

# Course Assignment

- Level 1 is required
- Level 2 is optional

Choosing appropriate variable and function names will form part of your assessment, as will proper and consistent formatting of your code.

## Level 1 Process - Find an API

Search for a public, free-to-use API.

You will need to make two calls to this API:

- to fetch an array of results
- to fetch a single result using an id, name, or other property.

You will need to read the API's documentation to see what URLs are available, if they require a key to be sent in the header, and any other configuration they might need.

## ...Level 1 Process - Find an API

There are many free APIs discoverable by a Google search.

You may not use the APIs used in the lessons.

## Level 1 Process - Styling

The focus of the CA is on JavaScript, not styling. Yet, as a front-end developer you will always need to produce user-interfaces that make sense and are easy to follow. You will need to provide navigation to and from the home page (`index.html`) and the contact page.

Both API calls should include a loading indicator.

Some CSS to make sense of objects on your page should be included.

## Level 1 Process - index.html

Make a call to your API URL. Loop through the results and create HTML for each result.

You must display at least 3 different properties inside the HTML. It's not required to display an image.

You will need to link each result to a details.html page and to pass a parameter in the query string to that page.

If you are going to fetch the individual result on the details page by its id, then pass an id in the query string.

## ...Level 1 Process - index.html

If you will be retrieving by another property, like name for example, pass the name in the query string.

You will fetch this parameter from the query string in the details page code.

Catch any errors and display a message on the page if an error occurs.

## Level 1 Process - details.html

(Remember, you will need a parameter in the query string on this page, so either click through to it from the index page or manually add a parameter to the URL).

Retrieve the id, name or other parameter from the query string.

Once you have the parameter, add it to the API URL in the format the API requires.

Make an API call using the URL you create.

## ...Level 1 Process - details.html

Display at least 3 different properties from the received JSON on this page.

Set the title of the HTML page to be one of the property values, like name, title or another relevant property.

Catch any errors and display a message on the page if an error occurs.



## Level 1 Process - contact.html

Create a form with the following inputs and validation rules.

- Name - required
- Subject - must have a value with a minimum length of 10
- Email - must have a value and be formatted like an email - -
- Address - must have a value with a minimum length of 25

When the form on this page is submitted, write code to validate the input. If any of the inputs fail validation display an error message for the relevant input.

## Level 2 Process - contact.html

If all validation passes, add a message above the form indicating the form passed validation.

## Information - Rules

- Sharing APIs and copying and sharing of any code will result in your assignment being given a mark of zero.
- You may only use plain JavaScript for this assignment, no libraries or frameworks. You will be given a mark of zero if you use a library or framework for your JavaScript code.

# Information - Submission

- Create a folder called your-name-js1-ca, e.g. eivind-holvik-js1-ca
- Add all your code to this folder, zip the folder and submit the zip file
- No report is required
  - If something needs to be specified, please comment your code
- Only one zip file is to be submitted
- The next slide illustrates how the zip should be structured:

# Information - Submission .zip example

```
eivind-holvik-js1-ca.zip
  index.html
  something.html
  something-else.html
  css
    styles.css
    maybe-some-other-css.css
  js
    script.js
    something.js
    something-else.js
  resources
    some-image.jpg
  other
    //other relevant files
```

# Information - Time

- The course assignment is due monday 28.02.22 at 23:59
- Estimated workload 40 hours