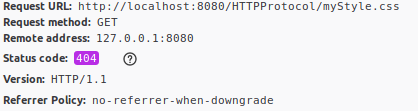
## 

## 1) Monitoring HTTP Headers 1

1. /HTTPProtocol/ - Cause: Document, Type: html
   1. 200 ok response
   2. Have sent the HTML site with “Hello World!” in the response body
   3. It’s a GET method
2. favicon.ico - Cause: img Type: x-icon
   1. 200 ok response
   2. Site have pointed out there's a favicon on site from the first HTTP request. Therefore we make this request to get the favicon.
   3. It’s a GET method

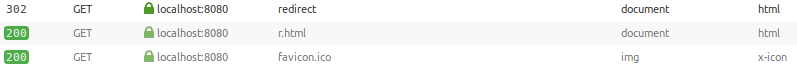
## 2) Monitoring HTTP Headers 2

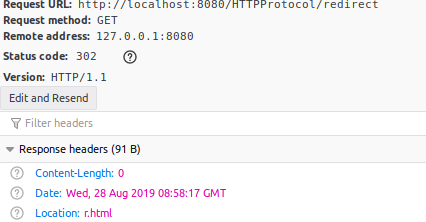




1. myStyle.css - Cause: stylesheet, type: html
   1. 404 not found response
   2. The stylesheet was not found on the requested URL, because it’s not been made, only linked to.
   3. The Request URL is the place to find the stylesheet file
   4. It’s a GET method
   5. Remote address is the ip and port of the server
2. IMG-20180914-144719.jpg - Cause: img, type: jpeg
   1. 304 not modified response
   2. The stylesheet was not found on the requested URL, because it’s not been made, only linked to.
   3. The Request URL is the place to find the stylesheet file
   4. It’s a GET method
   5. Remote address is the ip and port of the server

### 3) Monitoring HTTP Headers 3 (Response-codes 3xx)

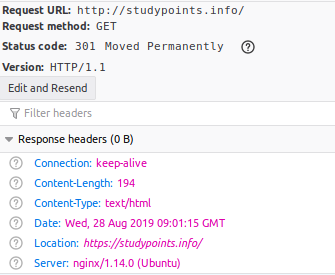




1. The response code 302 is a common way to redirect a page. The header information Location defines where to redirect to
2. The other one is just a plain HTML site request with a 200 ok response

### 3a) Redirecting to HTTPs instead of HTTP

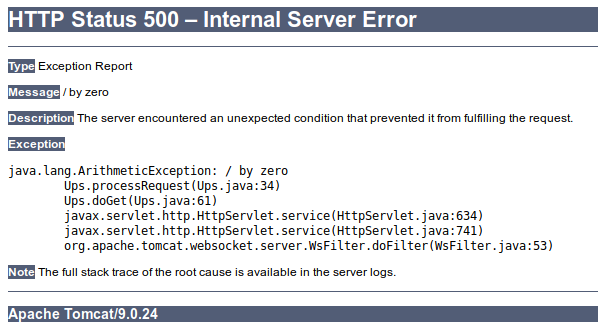




The http site has the status code 301 which means the site is moved to another site, therefore it should redirect to the location site.

### 4a) Status Codes (5xx)





The Ups has an status code 500 which means there's an error in the page, and the body contains an error message.

### 4b) Status Codes (4xx)



The status code 404 means the site is not found

### 4c) Status Codes - Ranges

Status 2: ok responses

Status 3: Redirect

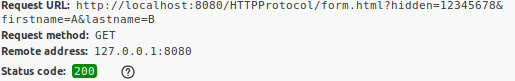
Status 4: Not found

Status 5: Error on page

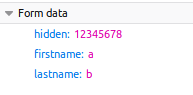
### 5) Get HTTP Request Headers on the Server

Created servlet: header

### 6) Get/Post-parameters



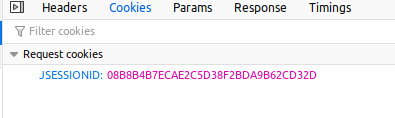
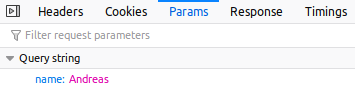
The form gets serialized and put into the URL like shown in the picture. Even the hidden value gets serialized



When posting the address field just has the action, because the fields are serialized to from data.

### 7) Sessions (Session Cookies)

When calling the SessionDemo servlet the session checks the local computer for a matching session cookie. If not found it will show the form where you can enter a name. When the name is submitted the session saves the name on the server and returns a session cookie for later matching. The cookie is saved locally in the browser, therefore if you close the browser the cookie will be deleted and the form will show up again.



### 8) Persistent Cookies

When using cookies the browser saves the information locally and not on the server. Therefore when you close the browser and open it again the cookie is saved. This way you can create functions like stay logged in. The server gets all the cookies and finds the cookie named username and uses it to call the functions.