkInterfaces.hasMoreElements()) {
                NetworkInterface network = networkInterfaces.nextElement();
                // Check if the network interface is not a virtual or loopback interface
                if (network.getHardwareAddress()!=null) {
                    String macAddress = getMacIdentifier(network);
                    System.out.println("MAC Address of " + network.getName() + ": " + macAddress);
                }
            }
        } catch (SocketException e) {
            e.printStackTrace();
        }
    }
```