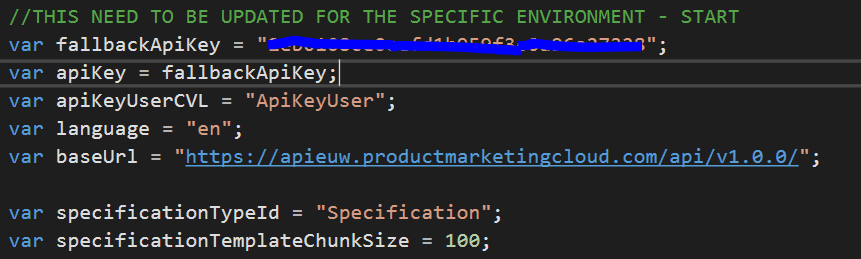
**Introduction**

This document covers the two (application) templates “Specification Mass Update” and “Specification Import/Export”. It is about the nitty gritty technical, behind the scenes stuff in the template code itself. How to set it up to fit you needs.

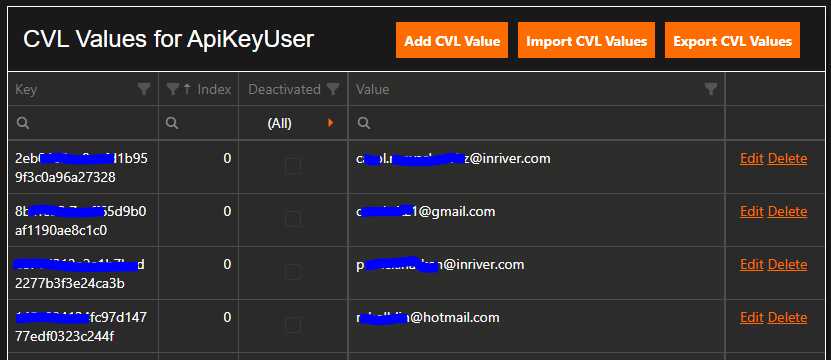
There is a section in the template where you can customize certain aspects of the template. Both templates have some common features and some features that is specific for that template.

**Common Features**

There is a section in the Html Templates that looks like this:

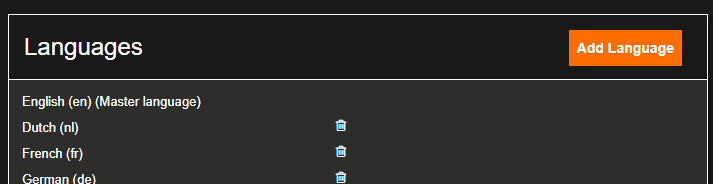


The Api Key is needed for accessing the Rest Api. There is a feature that simulates user context logins. It is based on a CVL which contain, key = Api Key and Value = Username. This CVL must be set up and maintained by either partner or customer. You must set the Id of the CVL by setting the ‘apiKeyUserCVL’ variable.



If you do not have set up the CVL or have an unmatched entry it will go to the ‘fallbackApiKey’ variable. This **must** be set with a valid REST Api Key.

The ‘language’ is setting the default language of the template. See Control Center for valid language codes.



The ‘specificationTemplateChunkSize‘-variable is setting the size of the chunks to fetch entities for a specific work area. So, if the work areas have 987 entities there will be 10 chunks of data fetched. 9 with a hundred entities each and on with 87 entities. The reason for this is:

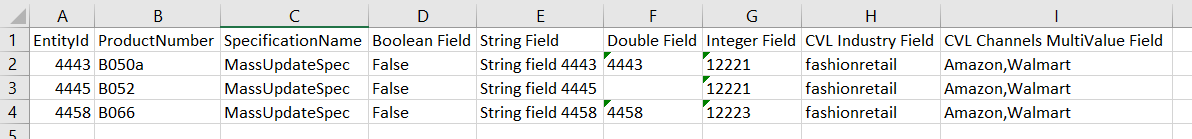
* Fetching a limited number of items takes shorter time than fetching the whole batch at once.
* If this is done in parallel, ten chunks with 100 entities each will take an overall time that is shorter than fetching all at once.
* The REST Api has a limitation of running 10 simultaneous requests. To limit the number of simultaneous requests is called throttling and is applied in the template.

**Template specific features - Import/Export**

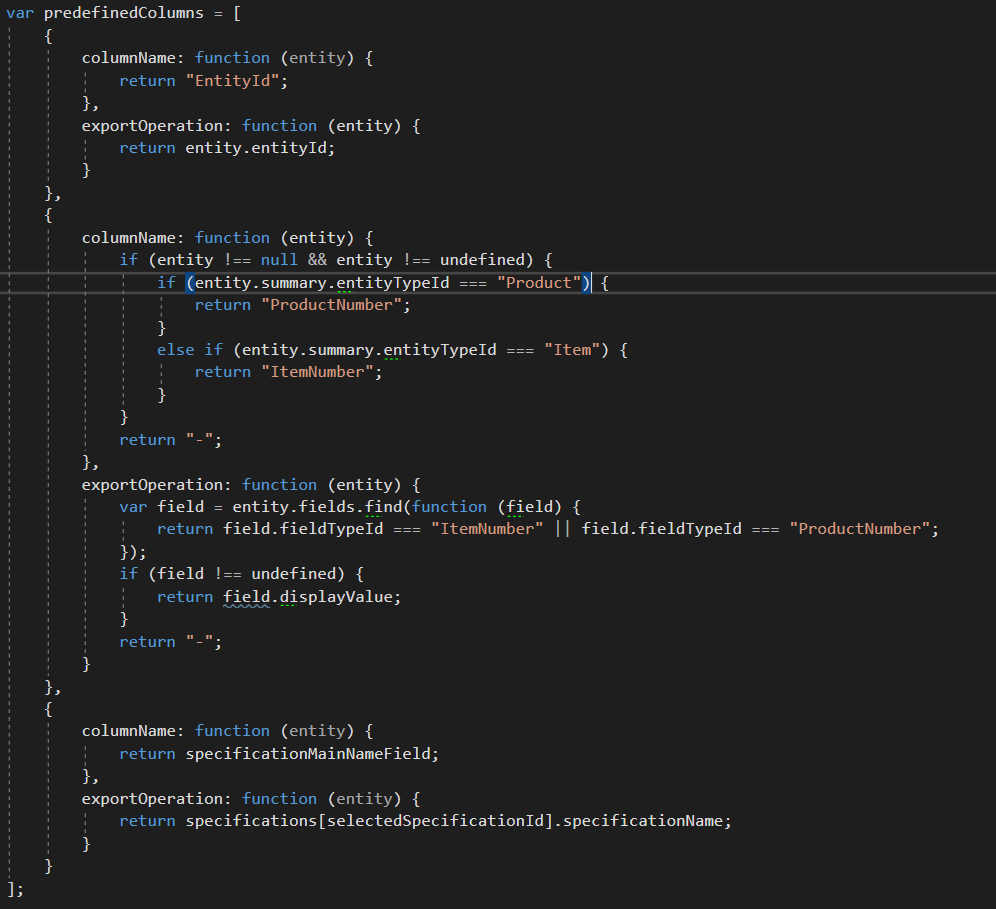
The Import/Export template has some additional features that **must** be set up. Before doing this, you must gather some information from the customer. The excel/csv file is divided into columns. Those columns are in turn divided into two sections:

1. Entity Id and Specification section
2. Fields section

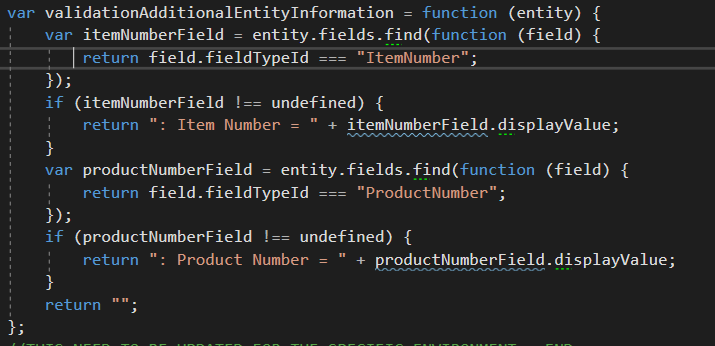
A **must** is always to have a column with the sys id of the entity and the specification name. The sys id of the entity **must** always be in the first column. As this is the base on which the template is resting on. In the below excel there is three columns, that are not Fields: EntityId, ProductNumber and SpecificationName:



These first three columns are specified with the following array of paired methods. Per array entry there is a method for the column name, the columnName method, and a method, the exportOperation method, for which value each row under the column name should have, based on the entity.



Another thing that is also dynamic is a method for making an validation error more readable. The method called in this case is the ‘validationAdditionalEntityInformation’-function.



What the method is doing is to generate this text (yellow marking):

