# Dokumentasjon NHMs samlingsportal

## Client side:

**search.js inneholder følgende funksjoner:**

download(filename, text)

downloadFunction()

drawMap(parsedData)

**download(filename, text)**

|  |  |  |
| --- | --- | --- |
| Beskrivelse | Laster ned søkeresultat til fil |  |
| Input | filename | Tekststreng, navn på fila som skal genereres |
|  | text | teksten som skal inn i fila, dvs søkeresultatet |
| Output | Nedlastet tab-separert txt-fil |  |

**downloadFunction()**

|  |  |  |
| --- | --- | --- |
| Beskrivelse | lager knappen «Last ned søkeresultat…» og funksjonalitet for denne |  |
| Input | - |  |
| Output | Last-ned-knapp med funksjon |  |

**drawMap(parsedData)**

|  |  |  |
| --- | --- | --- |
| Beskrivelse | Tegner kart med klikkbare markører for alle søketreffene |  |
| Input | parsedData | Søkeresultat i JSON format |
| Output | Kart med markører |  |

**objekt.js inneholder følgende funksjoner:**

loadString()

country(obj)

stateProvince(obj)

county(obj)

locality(obj)

coordinates(obj)

initializeMap()

**loadString()**

|  |  |
| --- | --- |
| Beskrivelse | Leser siste søkeresultat fra local storage og legger i en array. |
| Input | - |
| Output | En array med js-objekter som representerer museumsobjekter i siste søkeresultat.  Tom array hvis det ikke er noe søkeresultat i local storage. |

**country(obj)**

|  |  |
| --- | --- |
| Beskrivelse | Tilrettelegger for riktig formattert lokalitetsinformasjon for et museumsobjekt |
| Input | obj: et js-objekt som representerer det museumsobjektet som skal vises |
| Output | Tekststreng som representerer innsamlingsland med komma etter, eller tom tekststreng |

**stateProvince(obj)**

|  |  |
| --- | --- |
| Beskrivelse | Tilrettelegger for riktig formattert lokalitetsinformasjon for et museumsobjekt |
| Input | obj: et js-objekt som representerer det museumsobjektet som skal vises |
| Output | Tekststreng som representerer innsamlingsfylke med komma etter, eller tom tekststreng |

**county(obj)**

|  |  |
| --- | --- |
| Beskrivelse | Tilrettelegger for riktig formattert lokalitetsinformasjon for et museumsobjekt |
| Input | obj: et js-objekt som representerer det museumsobjektet som skal vises |
| Output | Tekststreng som representerer innsamlingskommune med komma etter, eller tom tekststreng |

**locality(obj)**

|  |  |
| --- | --- |
| Beskrivelse | Tilrettelegger for riktig formattert lokalitetsinformasjon for et museumsobjekt |
| Input | obj: et js-objekt som representerer det museumsobjektet som skal vises |
| Output | Tekststreng som representerer innsamlingslokalitet, eller tom tekststreng |

**coordinates(obj)**

|  |  |
| --- | --- |
| Beskrivelse | Formatterer koordinatene til fint lesbart format |
| Input | obj: et js-objekt som representerer det museumsobjektet som skal vises |
| Output | Tekststreng med objektets koordinater |

**initialize\_map()**

|  |  |
| --- | --- |
| Beskrivelse | Lager et kartobjekt |
| Input | - |
| Output | Et kartobjekt |
| Bruker | OpenLayers <https://openlayers.org/> |

## Server side

Rammeverktøy er NodeJS og Express

<https://nodejs.org/en/>

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

**Express**

<https://www.npmjs.com/package/express>

Fast, unopinionated, minimalist web framework for node.

## Dependencies

**Hbs**

<https://www.npmjs.com/package/hbs>

Express.js view engine for handlebars.js

**Path**

<https://www.npmjs.com/package/path>

This is an exact copy of the NodeJS ’path’ module published to the NPM registry.

### DEV:

**nodemon**

<https://www.npmjs.com/package/nodemon>

nodemon is a tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected.

## Web server

Express brukes som webserver for portalen. Dette er en liten og lett web server som passer utmerket til vårt bruk.

Web-serveren har følgende endepunkter:

|  |  |  |
| --- | --- | --- |
| Endepunkt | Returnerer | input |
| App.get('', (req, res) …) | Index.html | ingen |
| app.get('/search', (req, res) …) | **Unparsed.results**  Søkeresultater, som en tabseparert variable | http://localhost:3000/search/  ?search=' + searchTerm +'&samling=' + chosenCollection |
| app.get('/download', (req, res)…) | **Unparsed.results**  Søkeresultater, som en tabseparert variable | http://localhost:3000/search/  ?search=' + searchTerm +'&samling=' + chosenCollection |
| app.get('/footer-date', (req, res)...) | **date: date**  sist endret dato for datafila | http://localhost:3000/footer-date/?&samling=' + chosenCollection |
| app.get('/object', (req, res)…) | **myObject: req.query.id**  musit nummeret | http://localhost:3000/object/?id=${i} {data[i].catalogNumber} |
| app.get('/about', (req, res)…) | About.html |  |
| app.get('/help', (req, res)…) | Help.html |  |
| app.get('\*', (req, res)…) | 404.html  File not found | Plukker opp alle ugyldige nettadresser |

## Funksjoner

## Fil: footerDate.js

getFileUpdatedDate = (samling, callback) => {…}

**Input**:

text string fra nedvalgsliste, {karplanter; Sopp, Moser; Lav; Alger; Insekter}

**Retur**:

Leser filedatoen fra datafila som blir valgt og returnerer dato slik at den vises som last updated

## Fil: fileread.js

setCollection = (samling) => {…}

**Input**:

et valg fra droppdown feltet for samling

return musitFile

Oversetter samling til filnavn.

search = (samling, searchTerm, callback) => {… }

**input**:

samling (som oversettes til musti fil i funksjonen setCollection)

seachTrem =som er det brukeren søker på, kan være 2 ord skilt av space

**retur**:

results, en tabseparert variabel med linje for linje med resultatene