# Data

All data is stored in .js files in the /data folder. The format is an array delimited with [] brackets with object that are separated with {} brackets. The example below contains two object with two properties each, ‘title’ and ‘value’. The files may be opened in any simply text editor such as notepad.

*var NAME = [{title: ‘first, value: ‘11’}, {title: ‘second’, value: ‘22’}]*

## Product Data

The product data the system uses is stored in the file named: “ProductData.js” in the ./data folder. The format is CSV JSON data after the variable named PRODUCTS. The file has to contain “var PRODUCTS = [{data}] where the [] encloses the array and the {} encloses each object in the array.

An example is given below:

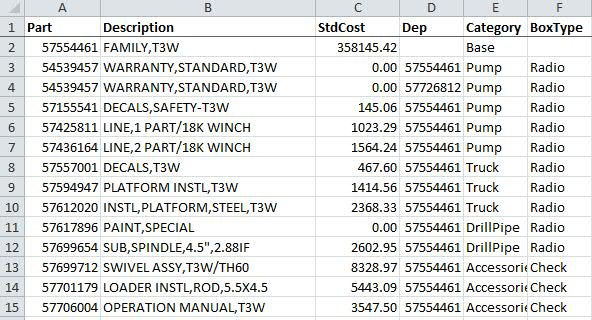
*var PRODUCTS =[{"Part":57554461,"Description":"FAMILY,T3W","StdCost":358145.42,"Dep":null,"Class":"Base","BoxType":""}, {"Part":54539457,"Description":"WARRANTY,STANDARD,T3W","StdCost":0.00,"Dep":57554461,"Class":"Pump","BoxType":"Radio"}]*

To alter the data you may use an excel file, where the columns represent the objects properties and the rows represent the objects. The following properties shall be included:

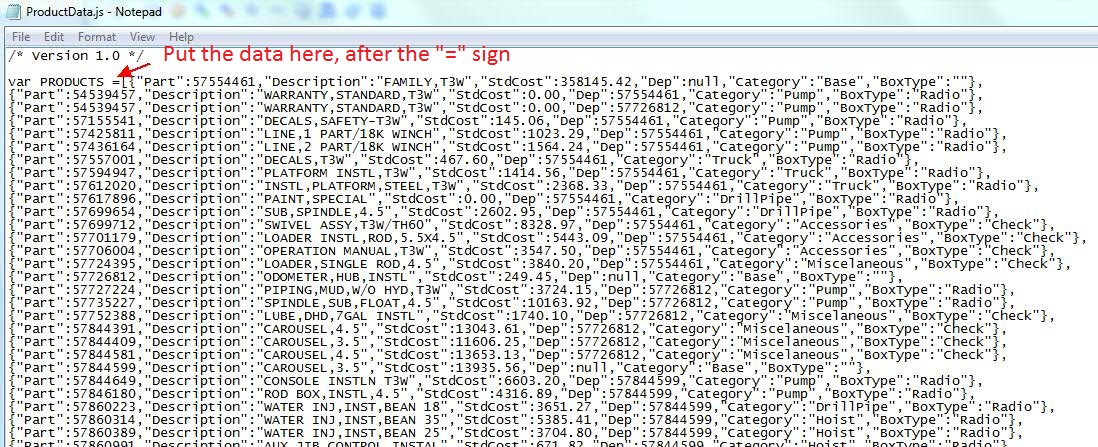
* **Part** – representing the unique part number
* **StdCost** – the cost of the product
* **Dep** – defines on which base machine the product is depending on. The unit will only show when the base it is depending on is selected. The value shall be the **Part** of the product it is depending on. If it is a base itself and thus not depending on any other unit the value shall be set to *null.*
* **Category** – Defines the category of the product, e.g. “Mud Pump” or “Accessories”. Product within the same category will appear together
* **BoxType** – defines if the selection shall be exclusive (radio box) or if multiple products within the same category may be selected.

**Note!** Make sure to remove any currency formatting from the standard cost in excel, e.g. 358,145.42 should only be written like 358145.42.

Example of excel data:



The excel file may then be converted to JSON format. This can be done e.g. by using an online tool like Mr. Data Converter: (<http://shancarter.github.io/mr-data-converter/>). Copy the excel data including the column names and paste it into the browser. Then copy the output and put it after the “var PRODUCTS =” in the ProductData.js.



## Cover Letters, Notes, and Terms and Conditions

### Standard letters

The standard cover letter are stored in ./data folder named CoverLetters.js. Format is an array with objects (the letters) where each object has a title and content. Example below:

*var LETTERS = [{title: 'Letter 1', content: 'Lorem ipsum dolor sit amet…’},{ title: 'Letter 2', content:’Sed ut perspiciatis…’}]*

The contents of the letters may use HTML markup. The title will be shown in the dropdown menu.

### Load/Saved Letters

Letter that are loaded and saved should be stored in a .txt file, preferable in the ./’Saved Letters’ folder. The content may use HTML markup.

### Notes

Notes may be saved and loaded the same way as the letter and use HTML markup.

### Terms and Conditions

Terms and conditions are stored in TermConditions.js and behave the same way as the Cover Letters.

## Product Descriptions and Technical Specifications

The product descriptions and technical specifications are stored as image files in the folders /data/desc and /data/spec with folder named the after the title e.g. the description of the TH60 will the stored in /data/desc/TH60 with file names TH60 1.png, TH60 2.png and so forth. Whenever a new specification or description is to be added a new folder has to be created.

The ProductDescription.js and TechnicalSpecification.js contain information about the descriptions and specifications and have to be modified whenever a change is being made. The data stored in the files contain information about the title, which has to be the same as the folder name, as well as the number of pages (images) that each folder contain. If there is only one folder with four images if would look like below:

*var DESCRIPTIONS = [{"Name": "TH60", "ImageCount":4}]*

**Note!** The images has to be named “title x” where x is the page number starting at 1 and be in .png format.