

## Lesson 06 Demo 02

### Implementing HTTP Client in Java

**Objective:** Implement the use of HTTP Client in Java 11 to efficiently send HTTP requests and handle responses

**Tools Required:** Eclipse IDE

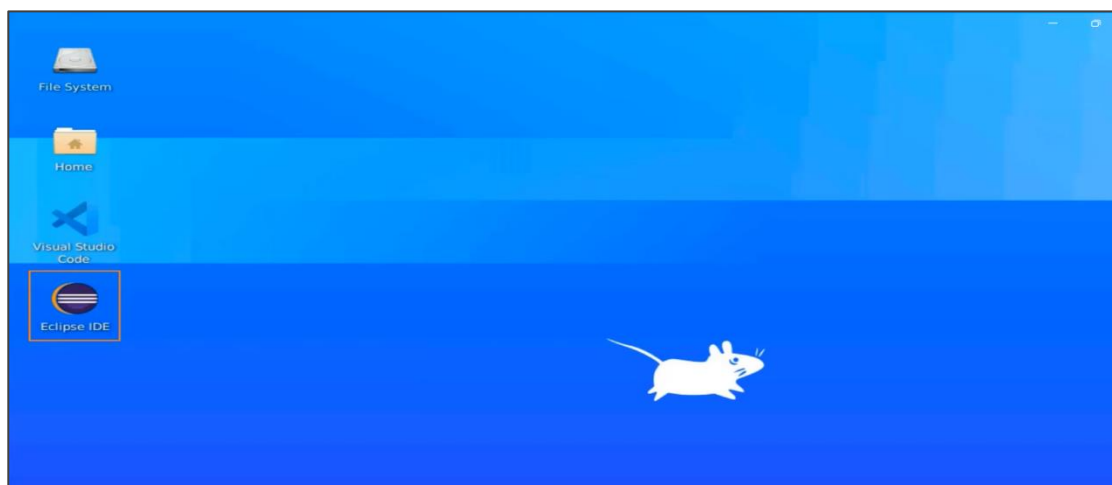
**Prerequisites:** None

Steps to be followed:

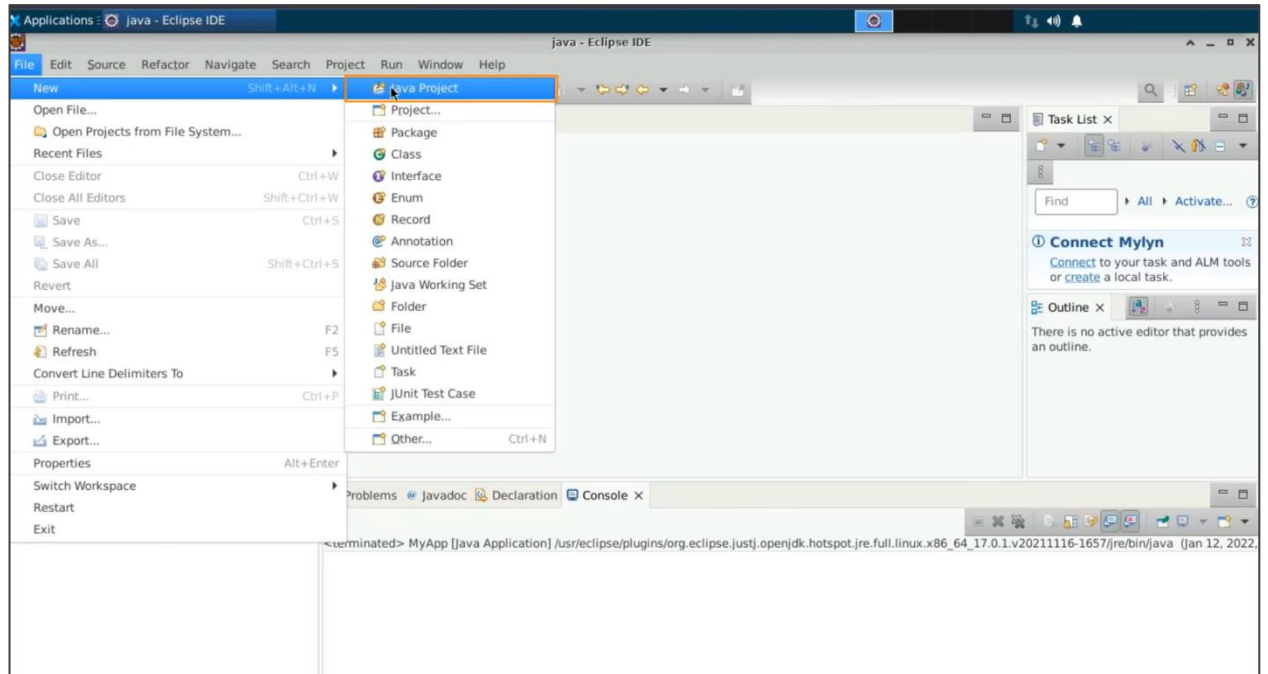
1. Open the Eclipse IDE and create a new Java project
2. Open a web browser and search for newsapi.org
3. Log in to your account to get an API key
4. Hit the URL to get the data in the form of a response
5. Record and concatenate the API key as a separate variable in the Eclipse IDE
6. Create HTTP client object, HTTP request object, and then the response objects
7. Execute the code

#### Step 1: Open the Eclipse IDE and create a new Java project

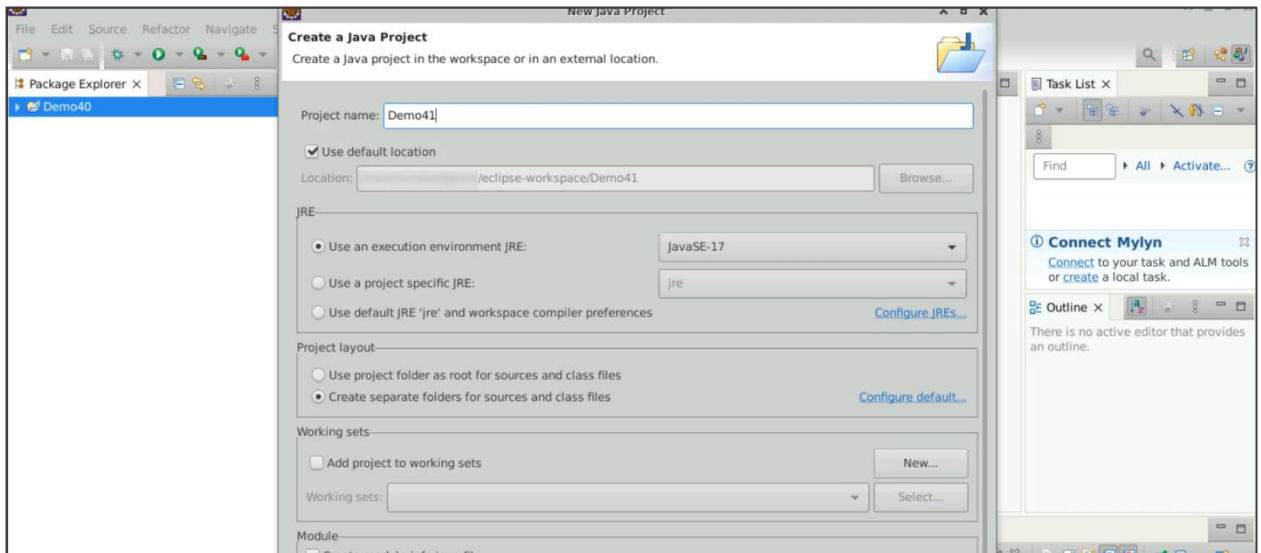
##### 1.1 Open the Eclipse IDE



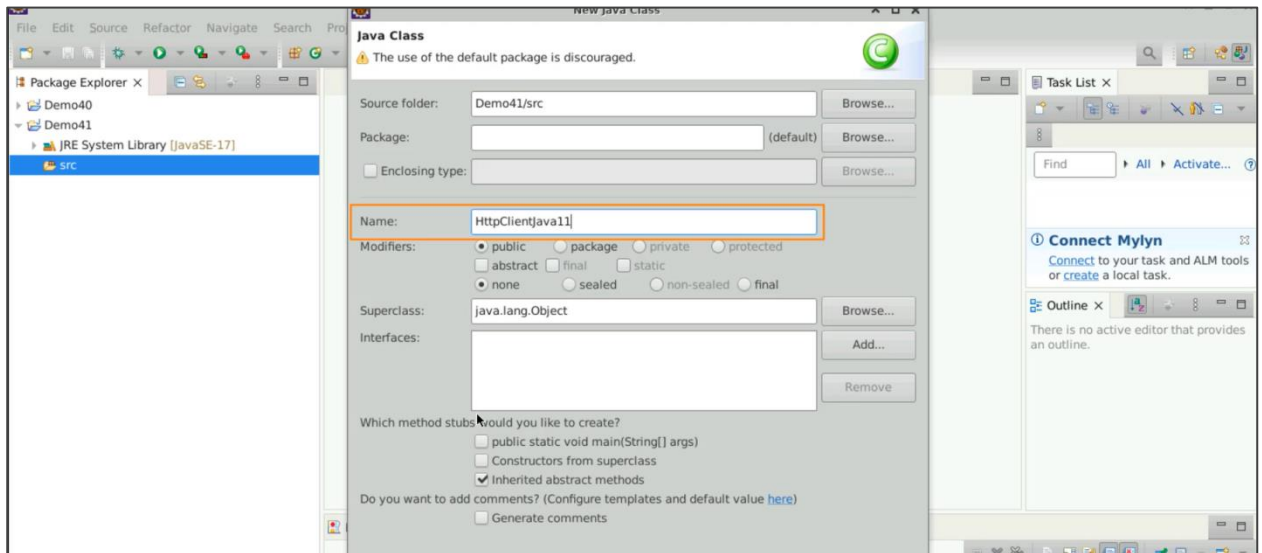
## 1.2 Select **File**, then **New**, and then **Java project**



## 1.3 Name the project Demo41, uncheck **Create a module-info.java** file, and press **Finish**

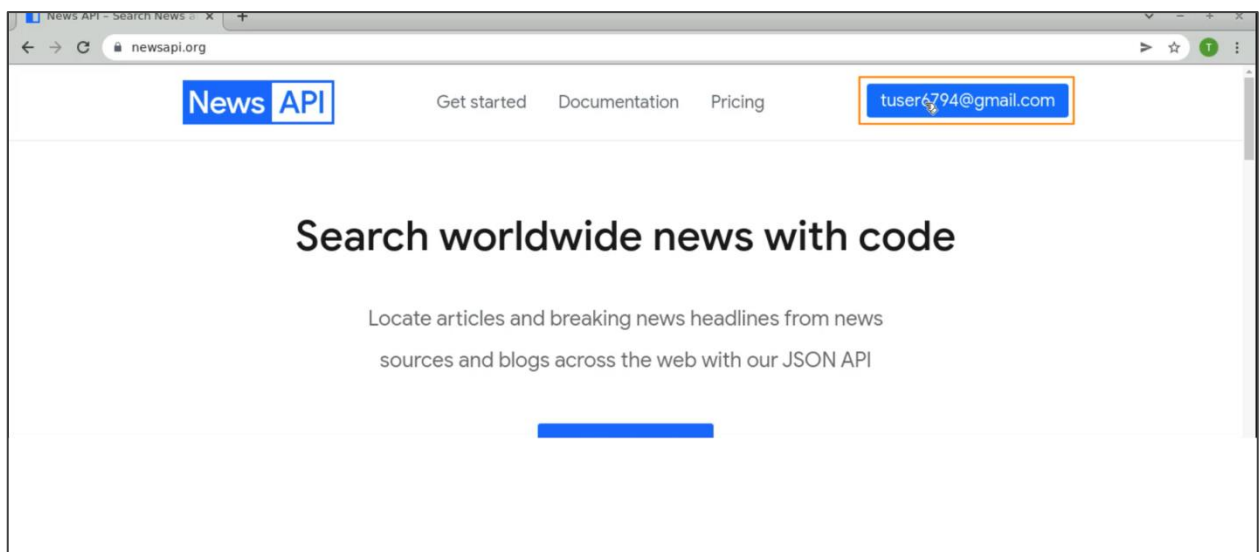


1.4 With **Demo41** selected, right-click on the **src**, and create a new class. Name this class **HttpClientJava11**, select the main method, and then select **Finish**



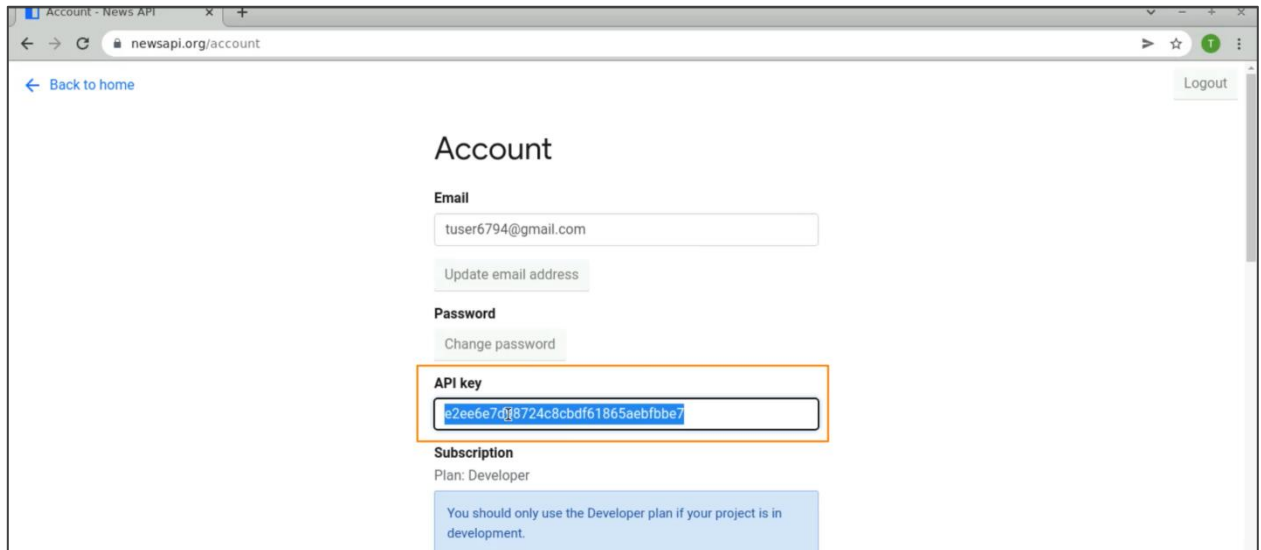
## Step 2: Open a web browser and search for newsapi.org

2.1 Open a web browser and search for **newsapi.org**. As you can see, it is already logged in here

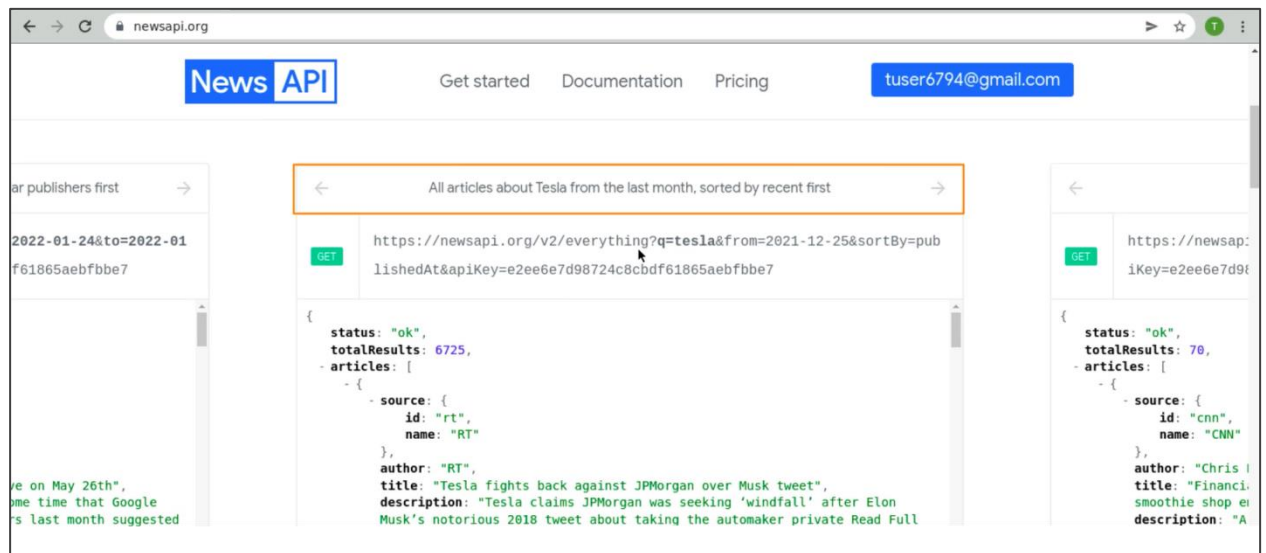


## Step 3: Log in to your account to get an API key

3.1 You need to log in to your account to get an **API key**.

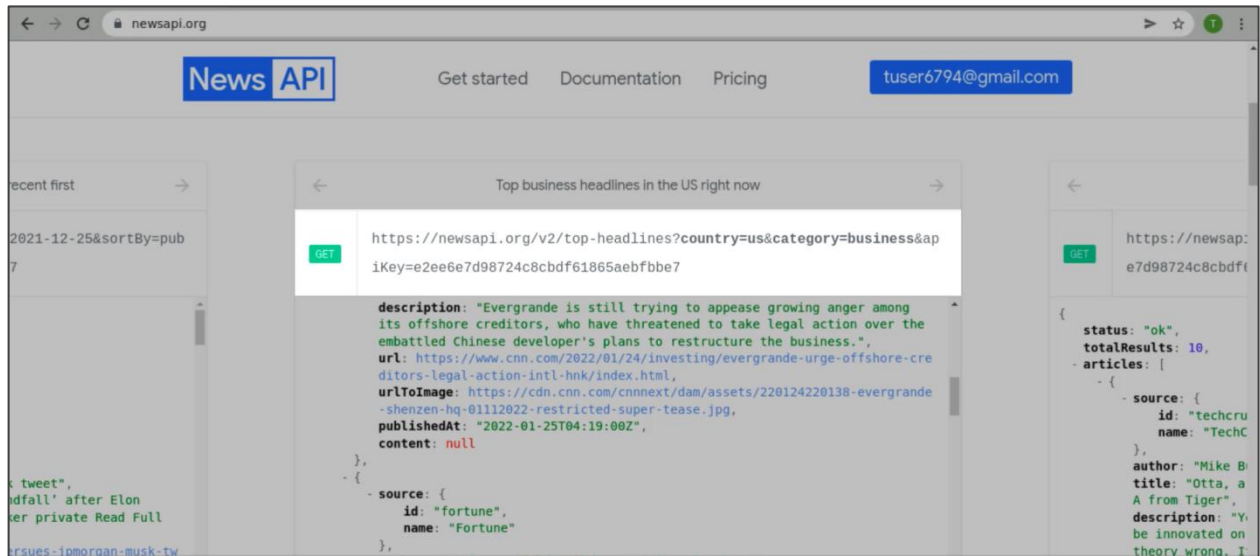


3.2 There are some standard URLs that can provide data, such as articles related to Tesla from the last month



## Step 4: Hit the URL to get the data in the form of a response

4.1 These URLs will have your API keys associated, and you can hit the URL to get data in the form of a response. First, copy this URL.

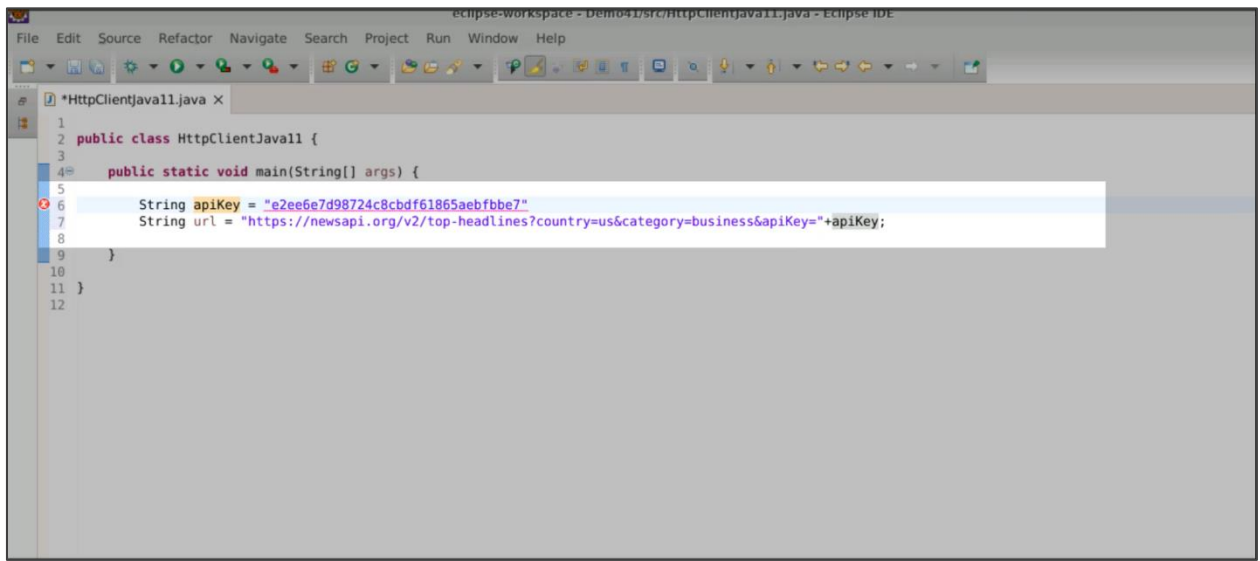


4.2 Paste it here. Now, when you hit this URL, you can see the JSON data comes as a response. The status is OK, and these are articles from the New York Times and so on.



## Step 5: Record and concatenate the API key as a separate variable in the Eclipse IDE

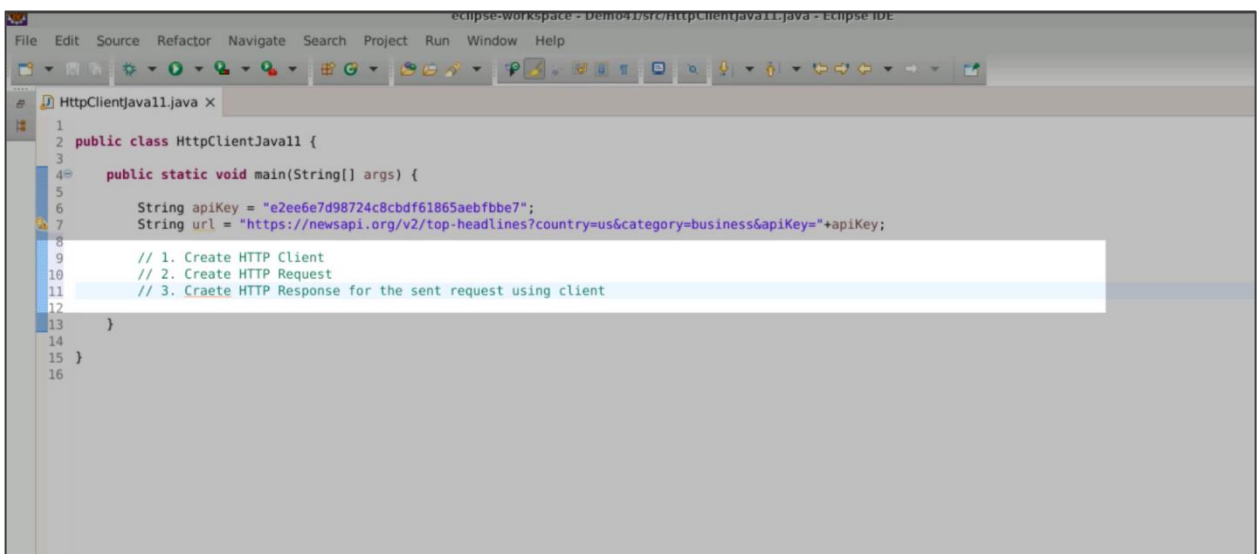
5.1 Let us create the URL endpoint. You have the news API URL coming in. As mentioned earlier, for the news API, you will have an API key provided by the website. Let us record this API key as a separate variable, and you can concatenate this API key here



```
eclipse-workspace - Demo041/src/Httpclientjavall.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*Httpclientjavall.java x
1
2 public class Httpclientjavall {
3
4     public static void main(String[] args) {
5
6         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbb7";
7         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
8
9     }
10
11 }
12
```

## Step 6: Create HTTP client object, HTTP request object, and then the response objects

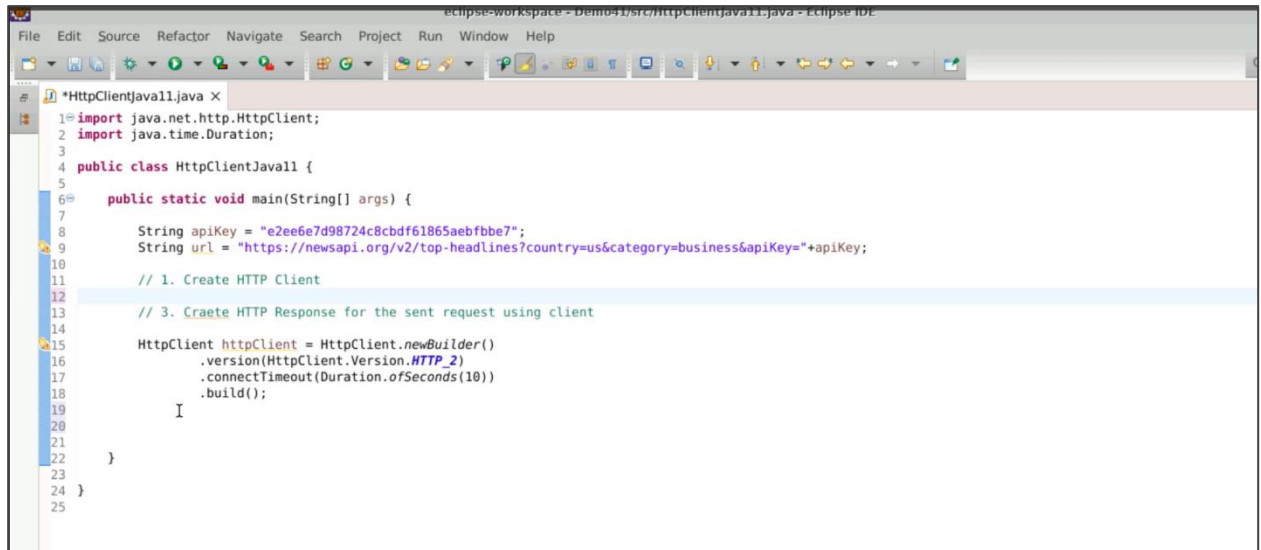
6.1 First, to send the request, you need to create an HTTP client object. Then, you will need to create an HTTP request object and the response objects. These are the three things you need to create in sequence. Let us follow these steps one by one:



```
eclipse-workspace - Demo041/src/Httpclientjavall.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Httpclientjavall.java x
1
2 public class Httpclientjavall {
3
4     public static void main(String[] args) {
5
6         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbb7";
7         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
8
9         // 1. Create HTTP Client
10        // 2. Create HTTP Request
11        // 3. Craete HTTP Response for the sent request using client
12
13    }
14
15 }
16
```



6.2 We can create an HTTP client using the java.net package's HTTP client API. It provides a builder for setting options like the version (e.g., HTTP/2) and timeout (e.g., 10 seconds) using the builder design pattern

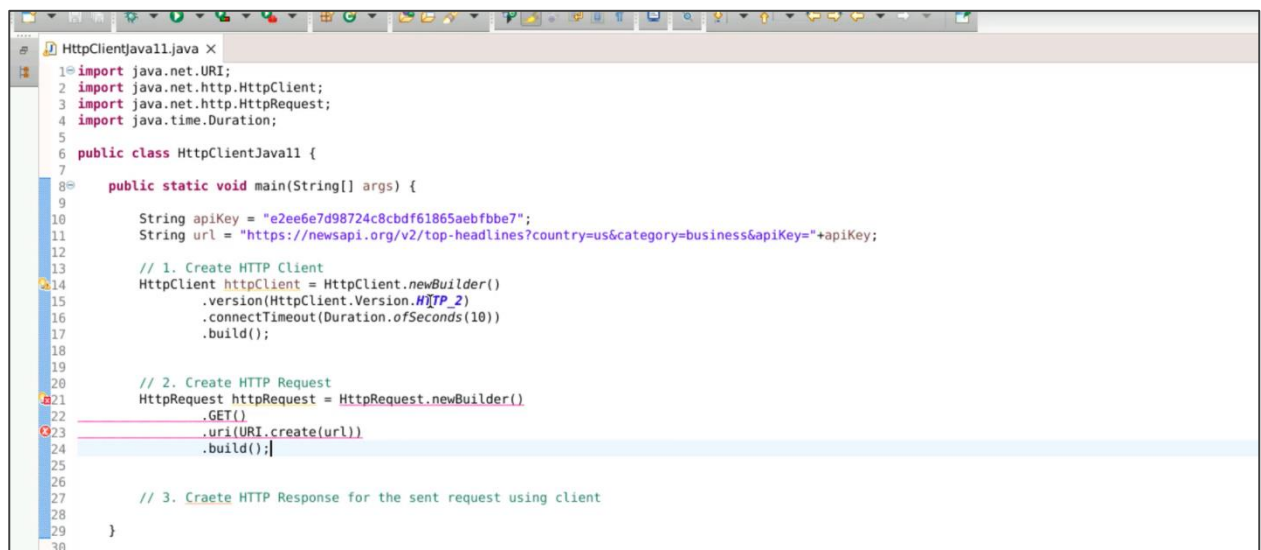


```

1 import java.net.http.HttpClient;
2 import java.time.Duration;
3
4 public class HttpclientJava11 {
5
6     public static void main(String[] args) {
7
8         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbbe7";
9         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
10
11         // 1. Create HTTP Client
12
13         // 3. Create HTTP Response for the sent request using client
14
15         HttpClient httpClient = HttpClient.newBuilder()
16             .version(HttpClient.Version.HTTP_2)
17             .connectTimeout(Duration.ofSeconds(10))
18             .build();
19
20
21     }
22 }
23
24
25

```

6.3 To send the request, we create an HTTP request object using the Builder API, specify the URI, and execute the **build** method. For a GET request without data, we use the **GET** method

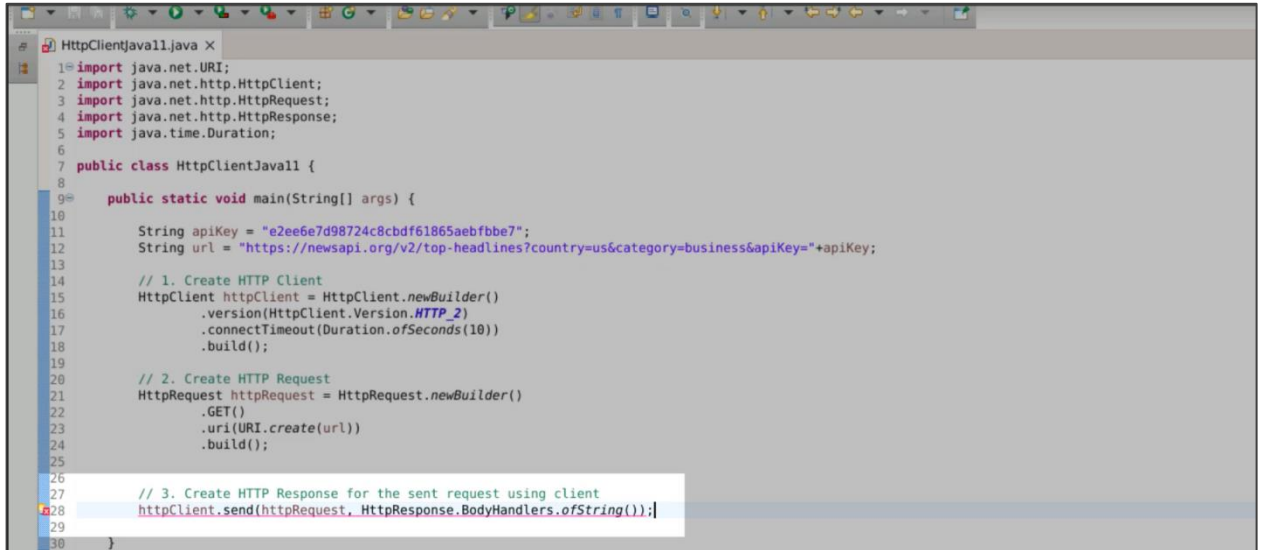


```

1 import java.net.URI;
2 import java.net.http.HttpClient;
3 import java.net.http.HttpRequest;
4 import java.time.Duration;
5
6 public class HttpclientJava11 {
7
8     public static void main(String[] args) {
9
10         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbbe7";
11         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
12
13         // 1. Create HTTP Client
14         HttpClient httpClient = HttpClient.newBuilder()
15             .version(HttpClient.Version.HTTP_2)
16             .connectTimeout(Duration.ofSeconds(10))
17             .build();
18
19
20         // 2. Create HTTP Request
21         HttpRequest httpRequest = HttpRequest.newBuilder()
22             .GET()
23             .uri(URI.create(url))
24             .build();
25
26
27         // 3. Create HTTP Response for the sent request using client
28
29     }
30 }

```

6.4 To handle the response, we use **HttpClient.send** with inputs for the body handler and the HTTP request. The **HttpResponse.BodyHandlers** API assists in handling the response, particularly for strings, using the **ofString** method

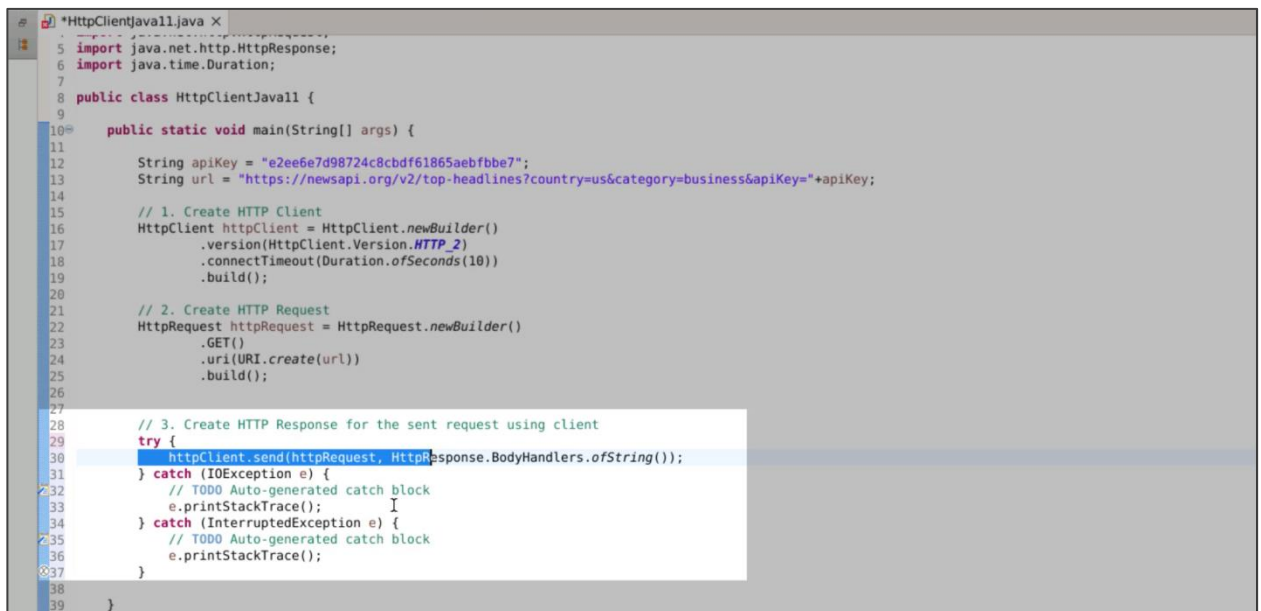


```

1 import java.net.URI;
2 import java.net.http.HttpClient;
3 import java.net.http.HttpRequest;
4 import java.net.http.HttpResponse;
5 import java.time.Duration;
6
7 public class HttpClientJavall {
8
9     public static void main(String[] args) {
10
11         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbbe7";
12         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
13
14         // 1. Create HTTP Client
15         HttpClient httpClient = HttpClient.newBuilder()
16             .version(HttpClient.Version.HTTP_2)
17             .connectTimeout(Duration.ofSeconds(10))
18             .build();
19
20         // 2. Create HTTP Request
21         HttpRequest httpRequest = HttpRequest.newBuilder()
22             .GET()
23             .uri(URI.create(url))
24             .build();
25
26         // 3. Create HTTP Response for the sent request using client
27         httpClient.send(httpRequest, HttpResponse.BodyHandlers.ofString());
28     }
29 }

```

6.5 Click on the error bulb. It shows that this code can throw an exception, and you should surround your code with a try-catch. Thus, surround with try-catch to handle exceptions



```

5 import java.net.http.HttpResponse;
6 import java.time.Duration;
7
8 public class HttpClientJavall {
9
10     public static void main(String[] args) {
11
12         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbbe7";
13         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
14
15         // 1. Create HTTP Client
16         HttpClient httpClient = HttpClient.newBuilder()
17             .version(HttpClient.Version.HTTP_2)
18             .connectTimeout(Duration.ofSeconds(10))
19             .build();
20
21         // 2. Create HTTP Request
22         HttpRequest httpRequest = HttpRequest.newBuilder()
23             .GET()
24             .uri(URI.create(url))
25             .build();
26
27         // 3. Create HTTP Response for the sent request using client
28         try {
29             httpClient.send(httpRequest, HttpResponse.BodyHandlers.ofString());
30         } catch (IOException e) {
31             // TODO Auto-generated catch block
32             e.printStackTrace();
33         } catch (InterruptedException e) {
34             // TODO Auto-generated catch block
35             e.printStackTrace();
36         }
37     }
38 }
39

```



6.6 Create the left-hand side for this expression, which is your HTTP response. The HTTP response will be of type String. This means you are ready with the JSON, but JSON is a string. Thus, you have your HTTP response

```

8 public class HttpClientJavall {
9
10     public static void main(String[] args) {
11
12         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbb7";
13         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
14
15         // 1. Create HTTP Client
16         HttpClient httpClient = HttpClient.newBuilder()
17             .version(HttpClient.Version.HTTP_2)
18             .connectTimeout(Duration.ofSeconds(10))
19             .build();
20
21         // 2. Create HTTP Request
22         HttpRequest httpRequest = HttpRequest.newBuilder()
23             .GET()
24             .uri(URI.create(url))
25             .build();
26
27         // 3. Create HTTP Response for the sent request using client
28         try {
29             // 30
29             HttpResponse<String> httpResponse = httpClient.send(httpRequest, HttpResponse.BodyHandlers.ofString());
30         } catch (IOException e) {
31             // TODO Auto-generated catch block
32             e.printStackTrace();
33         } catch (InterruptedException e) {
34             // TODO Auto-generated catch block
35         }
36     }
37 }

```

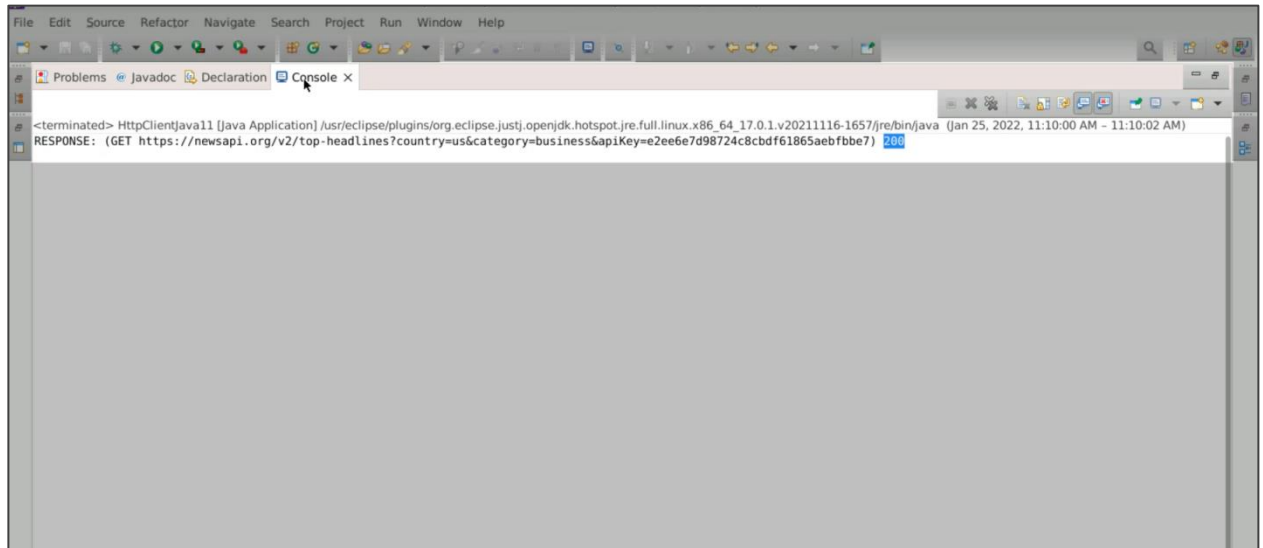
6.7 With the HTTP client, you can send a request and receive a response. We will print the response as it is, without any changes or formatting

```

File Edit Source Refactor Navigate Search Project Run Window Help
HttpclientJavall.java x
5 import java.net.http.HttpResponse;
6 import java.time.Duration;
7
8 public class HttpClientJavall {
9
10     public static void main(String[] args) {
11
12         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbb7";
13         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
14
15         // 1. Create HTTP Client
16         HttpClient httpClient = HttpClient.newBuilder()
17             .version(HttpClient.Version.HTTP_2)
18             .connectTimeout(Duration.ofSeconds(10))
19             .build();
20
21         // 2. Create HTTP Request
22         HttpRequest httpRequest = HttpRequest.newBuilder()
23             .GET()
24             .uri(URI.create(url))
25             .build();
26
27         // 3. Create HTTP Response for the sent request using client
28         try {
29             // 30
29             HttpResponse<String> httpResponse = httpClient.send(httpRequest, HttpResponse.BodyHandlers.ofString());
30             System.out.println("RESPONSE: "+httpResponse);
31         } catch (IOException e) {
32             // TODO Auto-generated catch block
33             e.printStackTrace();
34         } catch (InterruptedException e) {
35         }
36     }
37 }

```

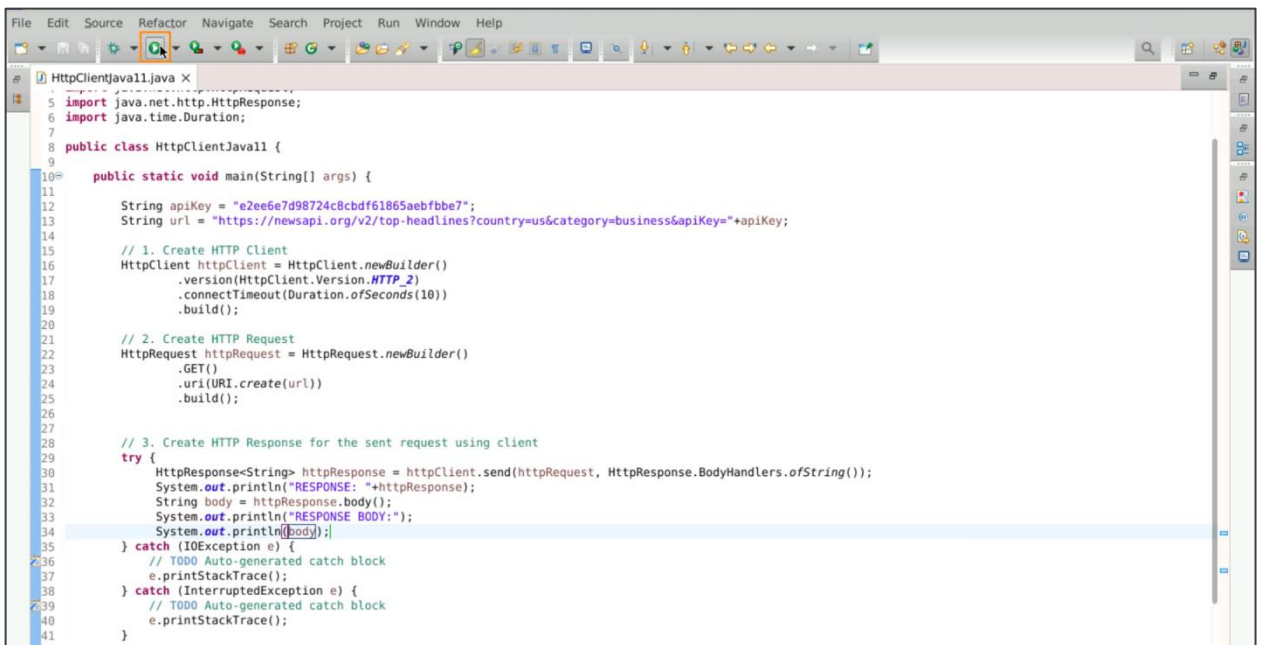
6.8 Let us run this code. As you can see, there is a response for URL get and you can note that the response code is 200, which means OK and that it's a successful request with the successful response



```

File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console X
<terminated> HttpClientJava11 [Java Application] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.1.v20211116-1657/jre/bin/java (Jan 25, 2022, 11:10:00 AM - 11:10:02 AM)
RESPONSE: (GET https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey=e2ee6e7d98724c8cbdf61865aebfbbe7) 200
  
```

6.9 Get the contents of the response, which is like a body. Write HTTP **response.body** and convert it to a string if you wish. However, when you call **response.body**, it is already a string. Therefore, print the body

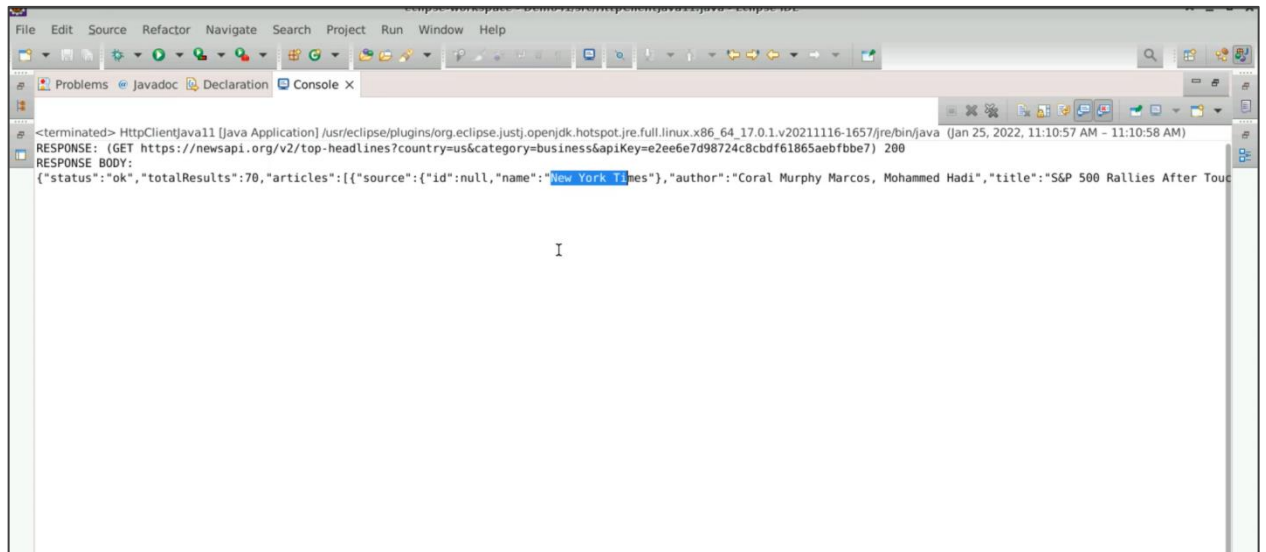


```

File Edit Source Refactor Navigate Search Project Run Window Help
HttpClientJava11.java X
5 import java.net.http.HttpResponse;
6 import java.time.Duration;
7
8 public class HttpClientJava11 {
9
10     public static void main(String[] args) {
11
12         String apiKey = "e2ee6e7d98724c8cbdf61865aebfbbe7";
13         String url = "https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey="+apiKey;
14
15         // 1. Create HTTP Client
16         HttpClient httpClient = HttpClient.newBuilder()
17             .version(HttpClient.Version.HTTP_2)
18             .connectTimeout(Duration.ofSeconds(10))
19             .build();
20
21         // 2. Create HTTP Request
22         HttpRequest httpRequest = HttpRequest.newBuilder()
23             .GET()
24             .uri(URI.create(url))
25             .build();
26
27         // 3. Create HTTP Response for the sent request using client
28         try {
29             HttpResponse<String> httpResponse = httpClient.send(httpRequest, HttpResponse.BodyHandlers.ofString());
30             System.out.println("RESPONSE: "+httpResponse);
31             String body = httpResponse.body();
32             System.out.println("RESPONSE BODY:");
33             System.out.println(body);
34         } catch (IOException e) {
35             // TODO Auto-generated catch block
36             e.printStackTrace();
37         } catch (InterruptedException e) {
38             // TODO Auto-generated catch block
39             e.printStackTrace();
40         }
41     }
42 }
  
```

## Step 7: Execute the code

- 7.1 Next, run the final code. As you can see, after the response, you have the response body which says the status is OK, and there are 70 articles found. This is the same content you see when you hit this URL in the browser



The screenshot shows the Eclipse IDE's console window. The console output indicates that an HTTP GET request was successfully executed to the URL `https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey=e2ee6e7d98724c8cbdf61865aebfbb7`. The response status is 200, and the response body is a JSON object. The JSON object contains a status of "ok", a totalResults of 70, and an array of articles. The first article shown is from the "New York Times" with the title "S&P 500 Rallies After Touchdown".

```
<terminated> HttpClientJava11 [Java Application] /usr/eclipse/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0.1.v20211116-1657/jre/bin/java (Jan 25, 2022, 11:10:57 AM - 11:10:58 AM)
RESPONSE: (GET https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey=e2ee6e7d98724c8cbdf61865aebfbb7) 200
RESPONSE BODY:
{"status":"ok","totalResults":70,"articles":[{"source":{"id":null,"name":"New York Times"},"author":"Coral Murphy Marcos, Mohammed Hadi","title":"S&P 500 Rallies After Touchdown"}]}
```

By following these steps, you have successfully implemented the use of HTTP Client in Java 11 to efficiently send HTTP requests and handle responses.