

AAPVL Tool Technical Documentation: Web Application

Release 0.1

Sebastian Sünkler

Aug 20, 2018

1. Used packages and libraries in the AAPVL Web application

Library	Further information
jQuery	https://jquery.com/
Datatables	https://datatables.net/
MySQL	https://www.mysql.com/
PDO PHP	http://php.net/manual/en/book.pdo.php
Bootstrap	https://getbootstrap.com/
phantomJS	http://phantomjs.org/
curl	https://curl.haxx.se/

2. Customization of the AAPVL Web Application

The AAPVL Web app is a highly customizable tool to require the needs of any classification and analysis task regarding to food control online. Their components are modular and could also be used to control other cases like counterfeit and piracy products. Such use cases need other documents as learning basis for the classifiers and there is also a need of customizing the datatables to meet such requirements. Most configuration will be customized by changing values in JSON-files located in the directory “/user”.

2.1. Customization of the Language

The user interface offers two standard templates for German and English-Language. But it is also possible to extend the languages by adding new files to the specific directories. All files accord to one language and are connected. The tool reads the labels and its values from the JSON files and saves them as \$_SESSION params. Is there a need to add some custom labels, so just insert such code snippet to the desired lines in the source code.

```
<?php echo $_SESSION["language_json"]["Label_XX"]; ?>
```

The main folder for the labels on the user interface is “/user/languages”. It has two files:

File	Explanation
user\language\languages_list.json	List with all supported languages: { "Language": ["Deutsch", "English"]

	}
user\language\Deutsch.json	Language file for German
user\language\English.json	Language file for English

2.1.1. Customize the language template

The template file for the naming of the labels is structured in blocks. Every block has a header which is written as a “comment”:

Start:

- **Language:** Name of the language. It must be identical to the name used for the language file and it is case sensitive.
- **Warning_NoScript:** Standard message if a user has disabled JavaScript in the Web Browser

Elements of the top navigation:

- **Label_Dashboard:** Label for the link text for the dashboard.
- **Label_Enterprise_Case:** Label for the link text for so called enterprise cases
- **Label_Mission_Case:** Label for the link text for so called mission cases
- **Label_Bio_Case:** Label for the link text for so called cases to check ecological certification fraud.
- **Label_Admin_Panel:** Label for the link text to the user administration.
- **Label_Logout:** Label for the link text to logout.

Alerts for the web form

- **Alert_Mandatory:** Custom message for mandatory form fields.
- **Alert_Date:** Alert message if some selects a start date for the query intervals which lies in the past.
- **Alert_Final_Date:** Alert message if the final date for query intervals is empty.

Names and labels for the cases

- **Header_Enterprise_Case:** Header for the index page of enterprise cases.
- **Header_Mission_Case:** Header for the index page of mission cases.
- **Header_Ecological_Case:** Header for the index page of ecological cases.

Column names for the cases

- **Column_Case_ID:** Column name for the index or name of a case (key).
- **Column_Date:** Column name for the creation date of the case
- **Column_Employee:** Name of Employee who is responsible for the case.
- **Column_Comment:** Column name for a comment
- **Column_Options:** Column name just for the options
- **Column_Case_Type:** Column name for the type of case
- **Column_Progress:** Column name to show the progress

Form buttons for the cases

- **Button_Add:** Label for the button to add a case
- **Button_New:** Label for the button to open a new case
- **Button_Save:** Label for the button to save a case
- **Button_Edit:** Label for the button to edit a case
- **Button_Save_Changes:** Label for the button to save changes
- **Button_Delete:** Label for the button to delete a case
- **Button_Assessment:** Label for the button to open the judgement interface
- **Add_Case:** Label for the button to add a case
- **Show_Case:** Label for the button to open a case
- **Alert_Delete:** Alert message to point out that a user is going to delete a case.

Column names for the search queries table

- **Header_Search_Queries:** Header for the table of search queries
- **Column_Query:** Column name for a search query
- **Column_Query_Start:** Label for the column to show the start date of a query scraping and the interval in days
- **Column_Query_End:** Label for the column to show the end date of a query scraping
- **Column_Progress:** Column name to show the scraping progresses

Column names for the urls table

- **Header_URL:** Header for the table of imported urls
- **Column_URLs:** Column label for the name of the url lists
- **Column_Start_URLs:** Label for the start date of url importing
- **Column_URLs_Days:** Label for interval in days to repeat the import of the urls
- **Column_URLs_Days_End:** Label for the end date of the interval processes

Search queries form

- **Textarea_Queries:** Textarea for the input of search queries
- **Queries_Start:** Start date for the web scraping
- **Queries_End:** End date for the web scraping
- **Queries_Case:** Header for the search queries form
- **Edit_Queries_Case:** Button label for the button to edit the queries
- **Button_Generate_Queries:** Button label for the button to generate some predefined search queries
- **Button_Extend_Queries:** Button label for the button to extend search queries by predefined keywords
- **Button_Add_Queries:** Button label for the button to add search queries

Urls form

- **ID_URLs:** Name for the url list
- **Textarea_URLs:** Header for the text area to insert urls of webpages
- **Case_URLs:** Label to show the case to the urls
- **Edit_Case_URLs:** Button label for the button to edit the urls list

- **Button_Add_URLs:** Button label for the button to add url lists

Footer text

- **Label_Project:** Footer text description to show the belonging to the research project AAPVL
- **Funded:** Reference text to demonstrate the funder of the project

Update Alerts

- **Save_Case:** Alert so show that a case has been saved
- **Save_Queries:** Alert so show that the search queries have been saved
- **Save_URLs:** Alert so show that the urls have been saved
- **Update_Case:** Update alert message on saved changes of a case
- **Update_Query:** Update alert message on saved changes of search queries
- **Update_URL:** Update alert message on saved changes of url lists
- **Delete_Case:** Delete message when a case has been deleted
- **Delete_Query:** Delete message when search queries have been deleted
- **Delete_URL:** Delete message when url list have been deleted
- **Missing_Queries_URLs:** Embedded message if a case has no search queries nor any url lists
- **Progress_URLs:** Embedded message on the progress of saved websites
- **Progress_Queries:** Embedded message on the progress of scraping search queries
- **Progress_Analysis:** Embedded message on the progress of analysis jobs
- **Manual_Judgement:** Header for the interface for manual judgement

Risk decision form fields

- **Risk Decision:** Form field to select the type of risk
- **Classification:** Label to classify the risk
- **Risk_Points:** Label for the text field to input own values for risk points

Labels for user administration

- **Header_Admin_Panel:** Header text for the index file of the admin panel
- **Button_Add_User:** Button label for the button to add a new user
- **Button_Edit_User:** Button label for the button to edit a user
- **Button_Delete_User:** Button label for the button to delete a user
- **Button_Reset_Password:** Button label for the button to reset a user password
- **Admin_User_ID:** Column label for the user id
- **Admin_User_Name:** Column label for the user name
- **Admin_User_First_Name:** Column label for the first name
- **Admin_User_Last_Name:** Column label for the last name
- **Admin_User_Mail:** Column label for the user mail
- **Admin_User_Symbol:** Column label for the user symbol
- **Admin_User_Options:** Column label for the user options
- **Admin_User_Pass:** Column label for the user password

- **Admin_Header_Pass:** Header for the page to reset a user password
- **Add_User:** Button label for the button to add a new user
- **Alert_Delete_User:** Button label for the button to delete a user

2.2. Customization of the JavaScript files for the jQuery datatables extension and Datepicker

The tool also supports custom languages for the extensions to render tables and for pick dates. The files are located in “/user/language/js”. The files are named like *{language_name}DataTables.js* and *{language_name}DatePicker.js*

2.3. Customization of the config files for the cases

The config of the cases is also affected by language settings. So, every case consists of some config and also of templates for queries for generating or extending search queries. All of these files can be found in “/user/cases”.

The file name for the config consists of the following parts:

{language_name}cases{cases_type}_config.json

The file name query extension consists of the following parts:

{language_name}cases{cases_type}_queries.json

Example file: Deutschcases01_config.json

```
{  "___comment___": "JSON file to choose the params for the case",

  "Config": [{ "ID": "1", "Name": "Lebensmittel", "Modules": [1,2,3], "Crawlers": [1],
  "Screenshots": 1}, { "ID": "2", "Name": "Futtermittel", "Modules": [1,2,3], "Crawlers": [1],
  "Screenshots": 1}, { "ID": "3", "Name": "Lebensmittel Kontaktmaterial", "Modules": [1,2,3],
  "Crawlers": [1], "Screenshots": 1}]

}
```

The file has values for these params:

- **ID:** id of the case (primary key for the database)
- **Name:** Name for the case in the preferred language
- **Modules:** Selection of Modules for the case (Module descriptions for the analysis application are described in “AAPVL_Analysis_App.pdf”).
- **Crawlers:** Selection of Crawlers for the case
- **Screenshots:** param to choose if the tool should take screenshots (actually pictures) or not (just saving the source codes of websites)

The file for the extension of search queries looks like:

```
{
  "__comment": "JSON file for the query extension",
  "Queries": [{"ID": "1",
    "Queries": ["Lebensmittel", "Nahrung", "Nahrungsergänzungsmittel", "Tiefkühlprodukte", "Frische
    produkte", "Backwaren", "Bier", "Brot", "Butter", "Eier", "Eis", "Fisch", "Fleisch", "Gemüse", "Getränk
    e", "Gewürze", "Honig", "Kaffee", "Käse", "Konfitüre", "Kuchen", "Marmelade", "Müsli", "Obst", "Pilze
    ", "Salat", "Schokolade", "Spirituosen", "Suppen", "Tee", "Teigwaren", "Torte", "Wein", "Wurst", "Zuc
    ker"], "Extension": ["liefern lassen", "lieferservice", "bestellen", "online", "online bestellen",
    "online kaufen", "kaufen", "onlinesupermarkt"]}, {"ID": "2", "Queries": ["", ""], "Extension": ["", ""]}
  ]
}
```

The file has values for these params:

- **ID:** This ID is the reference to the ID in the config file for the cases. So, it is possible to have different query lists for cases on food and cases on feed control.
- **Queries:** Just the list with the search terms to generate.
- **Extension:** The list with the keywords to extend the search queries.

2.4. Customization of web scrapers

The program has big advantages on tasks like gathering web search results from search engines using own search queries. It uses Curl and phantomJS to scrape results and to save local copies in text and as pictures of the websites. The main scraping process is realized by curl to save the search engine result pages (SERP) and by using XPath to extract the desired information from the pages. Scrapers can be customized and extended just by creating some specific JSON files. The files are located in "/user/scrapers".

The file *scrapers_list.json* has a similar task like the file to define the languages for the tool. It just consists of the available scrapers.

```
{
  "__comment": "JSON file for scrapers its important to call the scrapers like their
  filenames. Just add or delete scrapers here.",
  "Scrapers": ["GoogleDeutschland"]
}
```

It is important to name the config files for the scrapers similar to the name defined here. Like here it is *GoogleDeutschland.json*.

```
{
  "search engine": "GoogleDeutschland",
  "searchstring": "https://www.google.de/search?hl=de&site=&source=hp&q=[[term]]&lr=lang_d
  e&num=100", "xpath_results": "//h3[@class='r']/a", "next_serp":
  {"part_before": "https://www.google.de", "xpath": "//a[@id='pnnext']", "part_after": "&num=100"},
  "api-key": "", "max-pages": 10
}
```

The params for the file are:

- **search engine:** Name of the search engine to scrape
- **searchstring:** Search URL with GET-Parameters to generate the Urls for the scraping process ([[term]]) will be automatically replaced by search queries from the database)
- **xpath_results:** XPath phrase to find the URLs of the search results
- **next_serp:** That param is needed to get the dynamic links to the next SERPs. It consists of the param “part_before” for the first part of the URL, “xpath” to find the expression in the source code for the link to the next SERP and “part_after” for the end of the URL. It is important to know that “part_before” and “part_after” not requires. Every search url could look different so the source code with the SERP.
- **api-key:** Some search engines offer APIs. Such keys have to be stored here.
- **max-pages:** Definition of maximum SERPs to be scraped.

2.5. Customization of crawlers

The program offers the possibility to crawl some defined parts of websites for further analysis. Self-defined crawlers will be used to find links which include keywords. The crawler will search in the URL and the link text and use weights to determine which link fits best to the desired results.

The crawlers will be defined in “/users/crawler/” and the IDs of the crawler are used in the config files for the cases (see XX). The id is used for the file name too. So the crawler to find contact data subpages is called “1.json”.

2.6. Customization of the blacklist for urls

The file to change and add URLs to a blacklist is located in “/user/url_blacklist.json”. That file will be used to remove undesired Websites from search results and url lists.

2.7. Customization of the cronjobs

The tool has some tasks which have to be time controlled. The config file for the crons “/crons.txt” shows some examples of which tasks have to be repeat regularly.

2.8. Customization of the dynamic datatables for the manual judgement

The tool offers an interface to judge the results of all automatic analysis processes. We call that interface “Manual Judgement” and all of the processes there point to the cases. So, the results are datatables with some in grid editor to change the values. The configuration is similar to the previous ones and depend on the type of cases.

The following example demonstrates what is need to set up the manual judgement for a case.

2.8.1. Changing the language of the Columns for the manual judgement

The files to change the column names are in “/user/datatables”. Their name consists of the language name and type_cases. It is important to keep the number of the columns and just to change the names of them. Any change of the number will affect in an error.

2.8.2. Create a template for new cases

The access and the sql operation on the tables in the database are performed by PHP-PDO but the rendering happens by using jQuery Datatables. To realize all processes it is necessary to define a template in PHP and another in JavaScript.

The templates for the PHP server side processing are in “/server/” The name of a template looks like “server_processing_case01_template.php”.