

## **F28WP Web programming Week 8**

### **Lab exercises**

#### **Exercise 1**

Using the module http, write a nodejs server application that listens on port 8080.

The server can be a simple app.js file within a folder that you create.

When the server receives a HTTP GET request with the URL /, it should send the string message "Hello from Node.js server on <your name>' computer".

Using a web browser, connect to <http://localhost:8080/> and check that you get the message.

Try to connect to your neighbour's server using their IP address <http://ip:8080/>

What happens?

#### **Exercise 2**

Within the same folder, create a simple HTML file (index.html) that contains a title and an image (any image you like). We assume that the image is in the file image.jpg located in the same folder. You can add any other content you like to the HTML file.

Modify your server application to send the file index.html when a GET request with the URL /site/ is received.

Connect to the server using <http://localhost:8080/site>. What do you notice?

Add a new route in your app.js to send the file image.jpg when a HTTP get request with URL /site/image.jpg is received by the server.

#### **Exercise 3**

Within the same folder, create a nodejs module (called videos.js). This module must have the function getVideos() that creates and returns an array of 3 or more YouTube videos that you like most. Each video is an object that has a title and a URL. You need to export the function getVideos.

Add a new route to your app.js to send the array of videos returned by getVideos() as a JSON object when the server receives a HTTP GET request with the URL /site/videos.

Test your server. What do you notice?

#### **Exercise 4**

Add an empty div to your index.html with the id "videos".

Add a button to your index.html. When this button is clicked, the web page sends an AJAX HTTP GET query to /site/videos. The associated call-back function must display the list of videos in the received JSON file as items of a bullet list. Each video will be an item with the title followed by a hypertext link to the corresponding YouTube URL.