

Workshop 11: Basics

DAT076/DIT126 Chalmers/Göteborgs Universitet
Joachim von Hacht

Objectives

Get some background: HTTP, web servers some tools and the basic client techniques HTML and CSS.

Resources

Always look for README-files in supplied code (zip-files). If present, possibly more instructions.

Slides and code samples from lecture 11.

[HTTP for web developers](#)

[MDN HTML](#)

[MDN CSS](#)

[W3Schools](#)

[curl](#)

Workshop assumes [NodeJS](#) and [npm](#) (Node package manager) installed system wide and in your [path](#). (if at home you have to install NodeJS, npm is included in NodeJS). Also, curl must be installed and in path.

HTTP

Some explorations of HTTP.

1. Do some checks (from command line)

```
$ node -v      (show version)
$ v6.0.0
$ curl --version
$ curl 7.33.0 ...
```

2. Download **l11.zip** from lectures page. Unzip.
3. Step into http_node directory execute

```
$ node server.js (server should start)
```

4. Use curl and supplied curl.txt to send request (paste into other shell).
Comment/uncomment in server.js as needed. Try all the verbs and more! Inspect requests and responses.

Node Modules

We will set the location of (user) global packages (= set the directory to store downloaded packages when using -g with npm). This is not the same as installing modules systemwide everything should end up in your home directory. See also [beginners guide](#).

Go to home directory. Execute:

```
$ cd && mkdir .node_modules_global (create directory for modules)
$ npm config set prefix=$HOME/.node_modules_global
$ npm config get prefix              (check where npm will store)
```

Now install npm user globally

```
$ npm install -g npm
```

Check in .node_modules_global/lib/node_modules and node_modules_global/bin

Add the .node_modules_global/bin to your path.

The http-server

Install a development web server built on node, the [http-server](#)

```
$ npm install -g http-server
```

Inspect .node_modules_global/bin

Test if server works

```
$ hs
Starting up http-server, serving ./
Available on:
  http://127.0.0.1:8080
  ...
Hit CTRL-C to stop the server
```

Node Web Application

Create some directory for an application. Add an any text file and an HTML page named index.html in directory. Start hs-server in directory. Use browser to request the files (i.e. make them show up in browser).

Apache Tomcat

Exploring Tomcat server.

1. Start NetBeans. Check tab Services > Servers to find Tomcat. If not present contact TA.
2. Mark Tomcat icon > right click > Start
3. Visit localhost:8080.

Permissions

Possibly need to edit permissions: Stop server and edit
.netbeans/8..../conf/tomcat-users.xml Add (as second last line)

```
<user username="admin" password="admin" roles="admin,admin-gui,manager,manager-script, manager-gui"/>
```

JEE Web Application

This assumes the Netbeans IDE is installed. If preferring some other IDE you have to do some setup on your own. To start Netbeans in STUDAT/Linux:

```
$ /chalmers/groups/ws-devel/netbeans
```

Open project "mysite" in NetBeans (in directory mysite in downloaded I11.zip file) File > Open Inspect!

1. In Projects tab, mark project icon > right click > Properties > Run. Check that selected server is Tomcat (else select Tomcat).
2. Mark project again > right click > Run. Index.html should show up in the system's default browser (set it to Chrome!).
3. Add any other file. Make it show up browser.
4. Inspect Services > Servers > Tomcat. Your application should show up.
5. Undeploy your application. Stop Tomcat.

There are two "default" application in Tomcat, shown in Netbeans, the "/" and "/manager" applications. Never touch!

A Static CV-site

Task is to create a “static” (i.e. non JavaScript) site for a webified CV of yours (or someone... don't need to be real facts, use imagination). You can use NodeJS style or JEE style. Choose one of!

If using JEE everything i done in NetBeans. If using Node some nice development tools: [Atom](#), [Brackets](#) (probably have to install). Both have looooooots of plugins! Explore!

1. If Node: Create a directory for the site, if JEE use a Maven Web application project (reuse from above or create an new: File > New Project > Maven > Web Application)
2. Add two HTML-pages. The below HTML-elements that must be used in the pages (others may be use). It should be possible to navigate between the pages.

<h1> - <h3>	Heading
<p>	Paragraph
<a>	Anchor
 & 	Unordered List & List Item
	Image
<div>	Division

3. Use any HTML validation service to validate the HTML.
4. Add some style using a separate CSS stylesheets. All layout and style should be handled by CSS
 - One HTML-page should use own layout and styles. Create a default.css file and add a few layout rules and a few style rules. Add id and/or class to HTML-elements to match the selectors (or match on element).
 - One pages should use the [Pure CSS framework](#) (recommended) or any other pure (non JavaScript) [CSS framework](#). All seems to have a grid system, use it!
 - It's possible to [install PureCSS](#) using [Bower](#). Then you first have to install [Bower using npm](#) (-g).
5. Must use some extra webfont (simplest is using [Google fonts](#)).
6. Optional: Try to use some [media queries](#). Resize screen to check!

Possible use of [Sass](#)

If you CSS framework is distributed as Sass files (.scss) have to compile to .css. Install node-sass globally

```
$ npm install -g node-sass  
$ node-sass style.scss style.css (compile scss to css)
```

Optional

Try the other approach.

Examination

Contact teaching assistant for a demo.