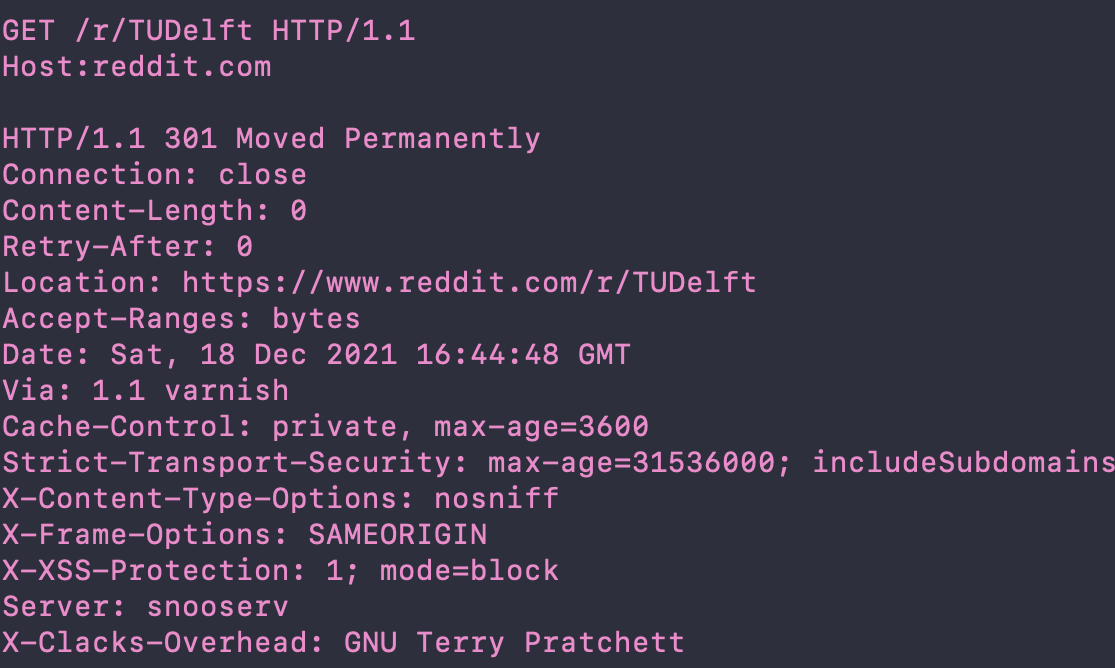
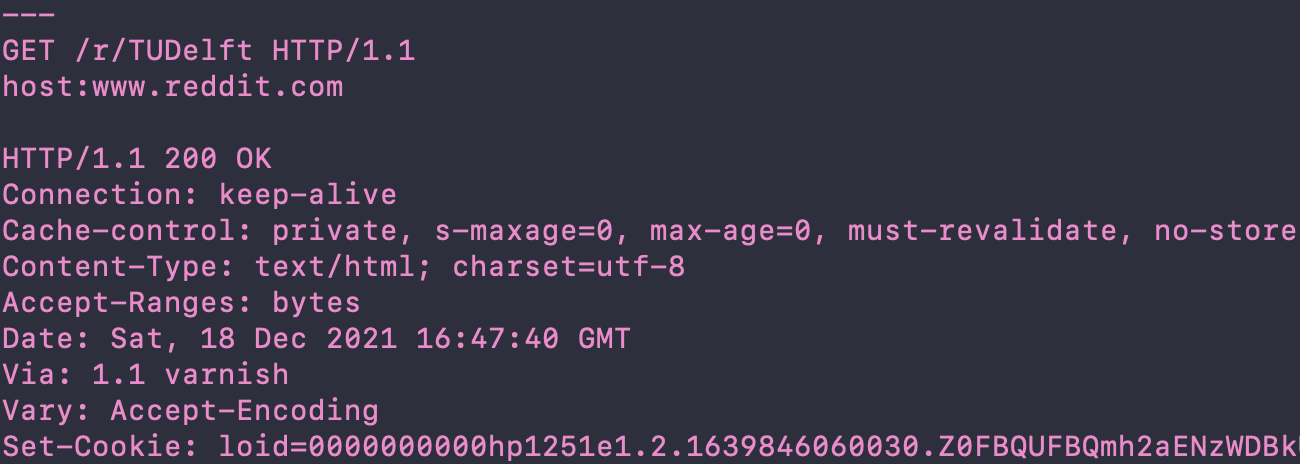
# Web Assignment 1

**1. HTTP request messages: GET/HEAD**

* **­** **openssl s\_client -connect reddit.com:443**
  + **GET /r/TUDelft HTTP/1.1**
  + **Host: reddit.com**

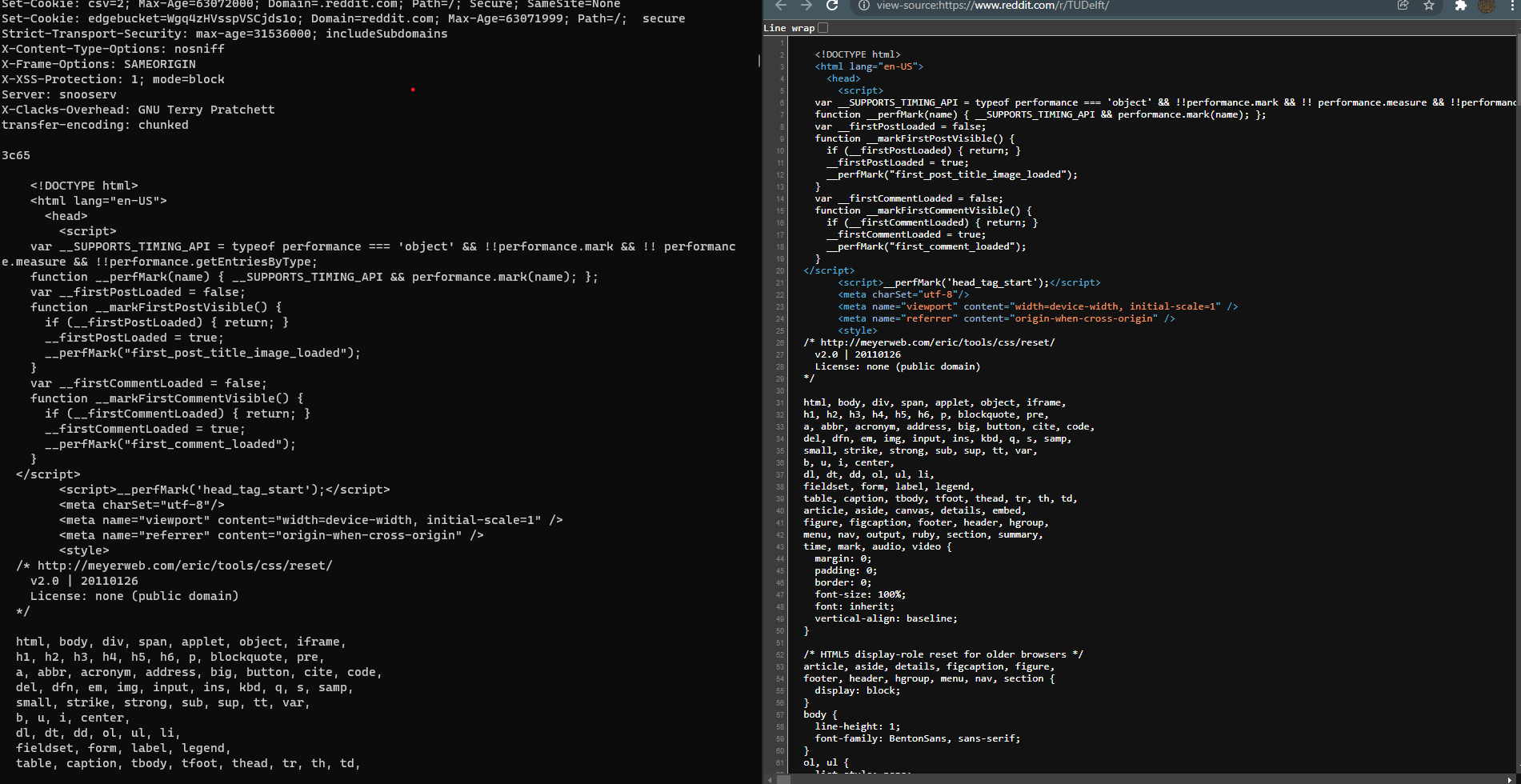
****

* **­** **openssl s\_client -connect www.reddit.com:443**
  + **GET /r/TUDelft HTTP/1.1**
  + **Host:** [**www.reddit.com**](http://www.reddit.com)

****

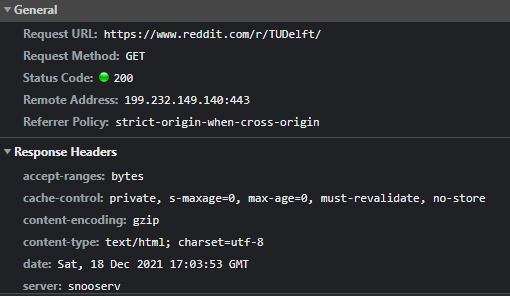
**Does the content you received correspond to what you see when accessing the resource with your browser?**

**Yes, it corresponds.**

****

**What does its Cache-Control header field mean?**

**Whenever a client asks for the resource, it must go to the server directly because the web caches cannot hold the resource as it expires immediately (this we can see from max-age=0).**

****

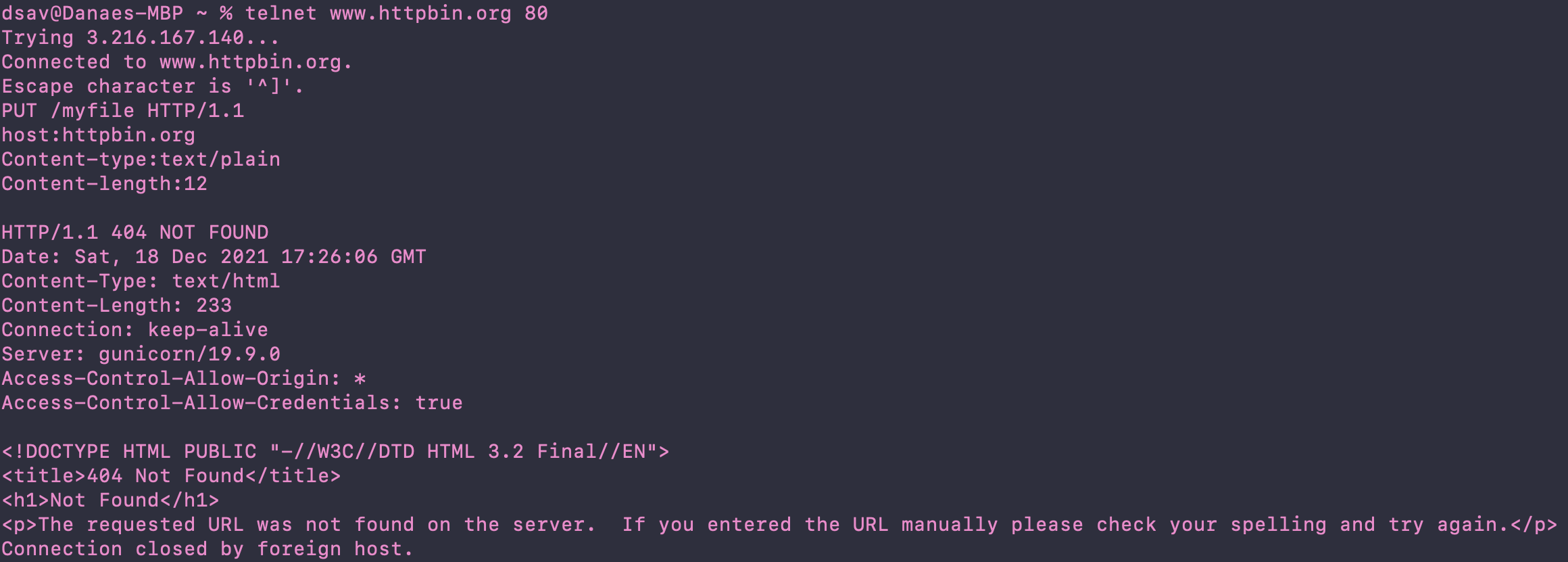
**If we stick to the resource considered in 1.3), what do we learn about the type of encodings your browser supports?**

**We receive the content-encoding field header which contains gzip meaning gzip is the best encoding that both the browser and the server support.**

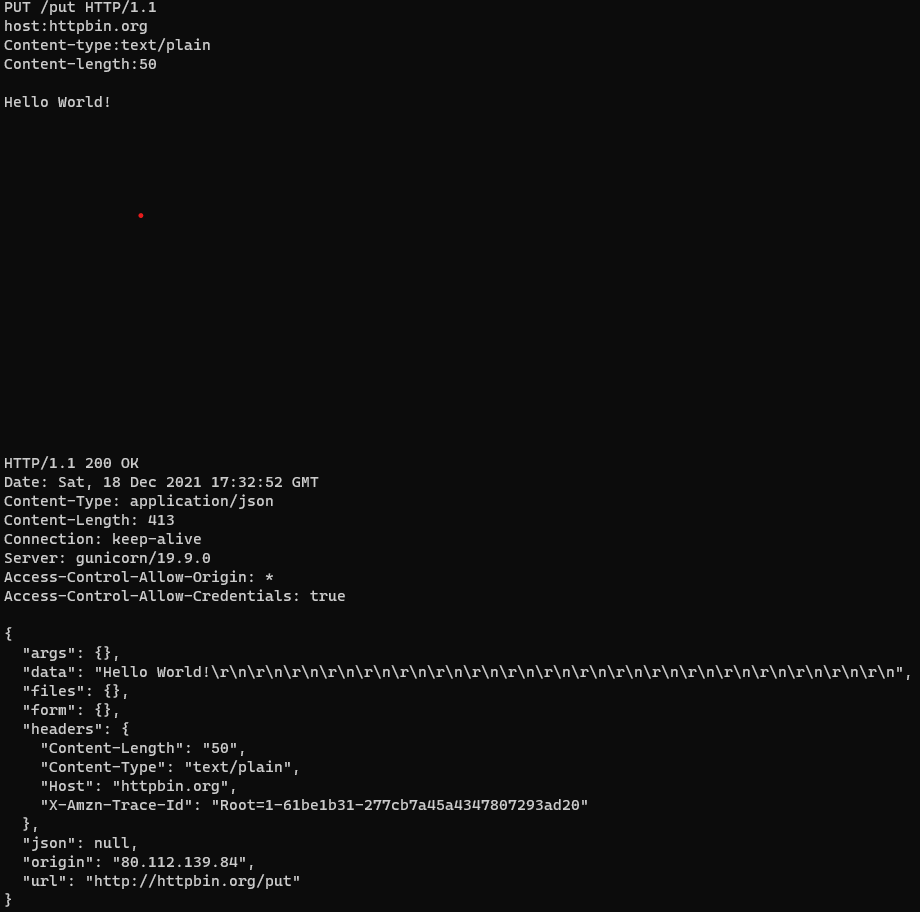
**2. HTTP request messages: PUT**

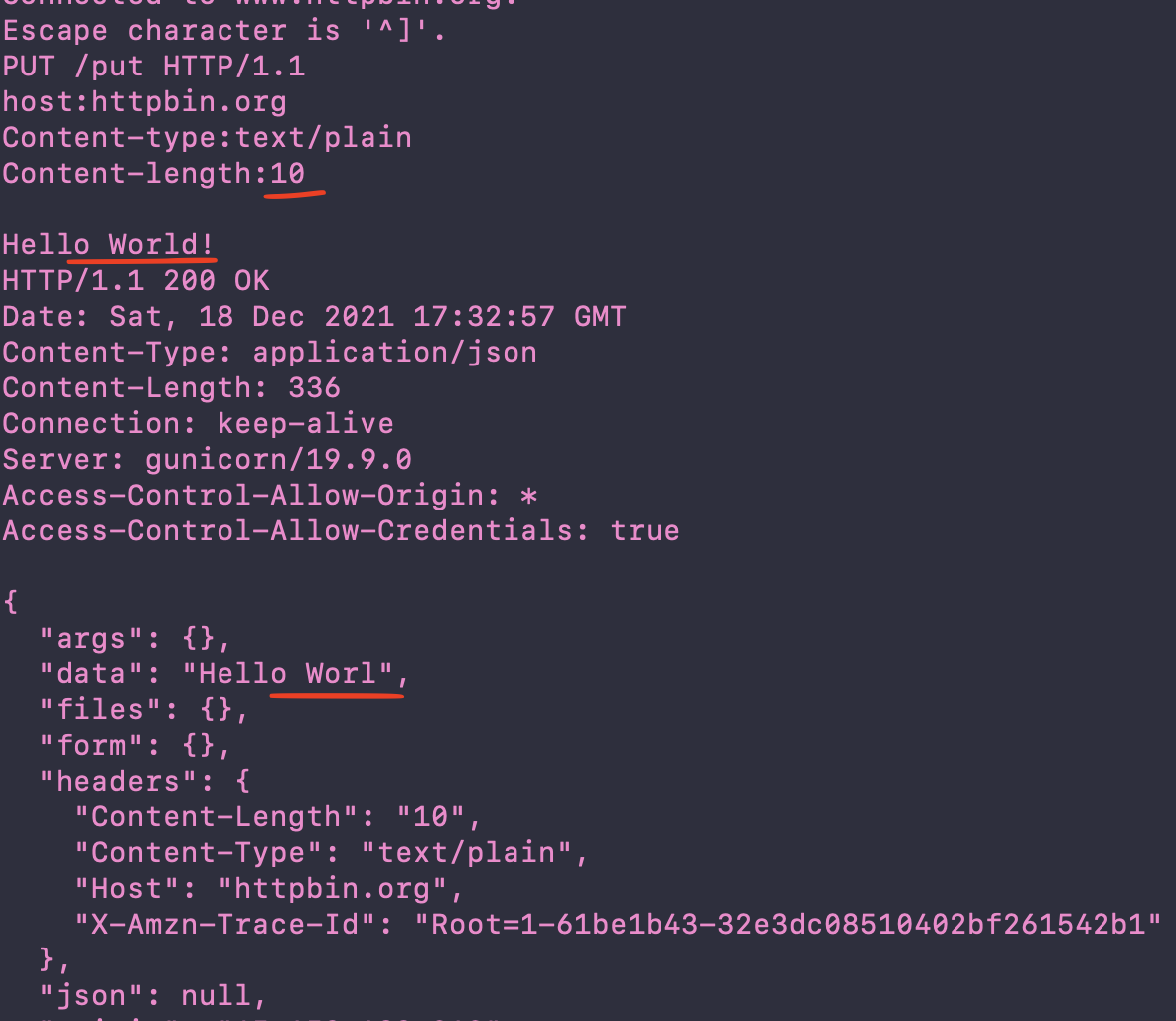
**What happens if you try to replace /put in this exercise with another resource (e.g. /myfile)? Does the httpbin.org server allow the creation of a new resource?**

**Creation of a new resource is not allowed. When trying to add a new resource it sends the request and returns this:**

****

**The Content-­length is exactly the number of characters (12 - we count the whitespace as well!) of Hello World!. What happens if the Content-length field is smaller or larger than the exact number of characters in the content?**

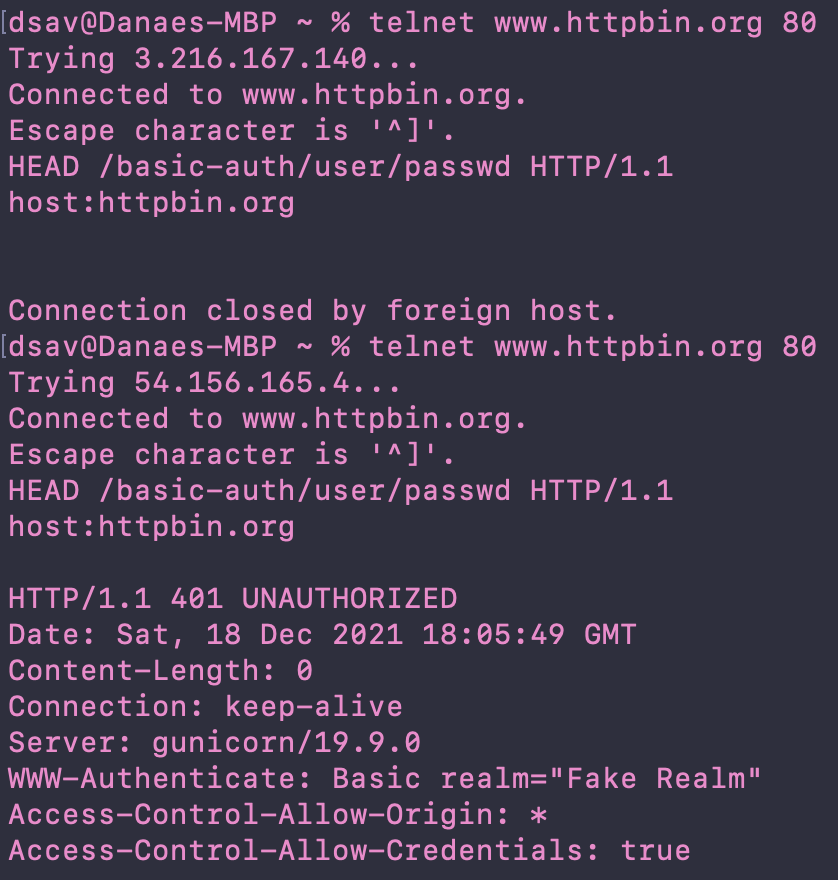
**When the content-length field is larger, we need to insert that many characters before the request can proceed. When the content-length is smaller, we can insert more characters but once the request is made, only the characters that are within the content-length specified will be shown in the data (participate in the request). **

****

**3. Basic Authentication**

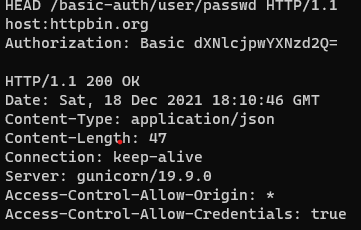
**3.1 Even tough HTTP is stateless, meaning it doesn't save any information passed on by past HTTP requests, we do not need to fill in the login details again. This is because after the first time of filling out the username and password the browser saves the information we entered and automatically uses that information in future HTTP requests with the same server.**

**3.2 We get UNAUTHORIZED**

****

**Request the same page without the Authorization header field—what happens? Is the behavior the same as reloading the page in the browser?**

**When inserting the encoded password and username we do get access (200 OK). But the behavior is not the same as reloading the page in the browser, since in the browser the authentication remains, whereas in this case when doing another request we will still need to provid the encoded username and password to gain access.**

****

**4. Web programming project: board game app**

**4.1) Memory game**

**4.2)**

[**https://webgamesonline.com/memory/**](https://webgamesonline.com/memory/)

[**https://www.improvememory.org/wp-content/games/magic-cards/index.html**](https://www.improvememory.org/wp-content/games/magic-cards/index.html)

[**https://www.improvememory.org/wp-content/games/memory-game/index.html**](https://www.improvememory.org/wp-content/games/memory-game/index.html)

**Design principles:**

The websites satisfy the rule of “Don’t make me think”, since how they work is pretty self-evident. Buttons and how the site is organized is very clear and all parts are divided well. However, there is too much clutter in a small area of the page, so the rule of “Minimize noise & clutter” is not followed. All the websites are self-evident as well as self-explanatory since the game is very simple, so the rule of “If you cannot make it self-evident make it self-explanatory” is followed. Aesthetics are acceptable, but outdated as of 2021.

**4.3)**

**Positive:**

1. **Well designed and functional.**
2. **Easy to use.**
3. **Animations make the game more interactive.**

**Negative**

1. **Sounds are very distracting**
2. **Site loads painfully slow-because of the animations.**
3. **Lack of information.**

**5)**

* **Think of a name for your application, a short description & a logo.**
* **Find images**