Første klient-server applikasjon – 2

Vi tar utgangspunkt i samme "hei-verden" løsning som over og legger inn en html-fil under katalogen **static** kalt **index.html**

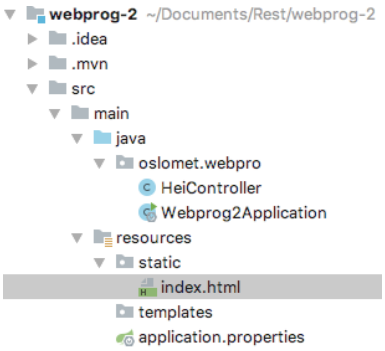
Denne filen skal kalle på metoden på server via et Ajax-kall. Ajax (Asynchronous JavaScript and XML) brukes enklest via JQuery som er et Javascript bibliotek. Dette biblioteket (en-fil) trenger vi å referere inn i html-filen. Filen med JQuery-koden kan enten lastes ned i løsningen og refereres der eller lastes fra en CDN (Content Delivery Node). En slik lenke kan f.eks brukes:

```
https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
```

index.html under / **src** / **main** / **resources** / **static** / blir slik:

```
<!DOCTYPE html>
<html lang="en">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></sc
ript>
<head>
    <meta charset="UTF-8">
    <title>Hei</title>
</head>
<body>
    <h1>Første webprogram</h1>
    <button id="knapp" onclick="hentData()">Trykk her</button>
    <div id="hallo">
    </div>
    <script type="application/javascript">
        function hentData() {
            $.get("http://localhost:8080/?navn=Per", function(data) {
                    $("#hallo").html=data;
                }
            );
        };
    </script>
</body>
</html>
```

Katalogstrukturen vil da se slik ut:



Når løsningen skal kjøres i nettleser oppgis følgende url:

localhost:8080/index.html

Da vises følgende i nettleseren:

Etter å ha trykket på knappen får man:

Virkemåte (**index.html**):

Først i html-filen har vi referansen til JQuery og Ajax via et Javascript:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
```

Så kommer det en knapp med en hendelse `onclick="hentData()`

```
<button id="knapp" onclick="hentData()">Trykk her</button>
```

Det betyr at når knappen trykkes skal javascript-funksjonen **hentData()** kjøres. Denne javascript funksjonen er så definert via dette javascriptet:

```
<script type="application/javascript">
    function hentData() {
        $.get("http://localhost:8080/?navn=Per", function(data) {
                document.getElementById("hallo").innerHTML=data;
              }
        );
    };
</script>
```

Først i scriptet er det angitt at dette scriptet er av typen **application/ javascript**. Dette er ikke strengt tatt nødvendig, men det er vanlig å dokumentere dette.

Deretter kommer selve funksjonen `hentData()`.

Denne inneholder bare en setning, selve ajax-kallet. Dette kallet starter med **$.get** som betyr av vi skal bruke JQuery sitt get-kall. $-tegnet brukes da for å indikere at dette er et JQuery kall og man trenger derfor referansen til dette JQuery-biblioteket først i filen.

Første parameter i **$.get**-kallet er url'en som skal kalles. Så kommer det en javascript-funksjon som skal kalles når **$.get**-kallet returnerer (svarer) fra server. Denne funksjonen mottar evt. data som blir sendt fra server i innparameteren **data** i funksjonen. I dette tilfelle vil serveren svare med "**Hallo verden Per"**. Denne strengen kommer da inn i variabelen **data**. Denne variabelen kan kalles hva som helst, her kalt **data**.

Denne funksjonen har så igjen klammeparenteser med javascript-kode for å legge det vi får fra serveren inn i den forhåndsdefinert -tagg med id = "hallo":

```
<div id="hallo">
</div>
```

Javascript-koden `document.getElementById("hallo").innerHTML=data;`

erstatter det som evt. skulle ligge inne i hallo-taggen (den er tom i utgangspunktet) med data-variabelen som inneholder svaret fra serveren.

Merk: Ajax-kallet er asynkront. Det betyr at kallet til `function(data)` ikke gjøres før svaret fra serveren kommer. Det betyr at all funksjonalitet som man ønsker å gjøre etter at svaret har kommet fra server må gjøres inne i denne funksjonen (i dette tilfelle overføre svaret til div'en**)**.

Merk: Vi ønsker å benytte oss av siste versjon av Javascript. Denne kalles ECMAScript 6. I Intellij har støtte for denne ved å velge: Preferences->Languages and Frameworks->Javascript. For å sette dette permanent : File->Other Settings->Preferences for new projects->Languages and Frameworks-> Javascript.