

Practical Sessions 1 – HTTP and graph representations

In your browser, open a new window. Open the developer tools (generally by right-clicking somewhere in the window and selecting ‘inspect’). Now go to the ‘network’ tab of the developer tools. This tab will show you all the HTTP requests carried out by your browser while you are using it. There might already be things there by you can ignore them for now.

Note: In several browsers, there is a small tick-box at the top of the network tab labeled ‘Disable Cache’. If you have it, make sure it is ticked, as it might otherwise make things a bit confusing in the following.

T1: With the network tab of the developer tools still open, go to the page www.loria.fr. In addition to the page displaying, you should see things happening in the network tab. Take a few minutes to understand what it all means.

T2: How many HTTP requests has the browser sent to load this page? Can you understand why it needs so many just for one page? Do you all have the exact same number?

T3: Click on the first request that was sent by the browser in the network tab. You should see information from the response header (what was sent back by the server) and from the request header (what was sent to the server by your browser). Can you understand what happened here? Did you get the same thing as your neighbor in all fields? What fields are different? Why? How much data was sent back?

T4: Click on the second request and try to understand what happened there.

T5: Still leaving the network tab open, click on one of the links on the page and inspect what happens in the network tab, what requests are sent, and what they might mean.

T6: Still keeping the network tab open, direct your browser to <https://data.open.ac.uk/course/t396> and see what happens.

curl is a terminal command, popular in particular on linux systems, to manually send HTTP requests. The site <https://reqbin.com/curl> allows you to use this command online, without having to have curl installed on your computer.

T7: Use curl to send an HTTP request for <https://data.open.ac.uk/course/t396>, using the command

```
curl "https://data.open.ac.uk/course/t396"
```

on the reqbin site. Note that this site automatically carries out the redirection. What it will show is equivalent to the second request from when you accessed that URI in your browser. In what format is the response from the server? Any idea why it is different from the results in the browser?

T8: Go back to your browser and copy the value of the Accept field of the request header from the first request to <https://data.open.ac.uk/course/t396>. With [BROWSER_ACCEPT] being the string you just copied, execute the following command on the reqbin site:

```
curl -H "Accept: [BROWSER_ACCEPT]" "https://data.open.ac.uk/course/t396"
```

Do you see the difference?

T9: Now obtain the RDF representation of the course in the turtle syntax using the following command:

```
curl -H "Accept: text/turtle" "https://data.open.ac.uk/course/t396"
```

T10: Copy this turtle representation in an editor and replace all URIs by prefixed versions.

T11: Draw the corresponding graph in a style similar to the one seen during the lecture (bubbles for resources, arrows for edges, etc.)