

05.01.02: Dynamic Restaurant Site

Description

Create a dynamic website for the imaginary restaurant *Petrograd*.

You have already made a static website for it, and now it's time to make it dynamic. The content must be fetched from <http://kea-alt-del.dk/t5/api/>

You are NOT supposed to download the data, but rather, request it whenever needed.

The site should show data from the service, and dynamically reflect the state of the data, meaning, if a product is marked as "sold out", the website should reflect that.

No products, not delivered by the service, should be shown on the website, and all products given by the service must be shown.

The assignment is intended to train you in creating a design, adjust it to dynamic, unknown data, and implement the code to show the dynamic content.

Requirements

Looks:

You should base your design on what you developed last wednesday, but you are more than welcome to adapt and adjust as needed.

There must be a list of products, ordered into categories.

When a product is clicked, it must display additional information (for instance, in a modal dialog). This is what is referred to in the chapter "Product details"

Be advised, that the API does not give you all needed information in the product list, you must make an additional call to the API when displaying additional information.

Product details

How you show the product details is up to you, but the following should be shown as a minimum:

- Name
- Image
- Long Description
- Price
- Sold out status
- Discount price
- Whether it's suitable for vegetarians

Data:

You must fetch the data from <http://kea-alt-del.dk/t5/api> - use <http://kea-alt-del.dk/t5/api/productlist> to get the list of products, and <http://kea-alt-del.dk/t5/api/product> with {id: id} to get information about a single product

Implementation:

HTML, CSS and JavaScript must be in separate files.

Sorting / ordering

The product list must be ordered into categories

Optional

The rest is optional, and is just a few ideas to get you up and running with more JS goodness

Filtrering

Make the product list filterable, so the user, for instance, can choose to see only courses with a discount, vegetarian etc.

Advanced sorting

Make the product list sortable, so the user can sort from cheapest to most expensive or similar