stukowin

1.0

Generated by Doxygen 1.8.7

Thu Jul 31 2014 03:17:04

Contents

1 Module Documentation

1.1 Content

1.2 Introduction

Please note that this document was automatically generated without much control over the layout. It is only meant to be used in print or if it is necessary to have one single file. For everything else, please refer to the HTML documentation, which is easier to navigate, better structured and prettier!

1.2.1 Module

This documentation describes the custom module "stukowin" for Drupal. Drupal is a free and widely-used content management system (CMS), which provides the possibility to extend it by implementing proprietary modules that hook into the Drupal core system. The basic idea of this module is to connect the Drupal system with the curricula development and support system (CEUS) provided by the Johannes Kepler University in Linz, available at https://lss.jku.at/wiki/index.php/Main/CEUS and https://lss.jku.at/studienhandbuch/. The main goal of this method was to reduce the maintenance effort needed for keeping the curricula data on a website up to date.

The module is composed of four components, as seen below, each of which provides different functionalities to reduce the administrative effort and all of which are somehow interconnected.

1.2.2 Project

This module was created as a result of the course "IT Project", a part of the bachelor's degree in business informatics at JKU, during the summer semester 2014.

The project name is "Relaunch der Homepage der Studienkommission WIN". The original site can be found at http://stukowin.jku.at/. The project sponsor is Dr. Stefan Berger (stefan.berger@jku.at).

The main functional project requirements include the following:

1. Automatic import of curricula data from CEUS
2. Possibility of editing curricula entries after the import
3. Version management for curricula entries
4. Automatically displaying different curricula on the site

5. Possibility to create new ideal courses of studies ("Idealtypischer Studienverlauf", ITSV) and specialisation curricula

6. Archiving and exporting curricula in the form of PDF documents

Remarks

This documentation is one of four documents that make up the entire project documentation, namely the operator guide, the user guide and the test documentation.

1.2.3 Language

As Drupal is an open source system that is mostly documented in English, with most of the modules available for it also documented in English, we decided to document our module accordingly. It is for this reason that the language of this documentation differs from the other parts of the documentation mentioned above.

1.3 CEUS API

In order to make functional requirements such as displaying curricula and course details possible, it is necessary to extract the required data from CEUS. For this reason, JKU offers a JSON API for CEUS at https://lss.jku.at/studienhandbuch/api/0.1 (Note that this URL is not available without authentication credentials). This API was developed for this project (hence the version number of 0.1) and is still prone to changes. Due to its novelty, no documentation for the API exists, which is why we try our best to document it together with our own module documentation.

Note

This is not the official documentation of the CEUS API, it is merely based on our own knowledge about it.
For any questions concerning the API please contact Andreas Roesch (Andreas.Roesch@jku.at) or Mag. Martin Stabauer (martin.stabauer@jku.at)

This API provides four public methods which will be described below.

1.3.1 auth.php

Method	/auth.php
Input	"username","password"
Output	"authtoken"

The method `auth` serves the purpose of authenticating a user and returns a token which is required for every other API method call. The user credentials at the time of development are as follows:

```
User: dke2
Password: studke2hb
```

1.3.2 list.php

Method	/list.php
Input	"authtoken"
Output	Array of Branch objects (see below) in JSON

The method `list` returns an array of all curricula to which the user has access. In the scope of this project, these are the bachelor and master studies in business informatics. The curricula are returned in the data structure shown below:

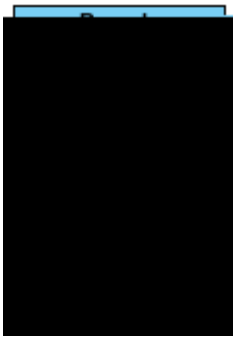


Figure 1: Branch Objects

The data structure contains a few elements. For this project, only the following are relevant:

Field	Description
id	CEUS id of the curriculum. This id is needed for the curr.php method.
name	German name of the curriculum
namme_en	English name of the curriculum
faculty	Faculty of the curriculum
type	Bachelor or master studies
typeshort	Short version of the type (B or M)
version	Semester when the curriculum was introduced (e.g. 2013W)

1.3.3 curr.php

Method	/curr.php
Input	"authtoken", "id"
Output	Object of type curriculum (see below) in JSON

The method `curr` returns general information about a curriculum and, more importantly, its structure. A curriculum is identified throug the id returned by list.php method. Curricula are returned in the following structure:

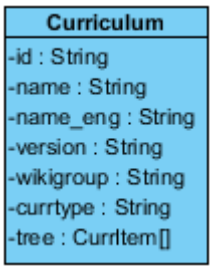


Figure 2: Curriculum Objects

All but the last element have already been described in list.php The last one is described below:

Field	Description
tree	A nested array of `CurrItems`, which represents the curriculum structure. The field `id` of each `CurrItem` is needed for the detail.php method. The field `subtree` contains the subcourses of each course

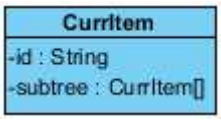


Figure 3: CurrItem Objects

1.3.4 detail.php

Method	/detail.php
Input	"authtoken", "id" [,"lang"]
Output	Array of Detail objects (see below) in JSON

The method `detail` returns the details for one `CurrItem`. A `CurrItem` can be a subject (Fach), a module (Modul) or a simple course (Lehrveranstaltung). For localised data the optional parameter `lang` with the value "de" (German) or "en" (English) can be given. Details are returned in the folling data structure:

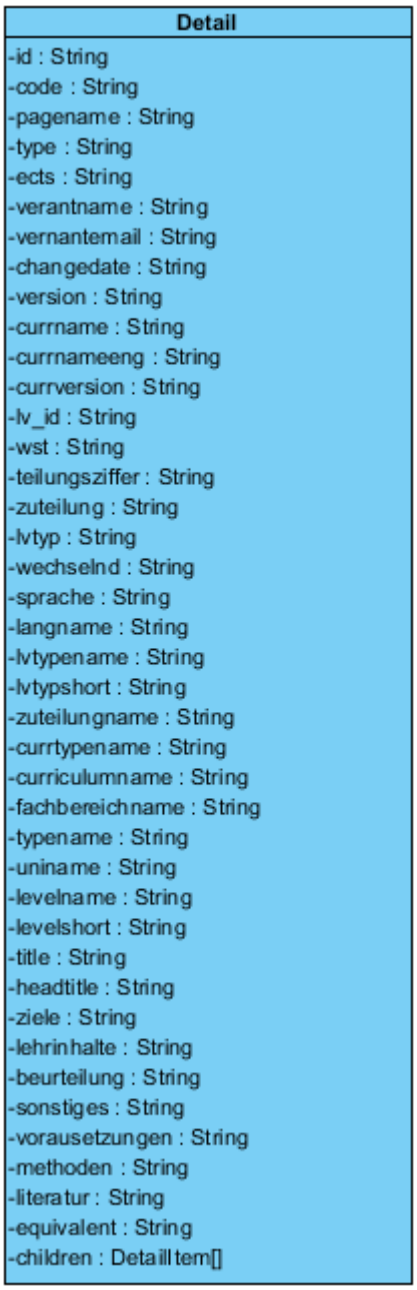


Figure 4: Detail Objects

For this project, mainly the following elements are relevant:

Field	Description
ects	Credits for the course
verantname	Name of the person responsible for this course
verantemail	Email of the person responsible for this course
changedate	Date and time of the last change of this entry (in the format "YYYY-MM-DD hh:mm:ss")
currname	Name of the curriculum that contains this course
wst	Semester periods per week (Semesterwochenstunden) for this course

langname	Language the course is held in
lvtypshort	Short type of the course (e.g. VL, UE, KV etc.)
lvatype	Long type of the course (Vorlesung, Uebung etc.)
typename	Long type of the entry (subject, module or course)
type	Number representative of the type of entry (1=subject, 2=module and 3=course)
title	Course title
ziele	Course goals
lehrinhalte	Contents of teaching
voraussetzungen	Course requirements
methoden	Teaching methods

1.3.5 Errors

In case of an error, the CEUS API returns a JSON object with the error message contained in the `error` field. Known errors are:

- Invalid id given
- Missing parameter
- Missing authentication token

1.4 Modules

1.4.1 CEUS2Drupal

1.4.1.1 Import

The import of data from CEUS to the Drupal environment is implemented in this component. All its functionality is collected in the CEUS2Drupal component. The data model for the Drupal-internal representation of courses is defined in the module's install component, the necessary database tables are automatically created during the installation. Because the API URL and user credentials are prone to change, they need to be configurable. This is achieved in the admin menu:

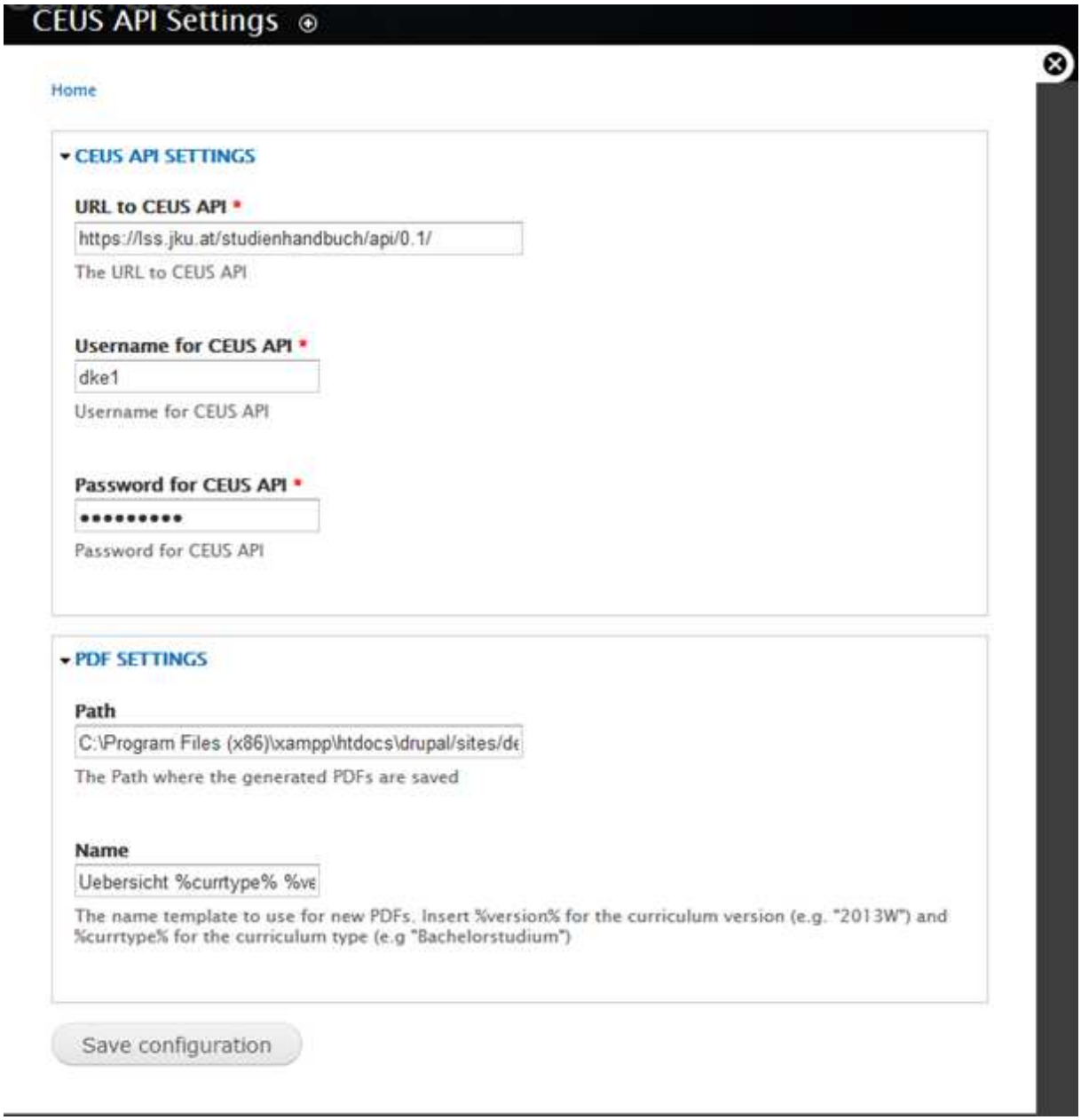


Figure 5: Module Configuration Page

Note

Administrator rights are needed to configure the login details

The communication between this module and CEUS occurs as shown in the included SequenceDiagramImport.pdf file.

The import from CEUS is performed roughly in the following steps:

1. The `auth` token is requested from the CEUS API
2. All available curricula are received from the API
3. For each curriculum, the curriculum tree is requested from the API
4. For each course in the tree, the details are requested in German and English

5. During the first import, new Drupal vocabularies, vocabulary terms and content nodes are created

6. We try our best to parse the field `voraussetzungen` and extract relations between courses (recommended/required)

This process is shown in the included FlowchartCEUS2Drupal.pdf file, with the corresponding methods in ceus_importer that perform each step. Here is a quick textual description of the process:

The data import is initiated in the file stukowin.module. This file first access ceus_importer::connect() in the file ceus_importer.inc.php and prepares the import process. In order for this to work, the necessary settings have to be made in the configuration menu shown above. All methods for the data import are collected in the ceus_importer.inc.php file.

After this, the method ceus_importer::get_curricula() is called, which starts the import process. It checks if a Drupal vocabulary for the imported curricula already exists. If none are found, a new one is created. If the curricula's version numbers are different, a new vocabulary is created as well.

After getting the tree for each curriculum, the details for its courses are requested and stored.

Another noteworthy part is the method ceus_importer::process_relations(). This method is responsible for parsing the field `voraussetzungen`, which can contain any text possible. The method examines the field in three steps:

1. Is the field empty? If not, go to Step 2.

2. Does the field begin with "empfohlen"? If yes, it is a recommended relation, if not it is a required relation.

3. Try to find <li> and <a> tags and extract a course title or code from them.

Requirements are important mostly because they are used and shown in the automatically generated graphic representation.

All of the code for the import is contained in the ceus_importer.inc.php file.

See also

CEUS2Drupal

1.4.1.2 Change Management

The functional requirements of this project make it necessary to periodically extract data from CEUS on the one hand and giving administrators and moderators the ability to overwrite data on the other hand.

Remarks

How to overwrite course data is described in the user documentation.

This creates the challenge of properly versioning the CMS content, which is why every import follows the rules shown in the included Changemanagement.pdf file

Every time a content node is created or updated, it gets a red New tag in the content overview (see image below), through which the administrator can easily see which nodes have been updated. In addition to this, the import returns a success message that tells the administrator how many nodes have been created or updated, so that one can easily tell if changes have occurred.

Figure 6: Red New Tag

1.4.2 Drupal2ITSV

As CEUS does not provide any information about fields of specialisation during the master studies and ideal courses of studies, henceforth called ITSV due to its German name, it was a project requirement that new curricula can be created by the administrator for such purposes.

A freely available Drupal module called Taxonomy Manager (https://www.drupal.org/project/taxonomy_manager) gives the administrator the ability to copy vocabulary terms from one vocabulary to another, which is most of the work. Unfortunately, this process cannot be simplified any further. Nevertheless, we tried to at least automate the task of creating a new vocabulary, copying all of the information over from the source curriculum and creating top-level terms (such as "1. Semester" etc.), tasks which will be performed every time a new ITSV or specialisation has to be created.

This component handles exactly that. It inserts a new menu item (at admin/settings/stukowin/taxonomy) where the administrator can select a source curriculum to base the new one on, select whether to create an ITSV or specialisation vocabulary, enter a name and choose how many top-level terms should be inserted. Once the administrator has filled out the form, the new vocabulary is automatically created and the browser is redirected to the Taxonomy Manager's "Dual View", where the administrator can begin copying courses into the new vocabulary.

Remarks

This component does not have its own file as it does not contain a lot of code. All of its functionality is in the stukowin.module file.

See also

Drupal2ITSV

1.4.3 Drupal2PDF

In order to permanently and easily archive past curricula there is the option to automatically create a course overview PDF document from the imported courses. For this component to work properly, the necessary settings have to be made in advance (see module configuration image above). The administrator is provided with a new menu (at admin/settings/stukowin/pdf) where the curriculum to be archived can be selected. Once he clicks on "Create PDF", the PDF generation in overviewPDF::createPDF() is started. (All of the PDF generation code is contained in the pdf_creator.inc.php file)

The PDF generation process will flow as shown in the included FlowchartDrupal2PDF.pdf file.

See also

Drupal2PDF

1.4.4 Drupal2AGG

So far we have described how to import and edit curricula into Drupal. But part of the functional requirements were also being able to automatically display the curricula on the website in the form of an automatically generated graphic (AGG). This component provides the functionality for doing exactly that. Again, it can be separated into three components:

1.4.4.1 Drupal JSON Interface

To make the curricula data available to clients, several JSON interfaces have been implemented that make curricula publicly reachable. This is needed so that the client browser can properly display the curricula.

The following interfaces are available:

Name	Path	Input Parameter	Description
Curriculum List	stukowin/crclmlst	"currtype" : Type of curriculum to get ("Bachelorstudium" or "Masterstudium") "taxtypes" : Types of vocabularies to get ("curriculum", "itsv" and/or "specialisation")	Returns a list of all curricula currently published (weight below 0), like list.php (see first image below)
Curriculum Tree	stukowin/crclm	Vocabulary id of the curriculum to get	Returns the curriculum tree of one curriculum, containing all courses and their details (see second image below)
Single course	stukowin/lva	Node id of the course to get	Returns the details of a single course (see third image below)

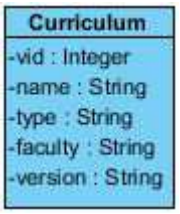


Figure 7: Curriculum Objects

Example:


```
[
    {
        "vid": "2",
        "name": "Wirtschaftsinformatik, Bachelorstudium 2013W",
        "type": "Bachelorstudium",
        "faculty": "Sozial- und Wirtschaftswissenschaftliche Fakultät",
        "version": "2013W"
    },
    {
        "vid": "3",
        "name": "Wirtschaftsinformatik, Bachelorstudium 2014W",
        "type": "Bachelorstudium",
        "faculty": "Sozial- und Wirtschaftswissenschaftliche Fakultät",
        "version": "2014W"
    },
    {
```

```
        "vid": "4",
        "name": "Wirtschaftsinformatik, Masterstudium 2013W",
        "type": "Masterstudium",
        "faculty": "Sozial- und Wirtschaftswissenschaftliche Fakultät",
        "version": "2013W"
    },
    {
        "vid": "5",
        "name": "Wirtschaftsinformatik, Masterstudium 2014W",
        "type": "Masterstudium",
        "faculty": "Sozial- und Wirtschaftswissenschaftliche Fakultät",
        "version": "2014W"
    }

]
```

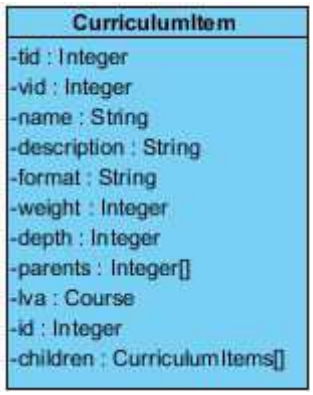



Figure 8: Items in a Curriculum Tree

Example:


```
[
    {
        "tid": "1",
        "vid": "2",
        "name": "Studienfach: Grundlagen der Wirtschaftsinformatik",
        "description": "1",
        "format": null,
        "weight": "0",
        "depth": 0,
        "parents": [
            "0"
        ],
        "lva": {
            "title": "Grundlagen der Wirtschaftsinformatik",
            "ceusid": "2895",
            "code": "526GLWN11",
            "ects": "24",
            "verantname": "Friedrich Roithmayr",
            "verantemail": "friedrich.roithmayr@jku.at",
            "changedate": "2012-02-14 00:00:00",
            "lvatype": "1",
            "typename": "Studienfach",
            "ziele": "Die Studierenden beherrschen die Fachterminologie, besitzen einen Überblick über
      Aufgabenstellungen, Konzepte, Methoden und Werkzeuge der Wirtschaftsinformatik. Sie sind mit den Grundlagen der
      Datenmodellierung, der Prozess- und Kommunikatonsmodellierung und der Algorithmik vertraut. Die Studierenden
      verstehen die daten- und verhaltensorientierten Aspekte betrieblicher Informationssysteme.\n",
            "lehrinhalte": "Das Fach dient der Vermittlung von Grundlagenwissen in verschiedenen
      Teilbereichen der Wirtschaftsinformatik. Details der Lehrinhalte können den zugehörigen Modulen entnommen werden.\n",
            "id": "1"
        },
        "id": "1",
        "children": {
            "2": {
                "tid": "2",
                "vid": "2",
                "name": "Modul: Einführung in die Wirtschaftsinformatik",
                "description": "2",
                "format": null,
                "weight": "0",
                "depth": 1,
                "parents": [
```

```
                "1"
            ],
            "lva": {
                "title": "Einführung in die Wirtschaftsinformatik",
                "ceusid": "7209",
                "code": "526GLWNEWI13",
                "ects": "6",
                "verantname": "Friedrich Roithmayr",
                "verantemail": "friedrich.roithmayr@jku.at",
                "changedate": "2013-08-23 00:00:00",
                "lvatype": "2",
                "typename": "Modul",
                "ziele": "Die Studierenden können begründen, dass Wirtschaftsinformatik eine
    Wissenschaft ist. Sie kennen die Aufgaben der Wirtschaftsinformatik und die Bedeutung der Fachsprache für die
    Wirtschaftsinformatik. Sie kennen die grundlegenden Phänomene, mit denen sich Wirtschaftsinformatik beschäftigt. Sie
    können den Gegenstandsbereich der Wirtschaftsinformatik systemtheoretisch erklären und erkennen, dass
    Wirtschaftsinformatik eine Interdisziplin ist. Sie kennen die Bezeichnung und Bedeutung von Instrumenten zur
    Konstruktion, Implementierung und zum Management von Informationssystemen und IT-Infrastrukturen. Ziele,
    Methoden, Modelle und Werkzeuge dazu können sie anhand von Beispielen erklären.\n",
                "lehrinhalte": "Gegenstandsbereich der Wirtschaftsinformatik, Berufsbilder in der
    Wirtschaftsinformatik, Studiengang und Studienpläne der Wirtschaftsinformatik, Lehrveranstaltungsformen und
    Lehrmethoden der Wirtschaftsinformatik, Geschichte der Wirtschaftsinformatik, Wissenschaftscharakter der
    Wirtschaftsinformatik, Begriffssystem der Wirtschaftsinformatik, Forschungsziele und Forschungsmethoden der
    Wirtschaftsinformatik, Nachbardisziplinen der Wirtschaftsinformatik, Praxisorientierung der Wirtschaftsinformatik,
    Information und Kommunikation, Informationsfunktion und -bedarf, Informationsverhalten und
    Informationsbedürfnis, Informations- und Kommunikationstechnik, Informationssystem und -infrastruktur, Benutzer und
    Benutzersystem, Ziele und Zielsystem, Methodik, Ansätze und Strategien, Grundsätze und Prinzipien, Modelle und
    Konzepte, Methoden und Werkzeuge, Evaluation und Bewertung.\n",
                "empfehlung": [
                    "56",
                    "63",
                    "75"
                ],
                "id": "2"
            },
            "id": "2",
        }
      }
    }
]
```


Figure 9: Course Objects

Example:


```
{

    "title": "Prozess- und Kommunikationsmodellierung",
    "ceusid": "6841",
    "code": "1WBWPKU",
    "ects": "3",
    "wst": "2",
    "verantname": "Christian Stary",
    "verantemail": "christian.stary@jku.at",
    "changedate": "2013-07-09 00:00:00",
    "lvtypname": "Übung",
    "lvtypshort": "UE",
    "lvatype": "3",
    "typename": "Lehrveranstaltung",
    "ziele": "Siehe <a class='wikilink'
       href='https://lss.jku.at/wiki/index.php/CurriculumWirtschaftsinformatikB02008W/ModulProzess-UndKommunikationsmodellieru
    "lehrinhalte": "Siehe <a class='wikilink'
       href='https://lss.jku.at/wiki/index.php/CurriculumWirtschaftsinformatikB02008W/ModulProzess-UndKommunikationsmodellieru
```

```
    ",
  "id": "13"
}
```

All of these interfaces are defined in the stukowin.module file.

1.4.4.2 CKEditor Plug-in

As the dynamic display of curricula in the browser requires a strict HTML document structure, we refrained from letting the administrator insert the code himself wherever a curriculum display was needed and instead implemented a plug-in for the CKEditor (https://www.drupal.org/project/ckeditor) which inserts the code automatically on the click of a button. The plug-in is registered with Drupal in the stukowin_ckeditor_plugin() method. It is then defined in the plugin.js file and the dialogue for inserting a curriculum view is defined in the stukowin_curriculum.js file.

Note

In order for this plug-in to work properly, some settings have to be made in the CKEditor. This is described in the operator manual.

1.4.4.3 Graph.js

The main task of displaying the curricula in the client's browser is performed by this file. It fetches the data from the JSON interfaces described above and creates the DOM elements in the browser. It also registers click handlers for things such as expanding/reducing a course, showing prerequisites and selecting a different curriculum.

Note

The display layout is configured in the curriculum_stlye.css file (not included in this documentation)

The process of generating the graphical representation will flow as shown in the included FlowchartDrupal2AGG.pdf file.

As an example, the output could look like this:

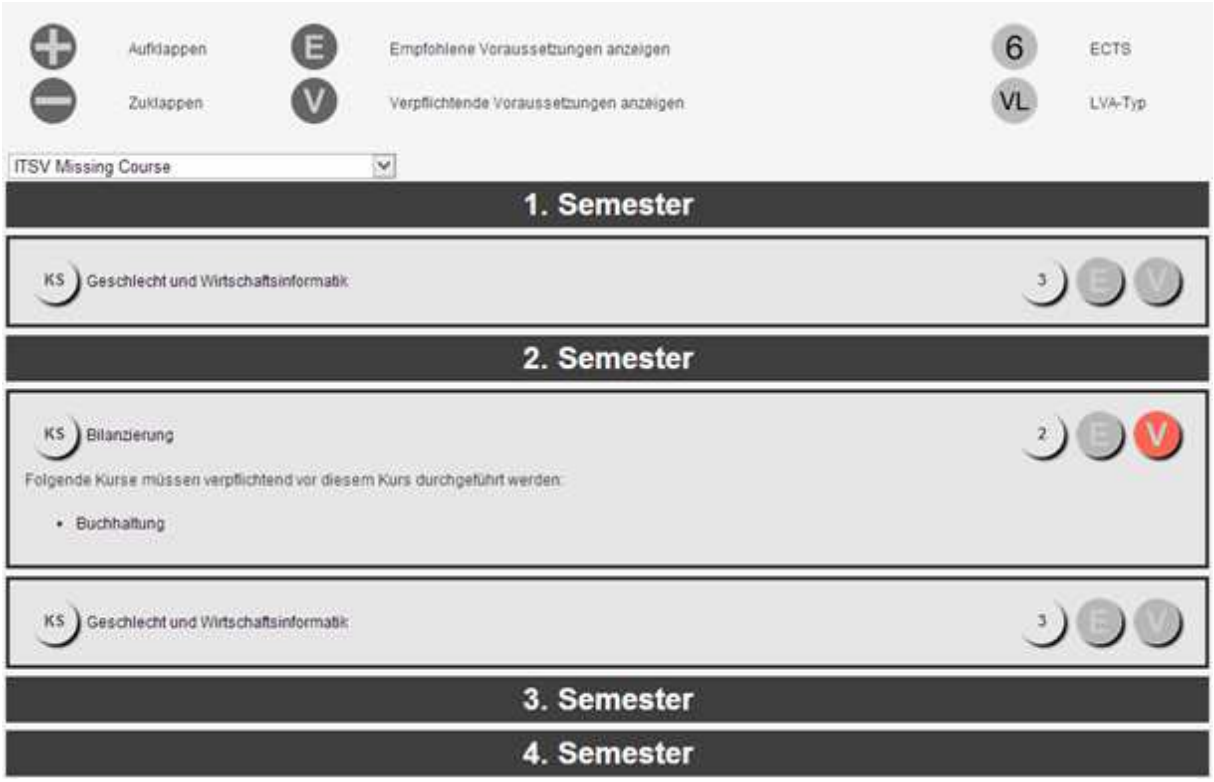


Figure 10: Graphical Representation

See also

Drupal2AGG

1.4.5 Module Integration into Drupal

All of the components described above need to work together somehow and at the same time be accessible to Drupal itself. The interaction with Drupal is carried out through so-called "hooks", which are functions that are known to Drupal and called when certain events occur. To quote the Drupal documentation:

> Drupal's module system is based on the concept of "hooks". A hook is a PHP function that is named foo_bar(), where "foo" is the name of the module (whose filename is thus foo.module) and "bar" is the name of the hook. Each hook has a defined set of parameters and a specified result type.
>
> To extend Drupal, a module need simply implement a hook. When Drupal wishes to allow intervention from modules, it determines which modules implement a hook and calls that hook in all enabled modules that implement it.

Thus, all our hooks are prefixed with "stukowin_" and are commented with "Implements hook_*". The following hooks are implemented:

Hook	Implementing method	Description
hook_install	stukowin_install()	Method that is run on the first activation of the module (=installation)

hook_uninstall	stukowin_uninstall()	Method that is run to uninstall the module
hook_help	stukowin_help()	Provides information about the module
hook_menu	stukowin_menu()	Registers menu links/buttons etc. with Drupal
hook_theme_registry_alter	stukowin_theme_registry_alter()	Registers our custom view template for displaying courses
hook_ckeditor_plugin	stukowin_ckeditor_plugin()	Registers our own CKEditor Plugin with CKEditor/Drupal

These hooks and other methods and files to ensure a smooth cooperation are spread across the following files:

- The content_manager class is responsible for accessing the imported curricula data
- The stukowin.install file is responsible for installing and uninstalling the module and all the data structures that come with it
- The stukowin.module file contains all the core hooks except for the installation routines
- The stukowin.info file contains all meta information about the module

1.5 Development

1.5.1 Authors

Note

Every file, class, method and member is documented with an `@author` tag. The person tagged as `@author` (thus being shown as the author) is the main/initial author of that file, class, method or member.
All other authors, (people that have helped during initial development, fixed bugs or made changes) are tagged with the `@author`**s** (plural) tag, thus being shown under Authors (plural).
If a file, class, method or member has been authored by multiple people equally, all of them are tagged with the `@author`**s** tag and there is no `@author`.
If you have any questions regarding the system, feel free to contact the respective author.

The following list shows an overview of every person that has participated in writing this module, with a quick summary of what they have mostly contributed to:

- Jakob Strasser - jakob.strasser@telenet.be
  Main author of the PDF generation component and the graph.js file. Participating author in most files/components. Functional testing.
- Konstantinos Dafalias - kdafalias@gmail.com
  Mostly responsible for the CEUS import and the Module Core
- Werner Breuer - bluescreenwerner@gmail.com
  Co-author of the CKEditor Plug-in and the Drupal2AGG component in general, as well as the Drupal2ITSV component. Participated in a lot of debugging.
- Markus Gutmayer - m.gutmayer@gmail.com
  Co-author of the CKEditor Plug-in and the Drupal2AGG component in general, as well as the Drupal2ITSV component. Participated in a lot of debugging.
- Fabian Puehringer - f.puehringer@24speed.at
  Assistance with Drupal2PDF and Drupal2AGG & Testing
- Manuel Muehlburger - Hansbert92@googlemail.com
  Layout and design for the Drupal2AGG component (mainly the Graph.js) & Testing

1.5.2 Version Numbers

Every file and class has a version number, composed of four parts: major.minor.revision date

- Major signifies a major release, such as the initial release
- Minor signifies the addition of a new feature
- Revision signifies a bug fix or something similarly small
- Date shows when the last change was made to the source code (in the format "YYYY-MM-DD")

1.5.3 Version Control

- When the project started, there were two groups, each of which used a different version control system.
- Group 1 used SVN on `http://cloudforge.com` while group 2 used `http://bitbucket.org`.
- When the two groups were merged, the SVN repository was dropped.
- Due to privilege problems with bitbucket, it was decided to move the entire repository to `http://github.com`, as everyone had access there.
- At the time of writing this documentation, the repository is available at `http://github.com/TheJake123/DrupalModul`.

As the repository is open source, the commit, issue and change history can be publicly viewed. Additionally, every file, class, method and member has been documented with a `@since` tag with the following format: "Commit <hash> on YYYY-MM-DD", so that it is easily visible when a file, class, method or member has been added.

1.6 Included Libraries/Files

For the PDF generation, the free and open source library `TCPDF` is used and included in the project.

For parsing course relationships, the `simple_html_dom.php` file is used

2 Module Index

2.1 Modules

Here is a list of all modules:

3 Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

4 Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

5 File Index

5.1 File List

Here is a list of all files with brief descriptions:

6 Module Documentation

6.1 CEUS2Drupal

Module for importing data from CEUS.

Files

- file ceus_importer.inc.php
  Imports data from CEUS.

Data Structures

- class ceus_importer
  Imports data from CEUS and stores it in the Drupal database.

Functions

- stukowin_pre_retreive ($form, &$form_state)
  Menu callback for import ("admin/settings/stukowin/import")
- stukowin_pre_retreive_submit ($form, &$form_state)
  Submit handler for stukowin_pre_retreive()

6.1.1 Detailed Description

Module for importing data from CEUS.

This module is reponsible for requesting the data from the CEUS API and storing it in the drupal database. It also implements the functions for the change management.

For a more detailed description see CEUS2Drupal

Author

Konstantinos Dafalias - kdafalias@gmail.com

Authors

Jakob Strasser - jakob.strasser@telenet.be
Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

6.1.2 Function Documentation

6.1.2.1 stukowin_pre_retreive ($form, & $form_state)

Menu callback for import ("admin/settings/stukowin/import")

Used as a warning before starting the import, as the process usually takes a long time to complete.

Remarks

There is nothing to do in this menu except click "Ok".

Parameters

array	$form	Form structure as given by Drupal
array	$form_state	Form state as given by Drupal

Returns

The filled out form structure

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit bcdb4ba on 2014-06-30

See also

stukowin_re_retrieve_submit()

Definition at line 248 of file stukowin.module.

6.1.2.2 stukowin_pre_retreive_submit ($form, & $form_state)

Submit handler for stukowin_pre_retreive()

Starts the import process and displays the outcome to the user.

Parameters

array	$form	Form structure as given by Drupal
array	$form_state	Form state as given by Drupal

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit bcdb4ba on 2014-06-30

See also

stukowin_pre_retreive()

Definition at line 275 of file stukowin.module.

6.2 Drupal2AGG

Module for transforming CEUS data to a proper graphical representation.

Files

- file graph.js

  Script for nicely displaying CEUS data.
- file stukowin_curriculum.js

  The stukowin_curriculum dialog definition.
- file plugin.js

  Registers the CKEditor plugin.

Functions

- stukowin_get_crclm_taxonomy ($iVID)

  JSON service for the nested array of courses details for one curriculum.
- stukowin_get_crclm_list ()

  JSON service for a list of all currently published curricula.
- stukowin_get_lva ($iNodeID)

  JSON service for the details of a single course.
- stukowin_ckeditor_plugin ()

  Implements hook_ckeditor_plugin.
- stukowin_theme_registry_alter (&$theme_registry)

  Implements hook_theme_registry_alter.

6.2.1 Detailed Description

Module for transforming CEUS data to a proper graphical representation.

This module contains all files, classes and methods that provide the functionality for automatically generating a visual representation of the imported curricula data.

6.2.2 Drupal JSON Interface

To make the curricula data available to clients, several JSON interfaces have been implemented that make curricula publicly reachable. This is needed so that the client browser can properly display the curricula.

The following interfaces are available:

Name	Path	Input Parameter	Description
Curriculum List	stukowin/crclmlst	"currtype" : Type of curriculum to get ("Bachelorstudium" or "Masterstudium") "taxtypes" : Types of vocabularies to get ("curriculum", "itsv" and/or "specialisation")	Returns a list of all curricula currently published (weight < 0), like list.php

Curriculum Tree	stukowin/crclm	Vocabulary id of the curriculum to get	Returns the curriculum tree of one curriculum, containing all courses and their details
Single course	stukowin/lva	Node id of the course to get	Returns the details of a single course

All of these interfaces are defined in the stukowin.module file.

6.2.3 CKEditor Plug-in

As the dynamic display of curricula in the browser requires a strict HTML document structure, we refrained from letting the administrator insert the code himself wherever a curriculum display was needed and instead implemented a plug-in for the CKEditor (https://www.drupal.org/project/ckeditor) which inserts the code automatically on the click of a button. The plug-in is registered with drupal in the stukowin_ckeditor_plugin() method. It is then defined in the plugin.js file and the dialog for inserting a curriculum view is defined in the stukowin_curriculum.js file.

Note

In order for this plug-in to work properly, some settings have to be made in the CKEditor. This is described in the operator manual.

6.2.4 Graph.js

The main task of displaying the curricula in the client's browser is performed by this file. It fetches the data from the JSON interfaces described above and creates the DOM elements in the browser. It also registers click handlers for things such as expanding/reducing a course, showing prerequisites and selecting a different curriculum.

Note

The display layout is configured in the curriculum_stlye.css file (not included in this documentation)

As an example, the output could look like this:

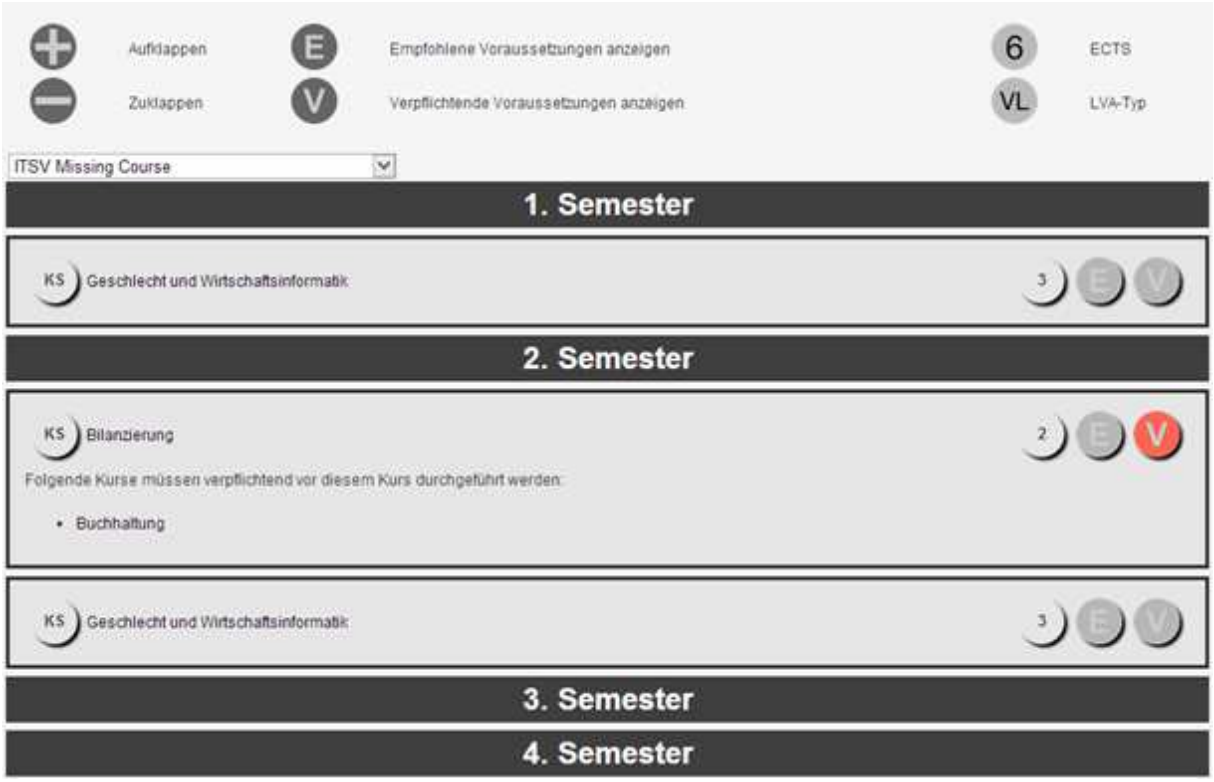


Figure 11: Graphical Representation

Authors

Jakob Strasser - jakob.strasser@telenet.be
Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com
Manuel Muehlburger - Hansbert92@googlemail.com

6.2.5 Function Documentation

6.2.5.1 stukowin_ckeditor_plugin ( )

Implements hook_ckeditor_plugin.

Adds the hook for the ckeditor plugin to insert a curriculum display.

Authors

Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com

Version

1.0.0 2014-07-16

Since

Commit b63de89 on 2014-07-01

Definition at line 599 of file stukowin.module.

6.2.5.2 stukowin_get_crclm_list ( )

JSON service for a list of all currently published curricula.

The type of curriculum (e.g. "Bachelorstudium", "Masterstudium"), types of taxonomies (e.g. "curriculum", "itsv" or "specialisation") and the language (e.g. "de") can be given as HTTP GET parameters through the respective parameter names `"currtype"`, `taxtypes` and `lang`.

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

content_manager::getCurricula()

Definition at line 317 of file stukowin.module.

6.2.5.3 stukowin_get_crclm_taxonomy ($iVID)

JSON service for the nested array of courses details for one curriculum.

Parameters

integer	$iVID	The Drupal vocabulary id of the curriculum to get

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

content_manager::json_service_curriculum()

Definition at line 297 of file stukowin.module.

6.2.5.4 stukowin_get_lva ($iNodeID)

JSON service for the details of a single course.

Parameters

integer	$iNodeID	The Drupal content node id of the course to get

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

content_manager::json_service_lva()

Definition at line 351 of file stukowin.module.

6.2.5.5 stukowin_theme_registry_alter (& $theme_registry)

Implements hook_theme_registry_alter.

Hook for telling the system to use our template on the stukowin custom content type.

Parameters

object	$theme_↩ registry	The entire cache of theme registry information, post-processing

Authors

Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com

Version

1.0.0 2014-07-16

Since

Commit 9f1f3db on 2014-07-05

Definition at line 632 of file stukowin.module.

6.3 Drupal2PDF

Module to create PDF documents from curricula data.

Files

- file pdf_creator.inc.php

  PDF document generation from curricula data.

Data Structures

- class overviewPDF

  Class for PDF document generation from curricula data.

Functions

- stukowin_pdf_menu ($form, &$form_state)

  Menu callback for generating PDF documents ("admin/settings/stukowin/pdf")
- stukowin_pdf_menu_submit ($form, &$form_state)

  Submit handler for stukowin_pdf_menu()

6.3.1 Detailed Description

Module to create PDF documents from curricula data.

This module contains all files, classes and methods that provide the functionality for automatically generating PDF documents from the imported curricula data.

In order to permanently and easily archive past curricula there is the option to automatically create a course overview PDF document from the imported courses. For this component to work properly, the necessary settings have to be made in advance (see module configuration image above). The administrator is provided with a new menu (at admin/settings/stukowin/pdf) where the curriculum to be archived can be selected. Once he clicks on "Create PDF", the PDF generation in overviewPDF::createPDF() is started. (All of the PDF generation code is contained in the pdf_creator.inc.php file)

The PDF generation process will flow as shown in the included FlowchartDrupal2PDF.pdf file.

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Fabian Puehringer - f.puehringer@24speed.at

6.3.2 Function Documentation

6.3.2.1 stukowin_pdf_menu ($form, & $form_state)

Menu callback for generating PDF documents ("admin/settings/stukowin/pdf")

The user can choose between all currently published curriculum taxonomies. If none are available, a warning message is displayed.

Parameters

array	$form	Form structure as given by Drupal
array	$form_state	Form state as given by Drupal

Returns

The filled out form structure

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit e6573f8 on 2014-06-30

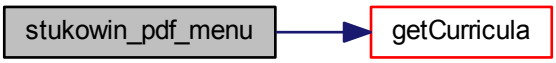
See also

stukowin_pdf_menu_submit()

Definition at line 183 of file stukowin.module.

References getCurricula().

Here is the call graph for this function:



6.3.2.2 stukowin_pdf_menu_submit ($form, & $form_state)

Submit handler for stukowin_pdf_menu()

Creates an instance of overviewPDF and starts the document generation. If successful, the full path of the generated document is displayed.

Parameters

array	$form	Form structure as given by Drupal
array	$form_state	Form state as given by Drupal

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit e6573f8 on 2014-06-30

See also

stukowin_pdf_menu()
overviewPDF::createPDF()

Definition at line 225 of file stukowin.module.

6.4 Module Core

Module that contains core functionality.

Files

- file content_manager.inc.php
  Access to curricula data.
- file stukowin.install
  Install script.

Data Structures

- class content_manager
  Access to drupal vocabularies and content nodes.

Functions

- stukowin_install ()
  Implements hook_install().
- _stukowin_installed_taxonomy_fields ()
  Contains the array of additional vocabulary fields.
- _stukowin_installed_taxonomy_instances ()
  Contains the array of additional fields for vocabulary instances.
- _stukowin_installed_fields ()
  Contains the array of additional fields for the stukowin content node type.
- _stukowin_installed_instances ()
  Contains the array of additional fields for the stukowin content node type.
- _stukowin_relation_types ()
  Contains custom relation types for recommendations and requirements.
- _stukowin_create_relation_types ()
  Creates the relation type.
- stukowin_uninstall ()
  Implements hook_uninstall().
- stukowin_admin ()
  Menu callback for module settings ("admin/stukowin/settings")
- stukowin_menu ()
  Implements hook_menu().

6.4.1 Detailed Description

Module that contains core functionality.

This module contains all files, classes and methods that provide the core functionality of the Drupal module. The members of this group ensure a working cooperation between all of the components and Drupal.

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be
Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com

6.4.2 Function Documentation

6.4.2.1 _stukowin_create_relation_types ( )

Creates the relation type.

This method creates the relation type for course recommendations and requirements

Version

1.0.0 2014-07-07

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

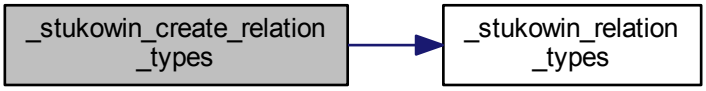
Commit d179abc on 2014-06-28

Definition at line 657 of file stukowin.install.

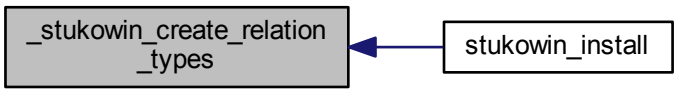
References _stukowin_relation_types().

Referenced by stukowin_install().

Here is the call graph for this function:



Here is the caller graph for this function:



6.4.2.2 _stukowin_installed_fields ( )

Contains the array of additional fields for the stukowin content node type.

Return a structured array defining the fields created by this content type. The additional fields are:

- ceusid

- code
- ects
- wst
- verantname
- verantemail
- changedate
- lvtypname
- lvtypshort
- lvatype
- typename
- ziele
- lehrinhalte
- voraussetzungen
- voraussetzung
- empfehlung
- term of the course as received from CEUS and saved in Drupal.

Returns

A structured array for Drupal field generation

Version

1.0.0 2014-07-07

Author

Konstantinos Dafalias - kdafalias@gmail.com

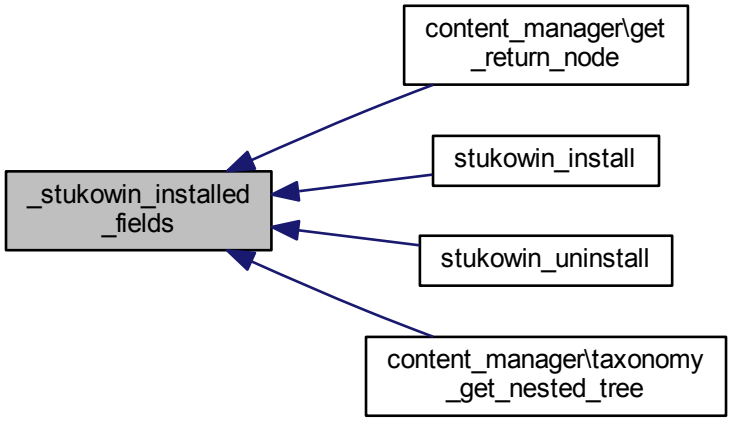
Since

Commit d179abc on 2014-06-28

Definition at line 211 of file stukowin.install.

Referenced by content_manager::get_return_node(), stukowin_install(), stukowin_uninstall(), and content_manager::taxonomy_get_nested_tree().

Here is the caller graph for this function:



6.4.2.3 _stukowin_installed_instances ( )

Contains the array of additional fields for the stukowin content node type.

Return a structured array defining and describing in greater detail the fields created by this content type. The additional fields are:

- ceusid
- code
- ects
- wst
- verantname
- verantemail
- changedate
- lvtypname
- lvtypshort
- lvatype
- typename
- ziele
- lehrinhalte
- voraussetzungen
- voraussetzung
- empfehlung
- term of the course as received from CEUS and saved in Drupal.

Returns

A structured array for Drupal field generation

Version

1.0.0 2014-07-07

Author

Konstantinos Dafalias - kdafalias@gmail.com

Authors

Fabian Puehringer - f.puehringer@24speed.at

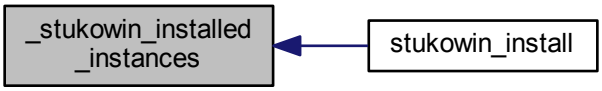
Since

Commit d179abc on 2014-06-28

Definition at line 335 of file stukowin.install.

Referenced by stukowin_install().

Here is the caller graph for this function:



6.4.2.4 _stukowin_installed_taxonomy_fields ( )

Contains the array of additional vocabulary fields.

Returns a structured array defining the custom fields for vocabularies. The additional fields are:

- faculty
- version
- type of curriculum as received from CEUS.

Returns

A structured array for Drupal field generation

Version

1.0.0 2014-07-07

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

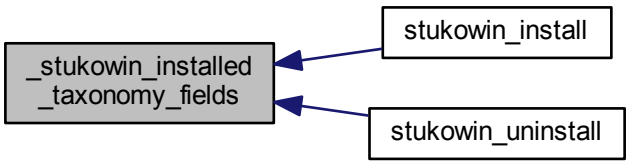
See also

_stukowin_installed_taxonomy_instances()

Definition at line 110 of file stukowin.install.

Referenced by stukowin_install(), and stukowin_uninstall().

Here is the caller graph for this function:



6.4.2.5 _stukowin_installed_taxonomy_instances ( )

Contains the array of additional fields for vocabulary instances.

Returns a structured array defining and describing in greater detail the custom fields for vocabularies. The additional fields are:

- faculty
- version
- type of curriculum as received from CEUS.

Returns

A structured array for drupal field instance generation

Version

1.0.0 2014-07-07

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

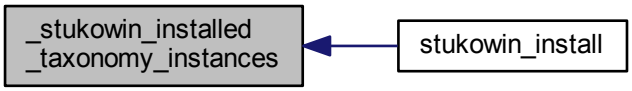
See also

_stukowin_installed_taxonomy_fields()

Definition at line 150 of file stukowin.install.

Referenced by stukowin_install().

Here is the caller graph for this function:



6.4.2.6 _stukowin_relation_types ( )

Contains custom relation types for recommendations and requirements.

Returns

An array containing the description for recommended and required prerequisite relations.

Version

1.0.0 2014-07-07

Author

Konstantinos Dafalias - kdafalias@gmail.com

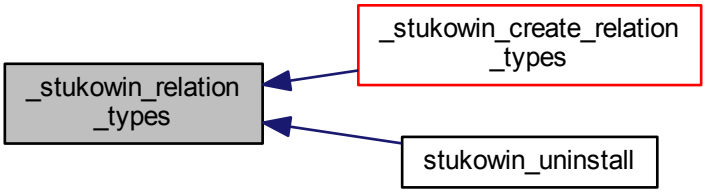
Since

Commit d179abc on 2014-06-28

Definition at line 606 of file stukowin.install.

Referenced by _stukowin_create_relation_types(), and stukowin_uninstall().

Here is the caller graph for this function:



6.4.2.7 stukowin_admin ( )

Menu callback for module settings ("admin/stukowin/settings")

Form for administration of the module settings This form has the following input fields:

- CEUS API Settings
  - URL to CEUS API
  - Username for CEUS API
  - Password for CEUS API
  - Last Update from CEUS API
- PDF Settings
  - PDF Path
  - PDF generic name

Returns

An array containing the form structure

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 105 of file stukowin.module.

6.4.2.8 stukowin_install ( )

Implements hook_install().

Sets up drupal for use with the stukowin module.

- Creates the content type for LVA nodes.
- Creates relation types for requirements and recommendations.
- Adds 3 fields to taxonomy nodes for `faculty`, `version` and `type` of curriculum

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-07

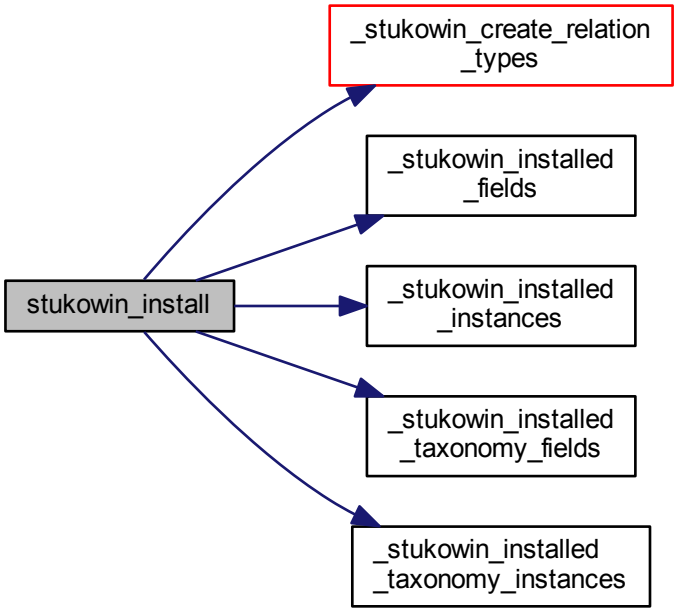
Since

Commit d179abc on 2014-06-28

Definition at line 29 of file stukowin.install.

References _stukowin_create_relation_types(), _stukowin_installed_fields(), _stukowin_installed_instances(), _stukowin_installed_taxonomy_fields(), and _stukowin_installed_taxonomy_instances().

Here is the call graph for this function:



6.4.2.9 stukowin_menu ( )

Implements hook_menu().

Registers all menu links with Drupal. The following links are registered:

- CEUS API Settings @ 'admin/stukowin/settings'
- CEUS Data import @ 'admin/settings/stukowin/import'
- PDF Generation @ 'admin/settings/stukowin/pdf'
- ITSV/Specialisation Taxonomy Creation @ 'admin/settings/stukowin/taxonomy'
- Curriculum JSON Service @ 'stukowin/crclm'
- Curricula List JSON Service @ 'stukowin/crclmlst'
- Course Detail JSON Service @ 'stukowin/lva'

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be
Werner Breuer - bluescreenwerner@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

stukowin_pdf_menu()
stukowin_taxonomy_menu()
stukowin_pre_retreive()
stukowin_admin()
stukowin_get_lva()
stukowin_get_crclm_taxonomy()
stukowin_get_crclm_list()

Definition at line 507 of file stukowin.module.

6.4.2.10 stukowin_uninstall ( )

Implements hook_uninstall().

- Deletes all content nodes
- Deletes all CEUS taxonomies
- Deletes the content type for LVA nodes

Version

1.0.0 2014-07-09

Author

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be
Werner Breuer - bluescreenwerner@gmail.com

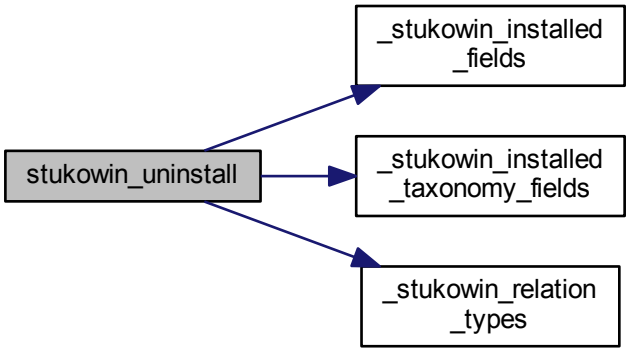
Since

Commit 2506486 on 2014-07-09

Definition at line 687 of file stukowin.install.

References _stukowin_installed_fields(), _stukowin_installed_taxonomy_fields(), and _stukowin_↩ relation_types().

Here is the call graph for this function:



6.5 Drupal2ITSV

Module to create new Drupal ITSV/specialisation vocabularies.

Functions

- stukowin_taxonomy_menu ($form, &$form_state)

  Menu callback for creating a new ITSV or specialisation vocabulary.
- stukowin_taxonomy_menu_submit ($form, &$form_state)

  Submit handler for stukowin_taxonomy_menu()

6.5.1 Detailed Description

Module to create new Drupal ITSV/specialisation vocabularies.

This module contains all files, classes and methods that provide the functionality for supporting the administrator when creating new Drupal vocabularies that represent either an ITSV ("Idealtypischer Studienverlauf") or a specialisation (mainly for Master curricula).

As CEUS does not provide any information about fields of specialisation during the master studies and ideal courses of studies, henceforth called ITSV due to its German name, it was a project requirement that new curricula can be created by the administrator for such purposes.

A freely available Drupal module called Taxonomy Manager (https://www.drupal.org/project/taxonomy_manager) gives the administrator the ability to copy vocabulary terms from one vocabulary to another, which is most of the work. Unfortunately, this process cannot be simplified any further. Nevertheless, we tried to at least automate the task of creating a new vocabulary, copying all of the information over from the source curriculum and creating top-level terms (such as "1. Semester" etc.), tasks which will be performed every time a new ITSV or specialisation has to be created.

This component handles exactly that. It inserts a new menu item (at admin/settings/stukowin/taxonomy) where the administrator can select a source curriculum to base the new one on, select whether to create an ITSV or specialisation vocabulary, enter a name and choose how many top-level terms should be inserted. Once the administrator has filled out the form, the new vocabulary is automatically created and the browser is redirected to the Taxonomy Manager's "Dual View", where the administrator can begin copying courses into the new vocabulary.

Remarks

This component does not have its own file as it does not contain a lot of code. All of its functionality is in the stukowin.module file.

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com

6.5.2 Function Documentation

6.5.2.1 stukowin_taxonomy_menu ($form, & $form_state)

Menu callback for creating a new ITSV or specialisation vocabulary.

Allows the user to

- Select a source vocabulary
- Choose which type of vocabulary to create (ITSV or specialisation)
- Set a name for the new vocabulary
- Choose how many structural terms (e.g. "1. Semester", "2. Semester") to automatically insert

Parameters

array	$form	Form structure as given by Drupal
array	$form_state	Form state as given by Drupal

Returns

The filled out form structure

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

stukowin_taxonomy_menu_submit()

Definition at line 376 of file stukowin.module.

References getCurricula().

Here is the call graph for this function:

6.5.2.2 stukowin_taxonomy_menu_submit ($form, & $form_state)

Submit handler for stukowin_taxonomy_menu()

Creates a new ITSV or specialisation vocabulary and redirects the user to the taxonomy manager's dual view.

Parameters

array	$form	Form structure as given by Drupal
array	$form_state	Form state as given by Drupal

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit 1905775 on 2014-07-02

See also

stukowin_taxonomy_menu()

Definition at line 443 of file stukowin.module.

7 Data Structure Documentation

7.1 ceus_importer Class Reference

Imports data from CEUS and stores it in the Drupal database.

Public Member Functions

- __construct ()
  Constructor.
- connect ()
  Connect to CEUS and receive Authtoken.
- get_curricula ()
  Main method: Imports curriculum data from CEUS.
- get_error ()
  Returns last error message.

Private Member Functions

- check_return_value ($sReturn)
  Checks the JSON returned by the CEUS API.
- get_detail ($iID)
  Fetches a single course from CEUS.
- save_node ($aDetail, $tid)
  Saves a course as a Drupal content node.
- has_relation ($sRelationfield)
  Checks if a course has recommendations or prerequisites.
- parse_link_term_code ($sLink)
  Parses linked web page and tries to extract a course code.
- find_nodeid_by_field ($sFieldtype, $sFieldcontent)
  Looks for Drupal content node by title or code.
- get_term_ids ($sRelationfield)
  Parses the *voraussetzungen* field of a course and tries to extract the relation.
- process_relations ()
  Creates Drupal relations out of CEUS relations.
- get_details ($aTree, $iParentID, $iCurriculumID)
  Recursive function that traverses a curriculum tree.
- check_vocabulary ($aCurriculum)
  Checks if a taxonomy vocabulary for the given curriculum exists and creates it if not.
- get_curricula_list ()
  Gets a list of all curricula (bachelor, master) from CEUS.
- get_curriculum ($iID)
  Gets one curriculum tree from CEUS.

Private Attributes

- $sCeusUrl
  Complete URL to CEUS API.
- $sUsername
  Username for CEUS API.
- $sPassword

Password for CEUS API.

- $sAuthtoken

  CEUS authentication token.
- $aFiles

  Names of the API methods.
- $sError

  Error message.
- $aLanguage

  All supported languages.
- $aVocabulary

  All vocabularies for current curricula.
- $aTerms

  All Terms for vocabularies.
- $aStats

  Import statistics.
- $aRelations

  Relations of every course (if available)

7.1.1 Detailed Description

Imports data from CEUS and stores it in the Drupal database.

This class is used to import data from the CEUS-API and saving them in the Drupal database. It also provides the change management functionality described in the system documentation.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Authors

Jakob Strasser - jakob.strasser@telenet.be
Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 49 of file ceus_importer.inc.php.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 __construct ( )

Constructor.

Creates a new instance of ceus_importer and reads the CEUS API configuration data from Drupal.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

$sCeusUrl
$sUsername
$sPassword

Definition at line 201 of file ceus_importer.inc.php.

7.1.3 Member Function Documentation

7.1.3.1 check_return_value ($sReturn) [private]

Checks the JSON returned by the CEUS API.

This method checks if the CEUS API server responded and if there was an error.

Parameters

string	$sReturn	JSON encoded string fetched from the CEUS API

Return values

decoded array	Success
false	An error occurred. The error is stored in the $sError member

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

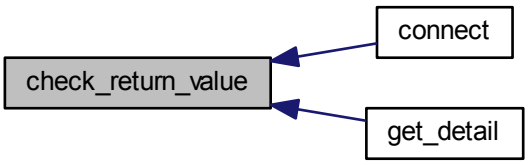
Since

Commit d179abc on 2014-06-28

Definition at line 221 of file ceus_importer.inc.php.

Referenced by connect(), and get_detail().

Here is the caller graph for this function:



7.1.3.2 check_vocabulary ($aCurriculum) [private]

Checks if a taxonomy vocabulary for the given curriculum exists and creates it if not.

This function stores the corresponding vocabulary (either a new or an existing one) into the $aVocabulary array.

Remarks

This method creates the new vocabulary with the default weight of 10, but only vocabularies with a weight below 0 will be shown publicly.

Parameters

array	$aCurriculum	Curriculum from CEUS as returned by get_curricula_list()

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 727 of file ceus_importer.inc.php.

7.1.3.3 connect ( )

Connect to CEUS and receive Authtoken.

This function tries to connect to the CEUS-API server and receives and stores the authtoken. Returns true if an authtoken was recieved otherwise it returns false.

Return values

true	Connection and authentication successful
false	Connection and/or authentication failed

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

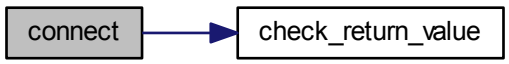
Since

Commit d179abc on 2014-06-28

Definition at line 247 of file ceus_importer.inc.php.

References check_return_value().

Here is the call graph for this function:



7.1.3.4 find_nodeid_by_field ($sFieldtype, $sFieldcontent) [private]

Looks for Drupal content node by title or code.

This function searches for the Drupal content node corresponding to a given course code or course title.

Parameters

string	$sFieldtype	The name of the field to search in. This can be either "code" or "title".
string	$sFieldcontent	The search term to look for

Return values

nid	The node's id
false	No content node with this title or code could be found

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

get_term_ids()

Definition at line 450 of file ceus_importer.inc.php.

7.1.3.5 get_curricula ( )

Main method: Imports curriculum data from CEUS.

This is the main public method of this class. It does the following things:

1. Reset the statistics
2. Imports all curricula
3. Create a new vocabulary if none exists
4. Load course data from CEUS API
5. Process relations
6. Store everything into the Drupal database

Return values

success message	The import was successful. This message contains the import statistics.
false	An error has occured and the import was not successful.

Authors

Konstantinos Dafalias - kdafalias@gmail.com

Author

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

get_curricula_list()
check_vocabulary()
get_curriculum()
get_details()
process_relations()

Definition at line 691 of file ceus_importer.inc.php.

7.1.3.6 get_curricula_list ( ) [private]

Gets a list of all curricula (bachelor, master) from CEUS.

This function retreives a list of all available curricula from the CEUS API.

Returns

Associative array containing all curricula

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 769 of file ceus_importer.inc.php.

7.1.3.7 get_curriculum ($iID) [private]

Gets one curriculum tree from CEUS.

This function retreives the complete tree of one certain curriculum from the CEUS API

Parameters

integer	$iID	CEUS id of the curriculum

Returns

Nested array of course ids

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 787 of file ceus_importer.inc.php.

7.1.3.8 get_detail ($iID) [private]

Fetches a single course from CEUS.

This function fetches a single course from the CEUS API and returns all its details in an array.

Parameters

integer	$iID	CEUS id of the course

Return values

details	2-dimensional array containing the course details. 1. dimension = language, 2. dimension = details.
false	An error occured while fetching the details

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

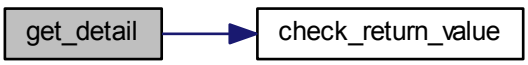
See also

get_details()

Definition at line 274 of file ceus_importer.inc.php.

References check_return_value().

Here is the call graph for this function:



7.1.3.9 get_details ($aTree, $iParentID, $iCurriculumID) [private]

Recursive function that traverses a curriculum tree.

This method goes through all the courses of a curriculum that have been returned by CEUS recursively and reads the details for each course into the $aTerms array. A vocabulary term is created for each item in the respective curriculum.

Parameters

array	$aTree	The current subtree in the curriculum
integer	$iParentID	Drupal term id of this course's parent term

integer	$iCurriculum↩ ID	CEUS id of the curriculum this tree belongs to

Return values

true	Courses have been read
false	The subtree has reached a leaf

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

get_detail()

Definition at line 638 of file ceus_importer.inc.php.

7.1.3.10 get_error ( )

Returns last error message.

This function acts as a public getter for the error message.

Returns

string

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

$sError

Definition at line 804 of file ceus_importer.inc.php.

7.1.3.11 get_term_ids ($sRelationfield) [private]

Parses the voraussetzungen field of a course and tries to extract the relation.

This function loops through all links and list items in the voraussetzugnen field and tries to extracts the referenced courses. It does this in 3 steps:

1. Extract the course names
2. Try to get the Drupal content node id through the course title
3. Try to get the Drupal content node id through the course code

Parameters

string	$sRelationfield	Textual content of the `voraussetzungen` field as received from CEUS

Returns

Drupal content node ids of related courses

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

find_nodeid_by_field()
parse_link_term_code()

Definition at line 486 of file ceus_importer.inc.php.

7.1.3.12 has_relation ($sRelationfield) [private]

Checks if a course has recommendations or prerequisites.

This function determines if a course has a relation with another course. This is the case when the $sRelationfield is not empty and does not begin with "kein" (case insensitive). Returns true if there are any rp955Td[4(r1050case.578-.0/F780g0t0/F78pr5(and)ce0Tdtiv444(e).rp95590(a)-333(i8)1(6)-5(when034nr333(con)28(t

Parameters

string	$sLink	HTML-Code with Link to CEUS entry

Return values

course code	The extracted code
false	The extraction was not successful

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

get_term_ids()

Definition at line 423 of file ceus_importer.inc.php.

7.1.3.14 process_relations ( ) [private]

Creates Drupal relations out of CEUS relations.

This procedure loops through $aRelations, parses the voraussetzungen field of each course, generates relations for required and suggested courses and stores them in the node

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

See also

get_term_ids()

Definition at line 534 of file ceus_importer.inc.php.

7.1.3.15 save_node ($aDetail, $tid) [private]

Saves a course as a Drupal content node.

If a node already exists, the function checks if the changedate has changed. If so, a new version of this conent node is created. If not, nothing is changed

Parameters

array	$aDetail	Array of course details as returned by get_detail()
array	$tid	Drupal vocabulary term id of the vocabulary term corresponding to the given course

Returns

Node id of the saved content node

Author

Konstantinos Dafalias - kdafalias@gmail.com

Authors

Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 308 of file ceus_importer.inc.php.

7.1.4 Field Documentation

7.1.4.1 $aFiles [private]

Initial value:

```
= array (
            'AUTH' => 'auth.php',
            'LIST' => 'list.php',
            'CURR' => 'curr.php',
            'DETAIL' => 'detail.php'
    )
```

Names of the API methods.

This array contains all the API method names for:

- Getting an authorization token
- Listing all curricula
- Getting one curriculum tree
- Getting one curriculum item

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 102 of file ceus_importer.inc.php.

7.1.4.2 $aLanguage [private]

Initial value:

```
= array (
            'de',
            'en'
    )
```

All supported languages.

This array contains the short names of all languages that should be imported.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 127 of file ceus_importer.inc.php.

7.1.4.3 $aRelations [private]

Relations of every course (if available)

This array contains all relations (recommended and needed) in their raw textual form It is filled during the import and evaluated at the end.

The key is the CEUS id of the cours, the value contains the content text of the voraussetzungen field from CEUS.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 186 of file ceus_importer.inc.php.

7.1.4.4 $aStats [private]

Initial value:

```
= array (
            'loaded' => 0,
            'new' => 0,
            'updated' => 0,
            'numcurrs' => 0,
            'relations' => 0
    )
```

Import statistics.

This array contains all necessary import statistics, namely:

- The number of courses loaded from CEUS
- The number of new content nodes created (= new courses)
- The number of content nodes that were updated during the import (= changes in CEUS)
- The number of curricula loaded from CEUS
- the number of relations automatically processed

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit 5ad002c on 2014-07-08

Definition at line 167 of file ceus_importer.inc.php.

7.1.4.5 $aTerms [private]

All Terms for vocabularies.

This array contains all the Drupal vocabulary terms associated with the curriculum currently being processed.

The item key represents the CEUS id of one course.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 152 of file ceus_importer.inc.php.

7.1.4.6 $aVocabulary [private]

All vocabularies for current curricula.

This array contains the respective Drupal vocabularies corresponding to the currently imported curricula.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 140 of file ceus_importer.inc.php.

7.1.4.7 $sAuthtoken [private]

CEUS authentication token.

This variable contains the authentication token as retreived from the CEUS API by calling its `auth` method.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 88 of file ceus_importer.inc.php.

7.1.4.8 $sCeusUrl [private]

Complete URL to CEUS API.

This variable contains the URL to the CEUS API as retreived from the module configuration.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 58 of file ceus_importer.inc.php.

7.1.4.9 $sError [private]

Error message.

If an error has occurred, this variable contains the error message.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 117 of file ceus_importer.inc.php.

7.1.4.10 $sPassword [private]

Password for CEUS API.

This variable contains the password for the CEUS API as retreived from the module configuration.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 78 of file ceus_importer.inc.php.

7.1.4.11 $sUsername [private]

Username for CEUS API.

This variable contains the username for the CEUS API as retreived from the module configuration.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit d179abc on 2014-06-28

Definition at line 68 of file ceus_importer.inc.php.

The documentation for this class was generated from the following file:

- ceus_importer.inc.php

7.2 content_manager Class Reference

Access to drupal vocabularies and content nodes.

Public Member Functions

- get_return_node ($iNodeID, $sLang= 'de')
  Gets course details.
- taxonomy_get_nested_tree ($vid_or_terms=array(), $max_depth=NULL, $parent=0, $parents_index=array(), $depth=0)
  Gets vocabulary tree.
- json_service_lva ($iNodeID)
  Returns course as JSON object.
- getCurricula ($sCurrType= '', $aVocabularyTypes=array('curriculum'), $sLang= 'de')
  Gets multiple curriculula.
- getUniqueMachineName ($sCoreName)
  Gets a unique and valid machine name.
- getCurriculum ($iVID)
  Gets one the curriculum.
- json_service_curriculum ($iVID)
  Returns curriculum tree as JSON array.

7.2.1 Detailed Description

Access to drupal vocabularies and content nodes.

This class is used to fetch LVA-nodes and vocabulary trees from the drupal database. As it is a utility class, it has mainly `public` functions.

Authors

Werner Breuer - bluescreenwerner@gmail.com
Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be

Version

1.0.0 2014-07-07

Since

Commit f90560a on 2014-06-28

Definition at line 32 of file content_manager.inc.php.

7.2.2 Member Function Documentation

7.2.2.1 get_return_node ($iNodeID, $sLang = 'de')

Gets course details.

This method returns the drupal content node corresponding to the given $iNodeID in the given $sLang with all fields required for displaying the course either in the PDF document or in the curriculum display (i.e. the fields set in _stukowin_installed_fields()).

Parameters

integer	$iNodeID	Drupal id of the content node
string	$sLang	Language to return. Default is 'de'.

Returns

The selected course with all the needed attributes as properties

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit 58a583a on 2014-06-29

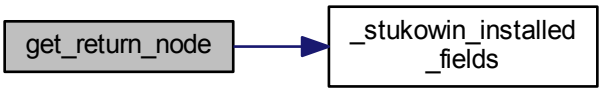
See also

_stukowin_installed_fields()
taxonomy_get_nested_tree()

Definition at line 54 of file content_manager.inc.php.

References _stukowin_installed_fields().

Here is the call graph for this function:



7.2.2.2 getCurricula ($sCurrType = '', $aVocabularyTypes = array('curriculum'), $sLang = 'de')

Gets multiple curriculula.

This method reads all curricula of curriculum type $sCurrType and vocabulary types $aVocabularyTypes in the language $sLang from the drupal database and returns them as an associative array.

Parameters

string	$sCurrType	The type of curriculum to get. Valid values are • Bachelorstudium • Masterstudium

array	$aVocabulary↩ Types	The taxonomy types to get. Default is 'curriculum'. Valid values are • curriculum • itsv • schwerpunkt
string	$sLang	The language to get the curricula in. Default is 'de'

Returns

Associative array of selected curricula

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit 1905775 on 2014-07-02

Definition at line 189 of file content_manager.inc.php.

7.2.2.3 getCurriculum ($iVID)

Gets one the curriculum.

This function fetches the curriculum object with the given vocabulary id ($iVID) from the database.

Note: it does not fetch the vocabulary tree, just its meta-information.

Parameters

integer	$iVID	The vid of the curriculum vocabulary

Returns

An associative array representing the curriculum object

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit 1905775 on 2014-07-02

Definition at line 268 of file content_manager.inc.php.

7.2.2.4 getUniqueMachineName ($sCoreName)

Gets a unique and valid machine name.

This function asserts that a machine name is valid and adds incrementing numbers at the end until it is also unique.

Parameters

string	$sCoreName	The initial name

Returns

The unique and valid machine name

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit 1905775 on 2014-07-02

See also

stukowin_taxonomy_menu_submit()
ceus_importer::check_vocabulary()

Definition at line 230 of file content_manager.inc.php.

7.2.2.5 json_service_curriculum ($iVID)

Returns curriculum tree as JSON array.

This method gets the entire tree of the curriculum with the vocabulary id `$iVID` and returns it as a JSON array.

Parameters

integer	$iVID	Drupal vocabulary id of the desired curriculum

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit f90560a on 2014-06-28

See also

taxonomy_get_nested_tree()

Definition at line 294 of file content_manager.inc.php.

7.2.2.6 json_service_lva ($iNodeID)

Returns course as JSON object.

This method fetches a single course from the drupal database and returns it as JSON.

Parameters

integer	$iNodeID	Content node id of the desired course

Author

Konstantinos Dafalias - kdafalias@gmail.com

Since

Commit f90560a on 2014-06-28

See also

stukowin_get_lva()

Definition at line 155 of file content_manager.inc.php.

7.2.2.7 taxonomy_get_nested_tree ($vid_or_terms = array(), $max_depth = NULL, $parent = 0, $parents_index = array(), $depth = 0)

Gets vocabulary tree.

This is a recursive function that reads an entire drupal vocabulary into a nested array. It can be called externally by giving the vocabulary id (vid) as the first parameter and leaving the other parameters empty.

Parameters

integer | array	$vid_or_terms	The vid of the curriculum to get. Default is an empty array
integer	$max_depth	The maximum depth of the nested array. Default is NULL
integer	$parent	The drupal term id (tid) of the next term's parent term. Default is 0
array	$parents_index	An array of all parents that have been traversed by the method in earlier recursion steps. Default is an empty array
integer	$depth	The current recursion depth. Defaul is 0

Returns

The nested array of all courses in the vocabulary

Author

Konstantinos Dafalias - kdafalias@gmail.com

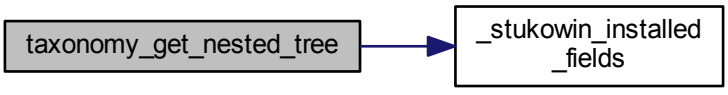
Since

Commit f90560a on 2014-06-28

Definition at line 106 of file content_manager.inc.php.

References _stukowin_installed_fields().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- content_manager.inc.php

7.3 EXSTYPE Union Reference

`#include <exparse.h>`

Data Fields

- struct Exnode_s * expr
- double floating
- struct Exref_s * reference
- struct Exid_s * id
- Sflong_t integer
- int op
- char * string
- void * user
- struct Exbuf_s * buffer

7.3.1 Detailed Description

Definition at line 208 of file exparse.h.

7.3.2 Field Documentation

7.3.2.1 struct Exbuf_s* buffer

Definition at line 222 of file exparse.h.

7.3.2.2 struct Exnode_s* expr

Definition at line 214 of file exparse.h.

7.3.2.3 double floating

Definition at line 215 of file exparse.h.

7.3.2.4 struct Exid_s* id

Definition at line 217 of file exparse.h.

7.3.2.5 Sflong_t integer

Definition at line 218 of file exparse.h.

7.3.2.6 int op

Definition at line 219 of file exparse.h.

7.3.2.7 struct Exref_s* reference

Definition at line 216 of file exparse.h.

7.3.2.8 char* string

Definition at line 220 of file exparse.h.

7.3.2.9 void* user

Definition at line 221 of file exparse.h.

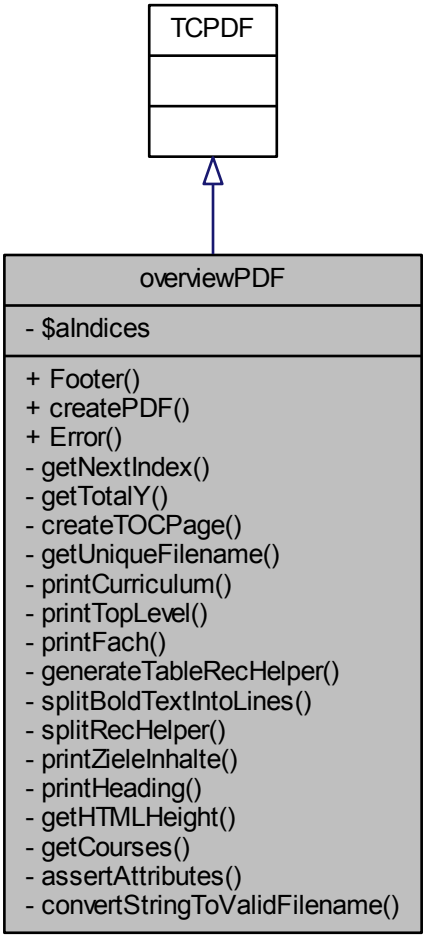
The documentation for this union was generated from the following file:

- exparse.h

7.4 overviewPDF Class Reference

Class for PDF document generation from curricula data.

Inheritance diagram for overviewPDF:



Public Member Functions

- Footer ()

  Generates Footer for each page.
- createPDF ($iVID)

  Creates the PDF document.

- Error ($msg)

  Throws exception in case of an error.

Private Member Functions

- getNextIndex ($iLevel)

  Gets index of the desired level.
- getTotalY ()

  Gets the full y-position in the document.
- createTOCPage ()

  Adds a table of contents page to the PDF document.
- getUniqueFilename ($sCurrType, $sCurrVersion)

  Creates a unique filename.
- printCurriculum ($oCurriculum)

  Prints a curriculum object and all its courses to the PDF document.
- printTopLevel ($oTopLevel)

  Decides whether to print a structural element or a subject.
- printFach ($oFach)

  Prints a subject to the PDF document.
- generateTableRecHelper ($oCourse)

  Recursive function for generating overview table.
- splitBoldTextIntoLines ($sText, $iMaxWidth)

  Inserts line breaks into bold text.
- splitRecHelper ($aWords, $iCurrIndex, $iMaxWidth)

  Recursive helper function for splitBoldTextIntoLines()
- printZieleInhalte ($oCourse)

  Prints the course goals and content of teaching and their respecitve headings to the document.
- printHeading ($sText, $iLevel, $bShowIndex=true, $sAlign= 'L', $bAddBookmark=true)

  Prints a heading to the PDF document.

Static Private Member Functions

- static getHTMLHeight ($sHTML)

  Determines height of HTML Code.
- static getCourses ($currId)

  Gets all courses for a given curriculum id.
- static assertAttributes ($oCourse)

  Guarantees the existence of the specified attributes in the course and all its children.
- static convertStringToValidFilename ($sString)

  Utility method that formats a string into a valid file name.

Private Attributes

- $aIndices = array ()

  Array of indices for continuous indexation of headings.

7.4.1 Detailed Description

Class for PDF document generation from curricula data.

This class provides the functionality for automatically generating a PDF document from a curriculum.

It is publicly accessible through the createPDF() method, which initiates and guides the PDF generation.

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Fabian Puehringer - f.puehringer@24speed.at

Version

1.0.3 2014-07-22

Since

Commit b9342d9 on 2014-06-30

See also

createPDF()

Definition at line 62 of file pdf_creator.inc.php.

7.4.2 Member Function Documentation

7.4.2.1 static assertAttributes ($oCourse) [static], [private]

Guarantees the existence of the specified attributes in the course and all its children.

This is a recursive helper method which asserts that the following attributes exist in the course object:

- title
- ects
- wst
- lvatype
- typename
- ziele
- lehrinhalte
- lvtypshort
- typename

Needed so that the other methods do not have to check everytime an attribute is accessed.

Parameters

object	$oCourse	The course to assert the attributes for

Author

Jakob Strasser - jakob.strasser@telenet.be

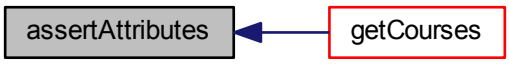
Since

Commit b9342d9 on 2014-06-30

Definition at line 638 of file pdf_creator.inc.php.

Referenced by getCourses().

Here is the caller graph for this function:



7.4.2.2 static convertStringToValidFilename ($sString) [static], [private]

Utility method that formats a string into a valid file name.

Parameters

string	$sString	The file name to validate

Returns

The valid file name created from the input

Author

Konstantinos Dafalias - kdafalias@gmail.com

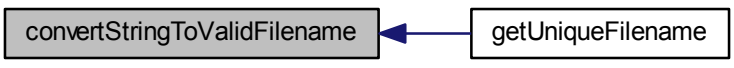
Since

Commit f157d51 on 2014-07-06

Definition at line 672 of file pdf_creator.inc.php.

Referenced by getUniqueFilename().

Here is the caller graph for this function:



7.4.2.3 createPDF ($iVID)

Creates the PDF document.

This method is the main public method of the overviewPDF class. It manages the entire document generation and saves the PDF to the preconfigured path. The steps for creating the document are as follows:

1. Set up PDF document
2. Set document meta information
3. Set up title page
4. Initiate printing of individual subjects
5. Add index page
6. Save the document

Parameters

integer	$iVID	Drupal vocabulary id of the desired Curriculum

Returns

Success message: 'PDF successfully created at ' and the filepath

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

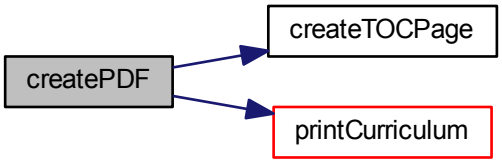
See also

stukowin_pdf_menu()
stukowin_pdf_menu_submit()

Definition at line 195 of file pdf_creator.inc.php.

References createTOCPage(), and printCurriculum().

Here is the call graph for this function:



7.4.2.4 createTOCPage ( ) [private]

Adds a table of contents page to the PDF document.

This method is called at the end of the document creation, after all subjects have been printed to the document, and inserts a table of contents (index) page as the second page in the document.

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

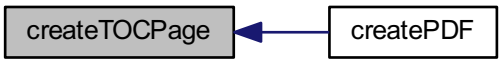
See also

TCPDF::addTOCPage()
printHeading()

Definition at line 262 of file pdf_creator.inc.php.

Referenced by createPDF().

Here is the caller graph for this function:



7.4.2.5 Error ($msg)

Throws exception in case of an error.

This method is needed for displaying errors as drupal messages, due to TCPDF normally displaying errors by itself and the dying.

Parameters

string	$msg	The error message to throw the ecxeption with

Exceptions

Exception	A new exception with the given message

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit e4fa523 on 2014-07-02

See also

TCPDF::Error()

Definition at line 245 of file pdf_creator.inc.php.

7.4.2.6 Footer ( )

Generates Footer for each page.

This method draws a 1px line across the entire page 15mm above the bottom and writes the page number in the format "current page/total pages" into the bottom right corner.

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

See also

TCPDF::Footer()

Definition at line 86 of file pdf_creator.inc.php.

7.4.2.7 generateTableRecHelper ($oCourse) [private]

Recursive function for generating overview table.

This is a recursive helper function for generating a subject overview table. It traverses the nested array of courses down to the leaves and creates a table row for each course.

- Subjects are printed in bold
- Modules are printed in italics
- Courses are printed in normal text

Parameters

object	$oCourse	The course to generate the table HTML code for

Returns

The HTML code of the table rows for this course and all its children

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

See also

printFach()

Definition at line 454 of file pdf_creator.inc.php.

References splitBoldTextIntoLines().

Referenced by printFach().

Here is the call graph for this function:

Here is the caller graph for this function:

7.4.2.8 static getCourses ($currId) [static], [private]

Gets all courses for a given curriculum id.

This method gets all the courses in a curriculum and prepares them for further processing in the PDF creation process.

Parameters

integer	$currId	The vocabulary id of the curriculum to get the courses from

Returns

The nested array of all courses in the given curriculum

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

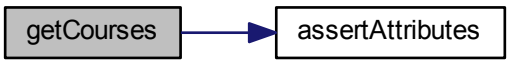
Commit b9342d9 on 2014-06-30

Definition at line 608 of file pdf_creator.inc.php.

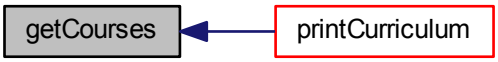
References assertAttributes().

Referenced by printCurriculum().

Here is the call graph for this function:



Here is the caller graph for this function:



7.4.2.9 static getHTMLHeight ($sHTML) [static], [private]

Determines height of HTML Code.

This utility method prints $sHTML to a test instance of overviewPDF and determines what height the HTML code has if printed.

This method is needed for evaluating whether an HTML table would fit into the remaining space on the page or if it would be broken into two pages.

Returns

The height of the HTML code

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

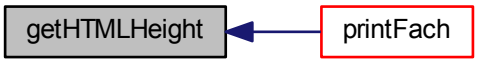
See also

printFach()

Definition at line 141 of file pdf_creator.inc.php.

Referenced by printFach().

Here is the caller graph for this function:



7.4.2.10 getNextIndex ($iLevel) [private]

Gets index of the desired level.

This utility method manages the $aIndices array and determines what heading index comes next in the given `$iLevel` (e.g. 1, 2, 3 etc.).

If some levels have been skipped (e.g. the first call to this method is with `$iLevel = 3`), it fills up the missing array values with 1.

After it has determined the correct index, it increments the corresponding value in the array by 1.

Parameters

integer	$iLevel	Level for which the index is wanted

Returns

The next index on the given level. One single integer (i.e. not "1.1.2")

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

Definition at line 113 of file pdf_creator.inc.php.

Referenced by printHeading().

Here is the caller graph for this function:

7.4.2.11 getTotalY ( ) [private]

Gets the full y-position in the document.

This utility method determines the current y position in relation to the beginning of the document, not just the current page (like TCPDF::GetY()), excluding top and bottom margins.

This method is needed to compare positions in the document across pages.

Returns

Full y-position in the entire document (not just on this page), excluding margins

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

See also

getHTMLHeight()
TCPDF::GetY()

Definition at line 168 of file pdf_creator.inc.php.

7.4.2.12 getUniqueFilename ($sCurrType, $sCurrVersion) [private]

Creates a unique filename.

Determines if a file with the standard filename as defined in the module settings already exists. If one exists, it appends a number and increases it until the name is not already taken.

Also, it creates the directory into which the document should be saved according to the module settings (if it does not already exist).

Parameters

string	$sCurrType	The type of the curriculum (Bachelorstudium, Masterstudium)
string	$sCurrVersion	The version of the curriculum (e.g. 2013W)

Returns

The unique filename

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit e8704fd on 2014-06-30

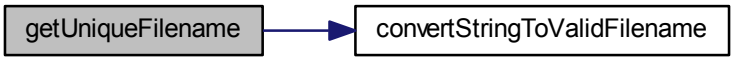
See also

stukowin_admin()

Definition at line 291 of file pdf_creator.inc.php.

References convertStringToValidFilename().

Here is the call graph for this function:



7.4.2.13 printCurriculum ($oCurriculum) [private]

Prints a curriculum object and all its courses to the PDF document.

This method does the following things:

1. Print the curriculum name as a heading
2. Create an overview table with all the subjects it contains
3. Print the details of each subject to the document

Parameters

object	$oCurriculum	The curriculum object to print

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

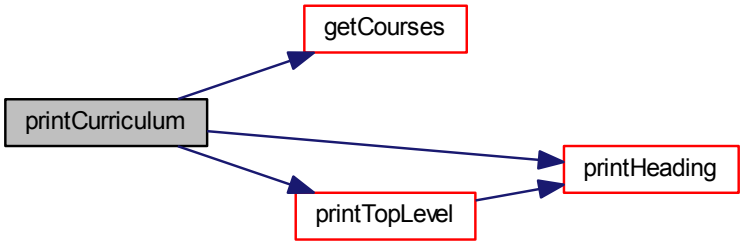
Commit b9342d9 on 2014-06-30

Definition at line 328 of file pdf_creator.inc.php.

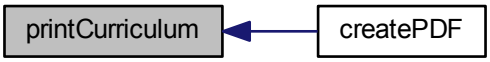
References getCourses(), printHeading(), and printTopLevel().

Referenced by createPDF().

Here is the call graph for this function:



Here is the caller graph for this function:



7.4.2.14 printFach ($oFach) [private]

Prints a subject to the PDF document.

This method prints the subject title and details, an overview of all its subcourses and their details to the document.

Parameters

object	$oFach	The course object to print

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

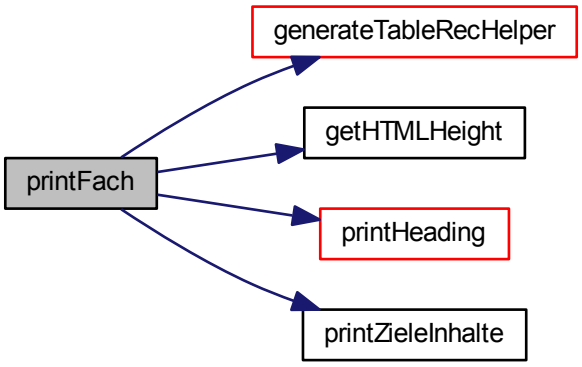
Commit b9342d9 on 2014-06-30

Definition at line 398 of file pdf_creator.inc.php.

References generateTableRecHelper(), getHTMLHeight(), printHeading(), and printZieleInhalte().

Referenced by printTopLevel().

Here is the call graph for this function:



Here is the caller graph for this function:

7.4.2.15 printHeading ($sText, $iLevel, $bShowIndex = `true`, $sAlign = `'L'`, $bAddBookmark = `true`) `[private]`

Prints a heading to the PDF document.

This method manages headings in the document. It automatically indexes the headings using getNext↩ Index() and optionally adds a bookmark for it.

Parameters

string	$sText	The heading text
int	$iLevel	The level at which the heading should be created and the bookmark set
boolean	$bShowIndex	`true` if the index should be shown in the heading
string	$sAlign	Alignment of the heading. For allowed values see TCPDF::MultiCell()
boolean	$bAdd↩ Bookmark	`true` if heading should be shown on the index page and bookmarked

Author

Jakob Strasser - `jakob.strasser@telenet.be`

Since

Commit b9342d9 on 2014-06-30

See also

createTOCPage()
getNextIndex()

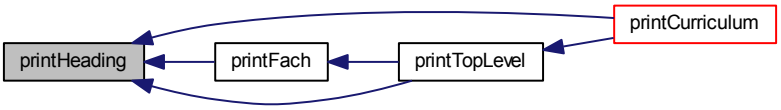
Definition at line 587 of file pdf_creator.inc.php.

References getNextIndex().

Referenced by printCurriculum(), printFach(), and printTopLevel().

Here is the call graph for this function:

Here is the caller graph for this function:



7.4.2.16 printTopLevel ($oTopLevel) [private]

Decides whether to print a structural element or a subject.

This is a dispatcher method that checks if an object is a structural element (1. Semester, 2. Semester etc.) or a course. If calls printHeading() if it is a structural element and then printFach() for all its children or just printFach() if it is a course object.

Parameters

object	$oTopLevel	The object to check

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

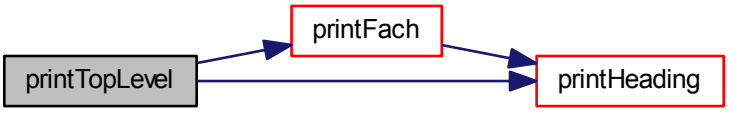
Commit a311596 on 2014-07-02

Definition at line 376 of file pdf_creator.inc.php.

References printFach(), and printHeading().

Referenced by printCurriculum().

Here is the call graph for this function:



Here is the caller graph for this function:

7.4.2.17 printZieleInhalte ($oCourse) [private]

Prints the course goals and content of teaching and their respecitve headings to the document.

If one of them is not set or empty, their heading is not printed either.

Parameters

object	$oCourse	The course to print the goals and contents for

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit b9342d9 on 2014-06-30

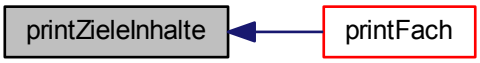
See also

printFach()

Definition at line 546 of file pdf_creator.inc.php.

Referenced by printFach().

Here is the caller graph for this function:



7.4.2.18 splitBoldTextIntoLines ($sText, $iMaxWidth) [private]

Inserts line breaks into bold text.

This method breaks the $sText into separate lines which are shorter than $iMaxWidth if printed as bold text. If one word alone is too long or the entire string is shorter than $iMaxWidth, it is not splitted.

This method is needed in generateTableRecHelper() because TCPDF does not break the words correctly in a table if the text is written in a <B> tag. This caused issues where text would overflow its table cell to the right by a few cm.

Parameters

string	$sText	The text to split
	$iMaxWidth	The The maximum width a line is allowed to have

Returns

The $sText with
 tags inserted whenever necessary

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

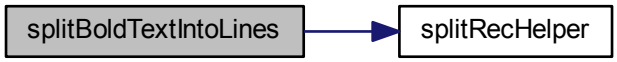
Commit 7e5b545 on 2014-07-06

Definition at line 498 of file pdf_creator.inc.php.

References splitRecHelper().

Referenced by generateTableRecHelper().

Here is the call graph for this function:



Here is the caller graph for this function:

7.4.2.19 splitRecHelper ($aWords, $iCurrIndex, $iMaxWidth) [private]

Recursive helper function for splitBoldTextIntoLines()

Parameters

array	$aWords	The array of words as split in splitBoldTextIntoLines()
integer	$iCurrIndex	The current index in the array
integer	$iMaxWidth	The maximum width a line is allowed to have

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit 7e5b545 on 2014-07-06

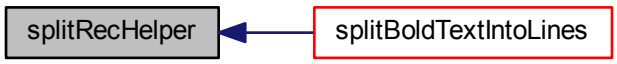
See also

splitBoldTextIntoLines()

Definition at line 520 of file pdf_creator.inc.php.

Referenced by splitBoldTextIntoLines().

Here is the caller graph for this function:



7.4.3 Field Documentation

7.4.3.1 $aIndices = array () [private]

Array of indices for continuous indexation of headings.

Every key in this array represents an indexation level (e.g. level 2 for 1.1, level 3 for 1.1.1) and every value in the array the current index for that level.

Since

Commit b9342d9 on 2014-06-30

See also

getNextIndex()

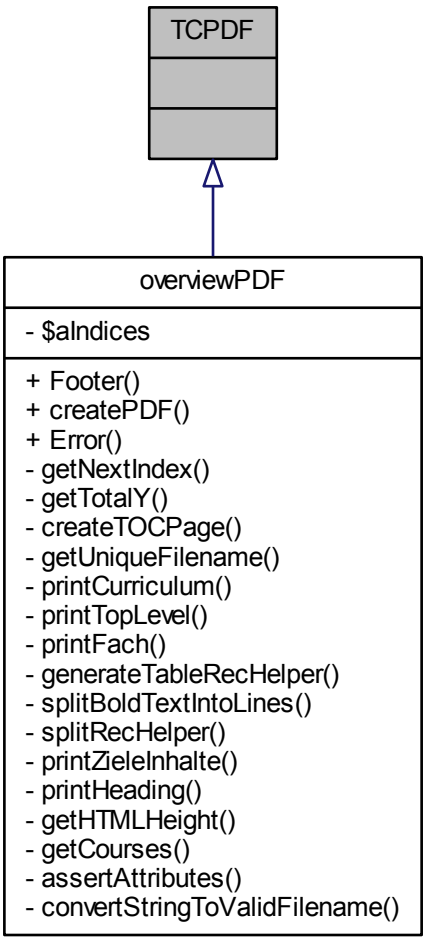
Definition at line 73 of file pdf_creator.inc.php.

The documentation for this class was generated from the following file:

- pdf_creator.inc.php

7.5 TCPDF Class Reference

Inheritance diagram for TCPDF:



The documentation for this class was generated from the following file:

- pdf_creator.inc.php

8 File Documentation

8.1 ceus_importer.inc.php File Reference

Imports data from CEUS.

Data Structures

- class ceus_importer

  Imports data from CEUS and stores it in the Drupal database.

8.1.1 Detailed Description

Imports data from CEUS.

This file manages the import of curricula data from CEUS.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Authors

Jakob Strasser - jakob.strasser@telenet.be
Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition in file ceus_importer.inc.php.

8.2 content_manager.inc.php File Reference

Access to curricula data.

Data Structures

- class content_manager

  Access to drupal vocabularies and content nodes.

8.2.1 Detailed Description

Access to curricula data.

This file contains all necessary functionality for accessing the curricula data stored in drupal

- #define LABEL 284
- #define MEMBER 285
- #define NAME 286
- #define POS 287
- #define PRAGMA 288
- #define PRE 289
- #define PRINT 290
- #define PRINTF 291
- #define PROCEDURE 292
- #define QUERY 293
- #define RAND 294
- #define RETURN 295
- #define SCANF 296
- #define SPLIT 297
- #define SPRINTF 298
- #define SRAND 299
- #define SSCANF 300
- #define SUB 301
- #define SUBSTR 302
- #define SWITCH 303
- #define TOKENS 304
- #define UNSET 305
- #define WHILE 306
- #define F2I 307
- #define F2S 308
- #define I2F 309
- #define I2S 310
- #define S2B 311
- #define S2F 312
- #define S2I 313
- #define F2X 314
- #define I2X 315
- #define S2X 316
- #define X2F 317
- #define X2I 318
- #define X2S 319
- #define X2X 320
- #define XPRINT 321
- #define OR 322
- #define AND 323
- #define NE 324
- #define EQ 325
- #define GE 326
- #define LE 327
- #define RS 328
- #define LS 329
- #define IN 330
- #define UNARY 331
- #define DEC 332
- #define INC 333
- #define CAST 334
- #define MAXTOKEN 335
- #define EXSTYPE_IS_TRIVIAL 1
- #define exstype EXSTYPE /* obsolescent; will be withdrawn */
- #define EXSTYPE_IS_DECLARED 1

Typedefs

- typedef union EXSTYPE EXSTYPE

Enumerations

- enum extokentype {
  MINTOKEN = 258, INTEGER = 259, UNSIGNED = 260, CHARACTER = 261,
  FLOATING = 262, STRING = 263, VOIDTYPE = 264, ADDRESS = 265,
  ARRAY = 266, BREAK = 267, CALL = 268, CASE = 269,
  CONSTANT = 270, CONTINUE = 271, DECLARE = 272, DEFAULT = 273,
  DYNAMIC = 274, ELSE = 275, EXIT = 276, FOR = 277,
  FUNCTION = 278, GSUB = 279, ITERATE = 280, ITERATER = 281,
  ID = 282, IF = 283, LABEL = 284, MEMBER = 285,
  NAME = 286, POS = 287, PRAGMA = 288, PRE = 289,
  PRINT = 290, PRINTF = 291, PROCEDURE = 292, QUERY = 293,
  RAND = 294, RETURN = 295, SCANF = 296, SPLIT = 297,
  SPRINTF = 298, SRAND = 299, SSCANF = 300, SUB = 301,
  SUBSTR = 302, SWITCH = 303, TOKENS = 304, UNSET = 305,
  WHILE = 306, F2I = 307, F2S = 308, I2F = 309,
  I2S = 310, S2B = 311, S2F = 312, S2I = 313,
  F2X = 314, I2X = 315, S2X = 316, X2F = 317,
  X2I = 318, X2S = 319, X2X = 320, XPRINT = 321,
  OR = 322, AND = 323, NE = 324, EQ = 325,
  GE = 326, LE = 327, RS = 328, LS = 329,
  UNARY = 331, DEC = 332, INC = 333, CAST = 334,
  MAXTOKEN = 335 }

Variables

- EXSTYPE exlval

8.3.1 Macro Definition Documentation

8.3.1.1 #define ADDRESS 265

Definition at line 132 of file exparse.h.

8.3.1.2 #define AND 323

Definition at line 190 of file exparse.h.

8.3.1.3 #define ARRAY 266

Definition at line 133 of file exparse.h.

8.3.1.4 #define BREAK 267

Definition at line 134 of file exparse.h.

8.3.1.5 #define CALL 268

Definition at line 135 of file exparse.h.

8.3.1.6 #define CASE 269

Definition at line 136 of file exparse.h.

8.3.1.7 #define CAST 334

Definition at line 201 of file exparse.h.

8.3.1.8 #define CHARACTER 261

Definition at line 128 of file exparse.h.

8.3.1.9 #define CONSTANT 270

Definition at line 137 of file exparse.h.

8.3.1.10 #define CONTINUE 271

Definition at line 138 of file exparse.h.

8.3.1.11 #define DEC 332

Definition at line 199 of file exparse.h.

8.3.1.12 #define DECLARE 272

Definition at line 139 of file exparse.h.

8.3.1.13 #define DEFAULT 273

Definition at line 140 of file exparse.h.

8.3.1.14 #define DYNAMIC 274

Definition at line 141 of file exparse.h.

8.3.1.15 #define ELSE 275

Definition at line 142 of file exparse.h.

8.3.1.16 #define EQ 325

Definition at line 192 of file exparse.h.

8.3.1.17 #define EXIT 276

Definition at line 143 of file exparse.h.

8.3.1.18 #define exstype EXSTYPE /* obsolescent; will be withdrawn */

Definition at line 230 of file exparse.h.

8.3.1.19 #define EXSTYPE_IS_DECLARED 1

Definition at line 231 of file exparse.h.

8.3.1.20 #define EXSTYPE_IS_TRIVIAL 1

Definition at line 229 of file exparse.h.

8.3.1.21 #define EXTOKENTYPE

Definition at line 40 of file exparse.h.

8.3.1.22 #define F2I 307

Definition at line 174 of file exparse.h.

8.3.1.23 #define F2S 308

Definition at line 175 of file exparse.h.

8.3.1.24 #define F2X 314

Definition at line 181 of file exparse.h.

8.3.1.25 #define FLOATING 262

Definition at line 129 of file exparse.h.

8.3.1.26 #define FOR 277

Definition at line 144 of file exparse.h.

8.3.1.27 #define FUNCTION 278

Definition at line 145 of file exparse.h.

8.3.1.28 #define GE 326

Definition at line 193 of file exparse.h.

8.3.1.29 #define GSUB 279

Definition at line 146 of file exparse.h.

8.3.1.30 #define I2F 309

Definition at line 176 of file exparse.h.

8.3.1.31 #define I2S 310

Definition at line 177 of file exparse.h.

8.3.1.32 #define I2X 315

Definition at line 182 of file exparse.h.

8.3.1.33 #define ID 282

Definition at line 149 of file exparse.h.

8.3.1.34 #define IF 283

Definition at line 150 of file exparse.h.

8.3.1.35 #define IN 330

Definition at line 197 of file exparse.h.

8.3.1.36 #define INC 333

Definition at line 200 of file exparse.h.

8.3.1.37 #define INTEGER 259

Definition at line 126 of file exparse.h.

8.3.1.38 #define ITERATE 280

Definition at line 147 of file exparse.h.

8.3.1.39 #define ITERATER 281

Definition at line 148 of file exparse.h.

8.3.1.40 #define LABEL 284

Definition at line 151 of file exparse.h.

8.3.1.41 #define LE 327

Definition at line 194 of file exparse.h.

8.3.1.42 #define LS 329

Definition at line 196 of file exparse.h.

8.3.1.43 #define MAXTOKEN 335

Definition at line 202 of file exparse.h.

8.3.1.44 #define MEMBER 285

Definition at line 152 of file exparse.h.

8.3.1.45 #define MINTOKEN 258

Definition at line 125 of file exparse.h.

8.3.1.46 #define NAME 286

Definition at line 153 of file exparse.h.

8.3.1.47 #define NE 324

Definition at line 191 of file exparse.h.

8.3.1.48 #define OR 322

Definition at line 189 of file exparse.h.

8.3.1.49 #define POS 287

Definition at line 154 of file exparse.h.

8.3.1.50 #define PRAGMA 288

Definition at line 155 of file exparse.h.

8.3.1.51 #define PRE 289

Definition at line 156 of file exparse.h.

8.3.1.52 #define PRINT 290

Definition at line 157 of file exparse.h.

8.3.1.53 #define PRINTF 291

Definition at line 158 of file exparse.h.

8.3.1.54 #define PROCEDURE 292

Definition at line 159 of file exparse.h.

8.3.1.55 #define QUERY 293

Definition at line 160 of file exparse.h.

8.3.1.56 #define RAND 294

Definition at line 161 of file exparse.h.

8.3.1.57 #define RETURN 295

Definition at line 162 of file exparse.h.

8.3.1.58 #define RS 328

Definition at line 195 of file exparse.h.

8.3.1.59 #define S2B 311

Definition at line 178 of file exparse.h.

8.3.1.60 #define S2F 312

Definition at line 179 of file exparse.h.

8.3.1.61 #define S2I 313

Definition at line 180 of file exparse.h.

8.3.1.62 #define S2X 316

Definition at line 183 of file exparse.h.

8.3.1.63 #define SCANF 296

Definition at line 163 of file exparse.h.

8.3.1.64 #define SPLIT 297

Definition at line 164 of file exparse.h.

8.3.1.65 #define SPRINTF 298

Definition at line 165 of file exparse.h.

8.3.1.66 #define SRAND 299

Definition at line 166 of file exparse.h.

8.3.1.67 #define SSCANF 300

Definition at line 167 of file exparse.h.

8.3.1.68 #define STRING 263

Definition at line 130 of file exparse.h.

8.3.1.69 #define SUB 301

Definition at line 168 of file exparse.h.

8.3.1.70 #define SUBSTR 302

Definition at line 169 of file exparse.h.

8.3.1.71 #define SWITCH 303

Definition at line 170 of file exparse.h.

8.3.1.72 #define TOKENS 304

Definition at line 171 of file exparse.h.

8.3.1.73 #define UNARY 331

Definition at line 198 of file exparse.h.

8.3.1.74 #define UNSET 305

Definition at line 172 of file exparse.h.

8.3.1.75 #define UNSIGNED 260

Definition at line 127 of file exparse.h.

8.3.1.76 #define VOIDTYPE 264

Definition at line 131 of file exparse.h.

8.3.1.77 #define WHILE 306

Definition at line 173 of file exparse.h.

8.3.1.78 #define X2F 317

Definition at line 184 of file exparse.h.

8.3.1.79 #define X2I 318

Definition at line 185 of file exparse.h.

8.3.1.80 #define X2S 319

Definition at line 186 of file exparse.h.

8.3.1.81 #define X2X 320

Definition at line 187 of file exparse.h.

8.3.1.82 #define XPRINT 321

Definition at line 188 of file exparse.h.

8.3.2 Typedef Documentation

8.3.2.1 typedef union EXSTYPE EXSTYPE

8.3.3 Enumeration Type Documentation

8.3.3.1 enum extokentype

Enumerator

MINTOKEN
INTEGER
UNSIGNED
CHARACTER

FLOATING
STRING
VOIDTYPE
ADDRESS
ARRAY
BREAK
CALL
CASE
CONSTANT
CONTINUE
DECLARE
DEFAULT
DYNAMIC
ELSE
EXIT
FOR
FUNCTION
GSUB
ITERATE
ITERATER
ID
IF
LABEL
MEMBER
NAME
POS
PRAGMA
PRE
PRINT
PRINTF
PROCEDURE
QUERY
RAND
RETURN
SCANF
SPLIT
SPRINTF
SRAND
SSCANF
SUB
SUBSTR
SWITCH
TOKENS
UNSET
WHILE

F2I
F2S
I2F
I2S
S2B
S2F
S2I
F2X
I2X
S2X
X2F
X2I
X2S
X2X
XPRINT
OR
AND
NE
EQ
GE
LE
RS
LS
UNARY
DEC
INC
CAST
MAXTOKEN

Definition at line 43 of file exparse.h.

8.3.4 Variable Documentation

8.3.4.1 EXSTYPE exlval

8.4 getopt.h File Reference

Macros

- #define GETOPT_H

Functions

- int getopt (int, char *const *, const char *)

Variables

- char * optarg
- int opterr
- int optind
- int optopt

8.4.1 Macro Definition Documentation

8.4.1.1 #define GETOPT_H

Definition at line 22 of file getopt.h.

8.4.2 Function Documentation

8.4.2.1 int getopt (int , char *const * , const char *)

8.4.3 Variable Documentation

8.4.3.1 char* optarg

8.4.3.2 int opterr

8.4.3.3 int optind

8.4.3.4 int optopt

8.5 graph.js File Reference

Script for nicely displaying CEUS data.

Functions

- jQuery (document).ready(function()
  Gets a list of all courses.
- function getCurricula (data)
  Extracts curricula from JSON.
- function fill_crclm (data)
  Extracts top-level courses and fills page with content.
- function showEmpfohlen (element)
  Shows or hides recommended courses.
- function showVoraussetzungen (element)
  Shows or hides required courses.
- function expand_reduce (element)
  Toggles the expansion/reduction of a course
- function expand_all ()
  Expands all divs.
- function reduce_all ()
  Reduces all divs.
- function expandAndScrollToElement (element, highlightColor)
  Expands the curriculum tree down to an element.
- function createDivs (kurs, level)
  Creates the s for all courses recursively.
- function createTds (kurs)
  Creates the header table cells.
- function isFullyVisible (elem)
  Checks if an element is fully visible.
- function buildRequestURL (baseUrl, type, curriculums)
  Creates the JSON request URL.
- function clearDiv ()
  Empties main

- function addResources ()

  Adds missing stylesheets and the jQuery effect library.
- function fillMissingDetail (data)

  Fetches course data for missing courses.

Variables

- var kurse = {}
- var drupal_root = ""
- var jsonCalls = []

8.5.1 Detailed Description

Script for nicely displaying CEUS data.

This script is responsible for creating the proper html/css/js needed for displaying the graphical representation of CEUS data.

In order for this script to work, a with the id "main" and the tags data-currtype and data-curriculums set needs to be present on the page. Example:

```
<div id="main" data-currtype="Bachelorstudium" data-curriculums="curriculum itsv specialisation">
</div>
```

The graphical representation will look approximately like this:

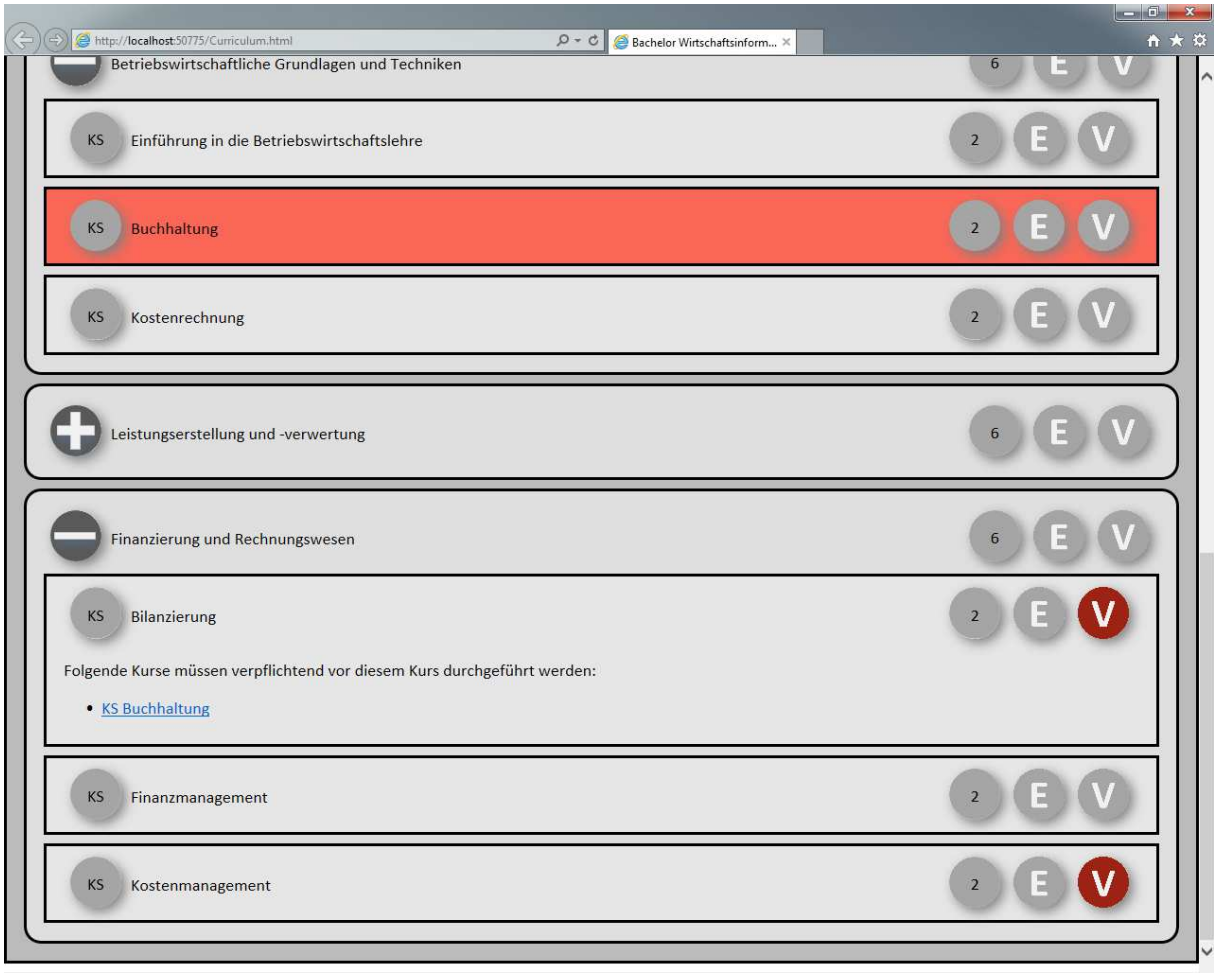


Figure 12: Graphical Display

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Version

1.0 2014-07-09

Since

Commit f90560a on 2014-06-28

Definition in file graph.js.

8.5.2 Function Documentation

8.5.2.1 function addResources ( )

Adds missing stylesheets and the jQuery effect library.

This utility procedure adds the curriculum_style.css stylesheet and the jQuery effect library to the document.

Author

Markus Gutmayer - m.gutmayer@gmail.com

Authors

Werner Breuer - bluescreenwerner@gmail.com

Since

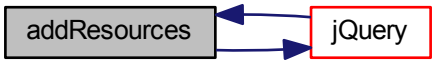
Commit 9f5e8a6 on 2014-07-05

Definition at line 645 of file graph.js.

References drupal_root, and jQuery().

Referenced by jQuery().

Here is the call graph for this function:



Here is the caller graph for this function:

8.5.2.2 function buildRequestURL (baseUrl, type, curriculums)

Creates the JSON request URL.

This utility function buils the URL for the JSON request based on the main s `data` tags.

Parameters

baseUrl	Drupal base URL to build the request on (Needed because drupal could be installed under a sub path)
type	The curriculum type to get ("Bachelorstudium","Masterstudium")
curriculums	The vocabulary types to get ("curriculum", "itsv" and/or "specialisation")

Returns

The complete request URL

Author

Werner Breuer - `bluescreenwerner@gmail.com`

Authors

Markus Gutmayer - `m.gutmayer@gmail.com`
Jakob Strasser - `jakob.strasser@telenet.be`

Since

Commit `58e2eac` on 2014-07-02

Definition at line 604 of file graph.js.

Referenced by jQuery().

Here is the caller graph for this function:

8.5.2.3 function clearDiv ( )

Empties main

This utility procedure deletes all contents except for the curriculum legend.

Author

Werner Breuer - bluescreenwerner@gmail.com

Authors

Jakob Strasser - jakob.strasser@telenet.be

Since

Commit 58e2eac on 2014-07-02

Definition at line 630 of file graph.js.

References jQuery().

Referenced by fill_crclm().

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.2.4 function createDivs (kurs, level)

Creates the s for all courses recursively.

This function creates the entire content of the graphical display recursively. For each course it creates a header table and its children.

Parameters

kurs	The course to create the for
level	The current recursion level (needed in order to decide if the element is top-level or not

Returns

The complete HTML for the given course's

Authors

Jakob Strasser - jakob.strasser@telenet.be
Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Since

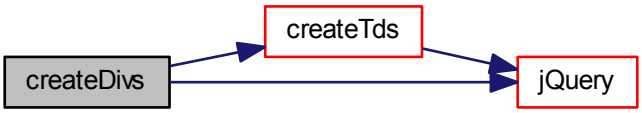
Commit 3372f36 on 2014-06-29

Definition at line 459 of file graph.js.

References createTds(), jQuery(), and kurse.

Referenced by fill_crclm().

Here is the call graph for this function:



Here is the caller graph for this function:

8.5.2.5 function createTds (kurs)

Creates the header table cells.

This utility function creates the table cells in the header used in all course, module and subject s. It also requests the details of all courses referenced (as recommendation/prerequisite) but not included in the loaded vocabulary.

Parameters

kurs	The course to create the cells for

Returns

The HTML code for the <td>s in the header

Author

Markus Gutmayer - m.gutmayer@gmail.com

Authors

Jakob Strasser - jakob.strasser@telenet.be
Werner Breuer - bluescreenwerner@gmail.com

Since

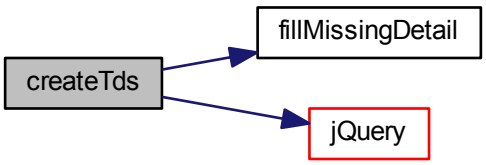
Commit 15c1f4d on 2014-06-30

Definition at line 515 of file graph.js.

References drupal_root, fillMissingDetail(), jQuery(), jsonCalls, and kurse.

Referenced by createDivs().

Here is the call graph for this function:



Here is the caller graph for this function:

8.5.2.6 function expand_all ( )

Expands all divs.

This procedure expands all divs that are currently reduced.

Author

Jakob Strasser - jakob.strasser@telenet.be

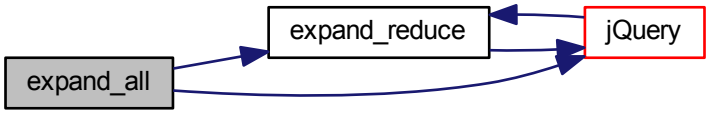
Since

Commit 3372f36 on 2014-06-29

Definition at line 383 of file graph.js.

References expand_reduce(), and jQuery().

Here is the call graph for this function:



8.5.2.7 function expand_reduce (element)

Toggles the expansion/reduction of a course

This procedure expands/reduces a course , thus showing or hiding its children.

Parameters

element	The element to expand/reduce

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

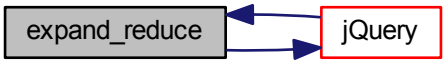
Commit 3372f36 on 2014-06-29

Definition at line 344 of file graph.js.

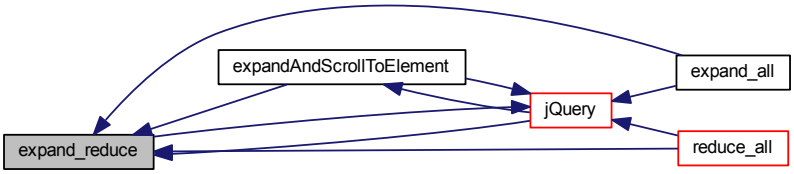
References drupal_root, and jQuery().

Referenced by expand_all(), expandAndScrollToElement(), jQuery(), and reduce_all().

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.2.8 function expandAndScrollToElement (element, highlightColor)

Expands the curriculum tree down to an element.

This procedure expands the curriculum tree down to the given element, scrolls to it and then highlights it in the given `highlightColor`.

Parameters

element	The element to expand and scroll to
highlightColor	The color to highlight the element in. Default is #90EE90.

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

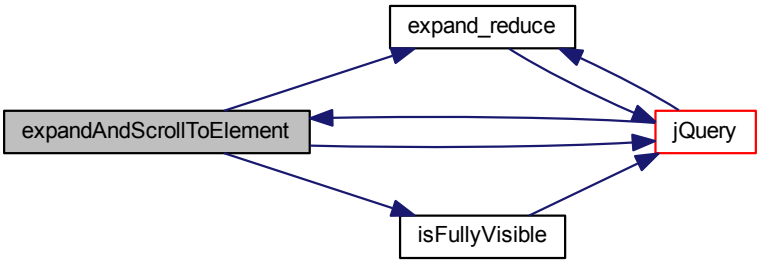
Commit 3372f36 on 2014-06-29

Definition at line 417 of file graph.js.

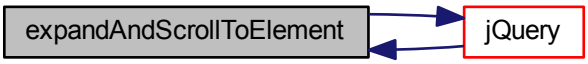
References expand_reduce(), isFullyVisible(), and jQuery().

Referenced by jQuery().

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.2.9 function fill_crclm (data)

Extracts top-level courses and fills page with content.

This event handler extracts the top-level courses out of the JSON-data and fills the page with content by calling createDivs().

Parameters

data	The received JSON file containing the curriculum tree

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Since

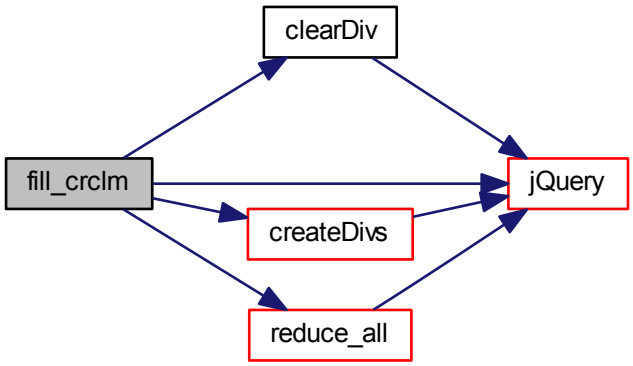
Commit f90560a on 2014-06-28

Definition at line 223 of file graph.js.

References clearDiv(), createDivs(), jQuery(), kurse, and reduce_all().

Referenced by getCurricula().

Here is the call graph for this function:



Here is the caller graph for this function:

8.5.2.10 function fillMissingDetail (data)

Fetches course data for missing courses.

This utility procedure and event handler is used for fetching the details of all those courses that are not included in the display but referenced as recommendation and/or prerequisite.

Parameters

data	The received JSON details array of one course

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

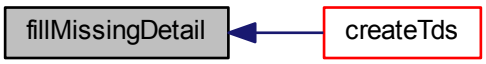
Commit f157d51 on 2014-07-06

Definition at line 673 of file graph.js.

References kurse.

Referenced by createTds().

Here is the caller graph for this function:



8.5.2.11 function getCurricula (data)

Extracts curricula from JSON.

This event handler extracts the curricula out of the JSON data, fills the select box at the top of the page and calls fill_crclm() for the first curriculum.

Parameters

data	The received JSON file containing the curricula list

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Since

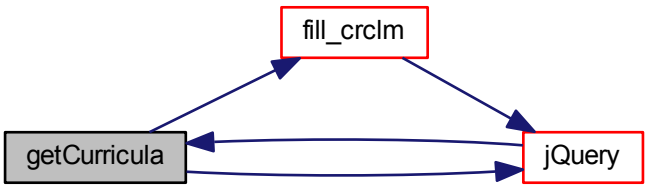
Commit f90560a on 2014-06-28

Definition at line 168 of file graph.js.

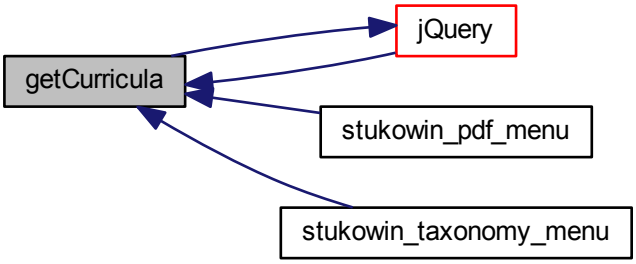
References drupal_root, fill_crclm(), jQuery(), and jsonCalls.

Referenced by jQuery(), stukowin_pdf_menu(), and stukowin_taxonomy_menu().

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.2.12 function isFullyVisible (elem)

Checks if an element is fully visible.

This utility function determines if an element is fully visible on the screen

Parameters

elem	The element to check

Return values

true	The element is fully visible
false	The element is not fully visible

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

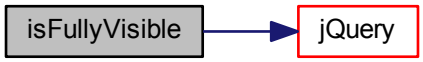
Commit 3372f36 on 2014-06-29

Definition at line 573 of file graph.js.

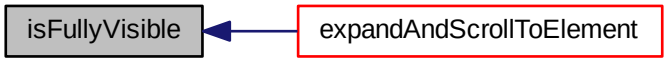
References jQuery().

Referenced by expandAndScrollToElement().

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.2.13 jQuery (document)

Gets a list of all courses.

This event handler gets a list of all courses from stukowin module's JSON service and sets up event handlers for different events.

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Since

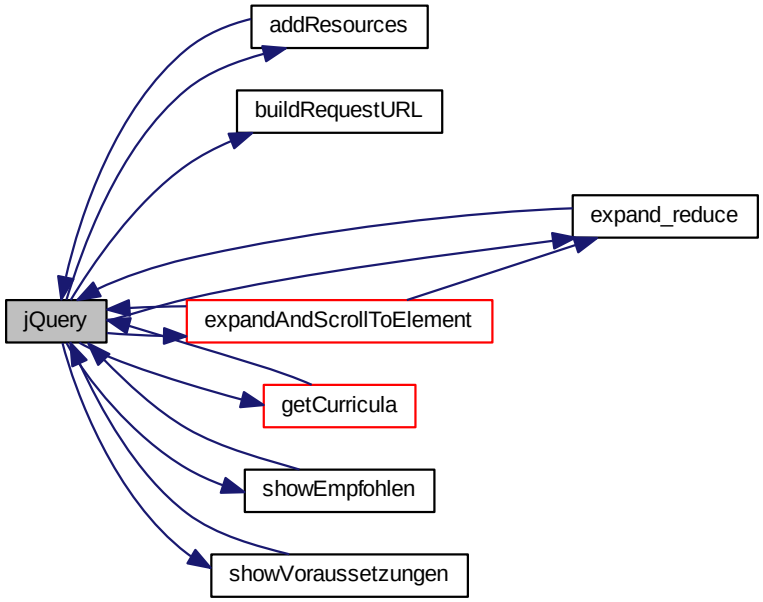
Commit f90560a on 2014-06-28

Definition at line 89 of file graph.js.

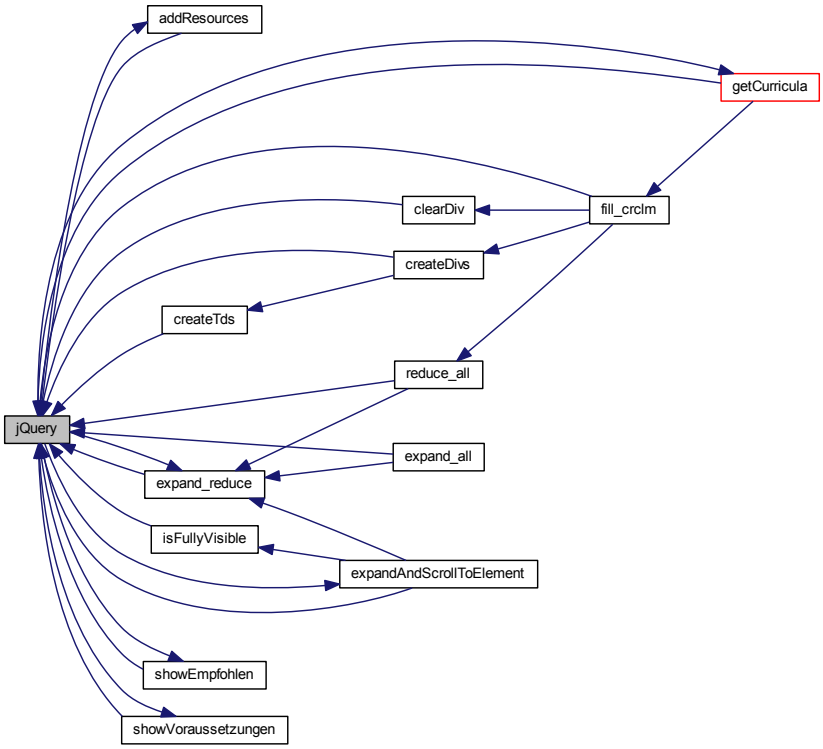
References addResources(), buildRequestURL(), drupal_root, expand_reduce(), expandAndScrollTo↩Element(), getCurricula(), showEmpfohlen(), and showVoraussetzungen().

Referenced by addResources(), clearDiv(), createDivs(), createTds(), expand_all(), expand_reduce(), expandAndScrollToElement(), fill_crclm(), getCurricula(), isFullyVisible(), reduce_all(), show↩Empfohlen(), and showVoraussetzungen().

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.2.14 function reduce_all ( )

Reduces all divs.

This procedure reduces all divs that are currently expanded.

Author

Jakob Strasser - jakob.strasser@telenet.be

Since

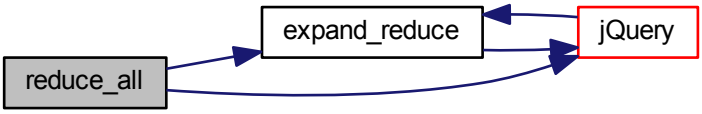
Commit d831fad on 2014-07-03

Definition at line 397 of file graph.js.

References expand_reduce(), and jQuery().

Referenced by fill_crclm().

Here is the call graph for this function:



Here is the caller graph for this function:

8.5.2.15 function showEmpfohlen (element)

Shows or hides recommended courses.

This click handler shows or hides the list of recommended courses for a given course.

Parameters

element	The element to show the recommended courses for

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Since

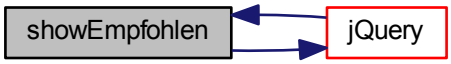
Commit 3372f36 on 2014-06-29

Definition at line 256 of file graph.js.

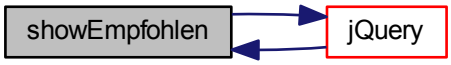
References jQuery(), and kurse.

Referenced by jQuery().

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.2.16 function showVoraussetzungen (element)

Shows or hides required courses.

This click handler shows or hides the list of required courses for a given course.

Parameters

element	The element to show the required courses for

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Markus Gutmayer - m.gutmayer@gmail.com
Werner Breuer - bluescreenwerner@gmail.com

Since

Commit 3372f36 on 2014-06-29

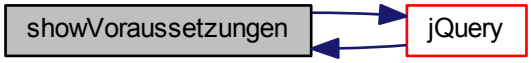
Definition at line 300 of file graph.js.

References jQuery(), and kurse.

Referenced by jQuery().

Here is the call graph for this function:

Here is the caller graph for this function:



8.5.3 Variable Documentation

8.5.3.1 var drupal_root = ""

Definition at line 75 of file graph.js.

Referenced by addResources(), createTds(), expand_reduce(), getCurricula(), and jQuery().

8.5.3.2 var jsonCalls = []

Definition at line 76 of file graph.js.

Referenced by createTds(), and getCurricula().

8.5.3.3 var kurse = {}

Definition at line 74 of file graph.js.

Referenced by createDivs(), createTds(), fill_crclm(), fillMissingDetail(), showEmpfohlen(), and showVoraussetzungen().

8.6 Mainpage.dox File Reference

8.7 pdf_creator.inc.php File Reference

PDF document generation from curricula data.

Data Structures

- class overviewPDF

  Class for PDF document generation from curricula data.

8.7.1 Detailed Description

PDF document generation from curricula data.

This file contains all necessary functionality for automatically generating PDF documents from the imported curricula data.

Author

Jakob Strasser - jakob.strasser@telenet.be

Authors

Fabian Puehringer - f.puehringer@24speed.at

Version

1.0.3 2014-07-22

Since

Commit b9342d9 on 2014-06-30

See also

overviewPDF

Definition in file pdf_creator.inc.php.

8.8 plugin.js File Reference

Registers the CKEditor plugin.

Functions

- CKEDITOR plugins add ('stukowin_curriculum',{icons: 'stukowin_curriculum', init↩ :function(editor){editor.addCommand('stukowin_curriculum_Dialog', new CKEDITOR.dialog↩ Command('stukowin_curriculum_Dialog'));editor.ui.addButton('stukowin_curriculum',{label: 'Insert Taxonomy', command: 'stukowin_curriculum_Dialog', toolbar: 'insert'});CKEDITOR.↩ dialog.add('stukowin_curriculum_Dialog', this.path+ 'dialogs/stukowin_curriculum.js');}})

8.8.1 Detailed Description

Registers the CKEditor plugin.

This file contains the commands for registering the CKEditor Plugin. It adds the button and the dialog box and adds the commands behind them.

Authors

Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com

Version

1.0.0 2014-07-01

Since

Commit 6f56770 on 2014-06-30

Definition in file plugin.js.

8.8.2 Function Documentation

8.8.2.1 CKEDITOR plugins add ('stukowin_curriculum' , {icons: 'stukowin_curriculum', init:function(editor){editor.addCommand('stukowin_curriculum_Dialog', new CKEDITOR.dialogCommand('stukowin_curriculum_Dialog'));editor.ui.addButton('stukowin_curriculum',{label: 'Insert Taxonomy', command: 'stukowin_curriculum_Dialog', toolbar: 'insert'});CKEDITOR.dialog.add('stukowin_curriculum_Dialog', this.path+ 'dialogs/stukowin_curriculum.js');}})

8.9 stukowin.install File Reference

Install script.

Functions

- stukowin_install ()

  Implements hook_install().
- _stukowin_installed_taxonomy_fields ()

  Contains the array of additional vocabulary fields.
- _stukowin_installed_taxonomy_instances ()

  Contains the array of additional fields for vocabulary instances.
- _stukowin_installed_fields ()

  Contains the array of additional fields for the stukowin content node type.
- _stukowin_installed_instances ()

  Contains the array of additional fields for the stukowin content node type.
- _stukowin_relation_types ()

  Contains custom relation types for recommendations and requirements.
- _stukowin_create_relation_types ()

  Creates the relation type.
- stukowin_uninstall ()

  Implements hook_uninstall().

8.9.1 Detailed Description

Install script.

This file contains the Drupal install script for the stukowin module.

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition in file stukowin.install.

8.10 stukowin.module File Reference

Main module file.

Functions

- stukowin_help ($path, $arg)
  Implements hook_help() Displays help information about the module.
- stukowin_admin ()
  Menu callback for module settings ("admin/stukowin/settings")
- stukowin_pdf_menu ($form, &$form_state)
  Menu callback for generating PDF documents ("admin/settings/stukowin/pdf")
- stukowin_pdf_menu_submit ($form, &$form_state)
  Submit handler for stukowin_pdf_menu()
- stukowin_pre_retreive ($form, &$form_state)
  Menu callback for import ("admin/settings/stukowin/import")
- stukowin_pre_retreive_submit ($form, &$form_state)
  Submit handler for stukowin_pre_retreive()
- stukowin_get_crclm_taxonomy ($iVID)
  JSON service for the nested array of courses details for one curriculum.
- stukowin_get_crclm_list ()
  JSON service for a list of all currently published curricula.
- stukowin_get_lva ($iNodeID)
  JSON service for the details of a single course.
- stukowin_taxonomy_menu ($form, &$form_state)
  Menu callback for creating a new ITSV or specialisation vocabulary.
- stukowin_taxonomy_menu_submit ($form, &$form_state)
  Submit handler for stukowin_taxonomy_menu()
- stukowin_menu ()
  Implements hook_menu().
- stukowin_ckeditor_plugin ()
  Implements hook_ckeditor_plugin.
- stukowin_theme_registry_alter (&$theme_registry)
  Implements hook_theme_registry_alter.

8.10.1 Detailed Description

Main module file.

This file is the central hook for the Drupal module. All functions accessible by Drupal are located in this file.

Authors

Konstantinos Dafalias - kdafalias@gmail.com
Jakob Strasser - jakob.strasser@telenet.be
Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition in file stukowin.module.

8.10.2 Function Documentation

8.10.2.1 stukowin_help ($path, $arg)

Implements hook_help() Displays help information about the module.

Parameters

string	$path	Drupal Path for help link
string	$arg	Dddional call arguments

Author

Konstantinos Dafalias - kdafalias@gmail.com

Version

1.0.0 2014-07-16

Since

Commit d179abc on 2014-06-28

Definition at line 74 of file stukowin.module.

8.11 stukowin_curriculum.js File Reference

The stukowin_curriculum dialog definition.

Functions

- CKEDITOR dialog add ('stukowin_curriculum_Dialog', function(editor){return{title: 'Insert a Curriculum View', minWidth:400, minHeight:200, contents:[{id: 'currdialog', elements:[{type: 'radio', id: 'currtype', label: 'Select the curricula type you want to display:', items:[['Bachelor', 'Bachelorstudium'], ['Master', 'Masterstudium']], 'default': 'Bachelorstudium'},{type: 'fieldset', label: 'Taxonomy types to display', children:[{type: 'vbox', padding:0, children:[{type: 'checkbox', id: 'normal', label: 'Normal', 'default': 'checked'},{type: 'checkbox', id: 'itsv', label: 'ITSV'},{type: 'checkbox', id: 'specialisation', label: 'Specialisation'}]}]}]}], onOk↩ :function(){var dialog=this;var drupal_root=Drupal.settings.basePath;var div=editor.document.↩ createElement('div');var currType=dialog.getValueOf('currdialog', 'currtype');var taxonomy↩ TypeSet=false;var curriculums="";if(dialog.getValueOf('currdialog', 'normal')){curriculums+=

'curriculum';taxonomyTypeSet=true;}if(dialog.getValueOf('currdialog', 'itsv')){if(curriculums) curriculums+= ' ';curriculums+= 'itsv';taxonomyTypeSet=true;}if(dialog.getValueOf('currdialog', 'specialisation')){if(curriculums) curriculums+= ' ';curriculums+= 'specialisation';taxonomyTypeSet=true;}if(!taxonomyTypeSet){alert('Error:Please select at least one taxonomy type to display');return;}div.setAttribute('data-currtype', currType);div.setAttribute('data-curriculums', curriculums);div.setAttribute('id',"curriculum_display");var curr_div=new CKEDITOR.dom.element('div');curr_div.setAttribute('id', 'curriculum_legende') curr_div.setStyle('padding-left', '1.5em');var table=new CKEDITOR.dom.element('table');table.id="Legende";var row1=new CKEDITOR.dom.element('tr');table.append(row1);var cell11=new CKEDITOR.dom.element('td');row1.append(cell11);var cell12=new CKEDITOR.dom.element('td');row1.append(cell12);var cell13=new CKEDITOR.dom.element('td');row1.append(cell13);var cell14=new CKEDITOR.dom.element('td');row1.append(cell14);var row2=new CKEDITOR.dom.element('tr');table.append(row2);var cell21=new CKEDITOR.dom.element('td');row2.append(cell21);var cell22=new CKEDITOR.dom.element('td');row2.append(cell22);var cell31=new CKEDITOR.dom.element('td');row2.append(cell31);var cell32=new CKEDITOR.dom.element('td');row2.append(cell32);var cell23=new CKEDITOR.dom.element('td');row1.append(cell23);var cell24=new CKEDITOR.dom.element('td');row1.append(cell24);var cell33=new CKEDITOR.dom.element('td');row2.append(cell33);var cell34=new CKEDITOR.dom.element('td');row2.append(cell34);var plusIcon=new CKEDITOR.dom.element('img');plusIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Plus.png");plusIcon.setStyle('width', '3em');plusIcon.setStyle('height', '3em');var minusIcon=new CKEDITOR.dom.element('img');minusIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Minus.png");minusIcon.setStyle('width', '3em');minusIcon.setStyle('height', '3em');var vorIcon=new CKEDITOR.dom.element('img');vorIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Voraussetzung.png");vorIcon.setStyle('width', '3em');vorIcon.setStyle('height', '3em');var empfIcon=new CKEDITOR.dom.element('img');empfIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin Empfohlen.png");empfIcon.setStyle('width', '3em');empfIcon.setStyle('height', '3em');var ectsIcon=new CKEDITOR.dom.element('img');ectsIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/ECTS.png");ectsIcon.setStyle('width', '3em');ectsIcon.setStyle('height', '3em');var voIcon=new CKEDITOR.dom.element('img');voIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/V300.png");voIcon.setStyle('width', '3em');voIcon.setStyle('height', '3em');cell11.append(plusIcon);cell12.appendText("Aufklappen");cell13.append(empfIcon);cell14.appendText("Empfohlene Voraussetzungen anzeigen");cell21.append(minusIcon);cell22.appendText("Zuklappen");cell23.append(ectsIcon);cell24.appendText("ECTS");cell31.append(vorIcon);cell32.appendText("Verpflichtende Voraussetzungen anzeigen");cell33.append(voIcon);cell34.appendText("LVA-Typ");div.append(curr_div);curr_div.append(table);editor.insertElement(div);}};})

8.11.1 Detailed Description

The stukowin_curriculum dialog definition.

This file contains the dialog definition for our CKEditor plugin. It is called once a user clicks on the button in the CKEditor's menu bar.

Remarks

For this plugin to work, it has to be enabled in the CKEditor configuration.

Authors

Werner Breuer - bluescreenwerner@gmail.com
Markus Gutmayer - m.gutmayer@gmail.com
Jakob Strasser - jakob.strasser@telenet.be
Fabian Puehringer - f.puehringer@24speed.at

Version

1.0.0 2014-07-01

Since

Commit 6f56770 on 2014-06-30

Definition in file stukowin_curriculum.js.

8.11.2 Function Documentation

8.11.2.1 CKEDITOR dialog add ('stukowin_curriculum_Dialog' , function(editor){return{title: 'Insert a Curriculum View', minWidth:400, minHeight:200, contents:[{id: 'currdialog', elements:[{type: 'radio', id: 'currtype', label: 'Select the curricula type you want to display:', items:[['Bachelor', 'Bachelorstudium'], ['Master', 'Masterstudium']], 'default': 'Bachelorstudium'},{type: 'fieldset', label: 'Taxonomy types to display', children:[{type: 'vbox', padding:0, children:[{type: 'checkbox', id: 'normal', label: 'Normal', 'default': 'checked'},{type: 'checkbox', id: 'itsv', label: 'ITSV'},{type: 'checkbox', id: 'specialisation', label: 'Specialisation'}]}]}]}], onOk:function(){var dialog=this;var drupal_root=Drupal.settings.basePath;var div=editor.document.createElement('div');var currType=dialog.getValueOf('currdialog', 'currtype');var taxonomyTypeSet=false;var curriculums="";if(dialog.getValueOf('currdialog', 'normal')){curriculums+= 'curriculum';taxonomyTypeSet=true;}if(dialog.getValueOf('currdialog', 'itsv')){if(curriculums) curriculums+= ' ';curriculums+= 'itsv';taxonomyTypeSet=true;}if(dialog.getValueOf('currdialog', 'specialisation')){if(curriculums) curriculums+= ' ';curriculums+= 'specialisation';taxonomyTypeSet=true;}if(!taxonomyTypeSet){alert('Error:Please select at least one taxonomy type to display');return;}div.setAttribute('data-currtype', currType);div.setAttribute('data-curriculums', curriculums);div.setAttribute('id',"curriculum_display");var curr_div=new CKEDITOR.dom.element('div');curr_div.setAttribute('id', 'curriculum_legende') curr_div.setStyle('padding-left', '1.5em');var table=new CKEDITOR.dom.element('table');table.id="Legende";var row1=new CKEDITOR.dom.element('tr');table.append(row1);var cell11=new CKEDITOR.dom.element('td');row1.append(cell11);var cell12=new CKEDITOR.dom.element('td');row1.append(cell12);var cell13=new CKEDITOR.dom.element('td');row1.append(cell13);var cell14=new CKEDITOR.dom.element('td');row1.append(cell14);var row2=new CKEDITOR.dom.element('tr');table.append(row2);var cell21=new CKEDITOR.dom.element('td');row2.append(cell21);var cell22=new CKEDITOR.dom.element('td');row2.append(cell22);var cell31=new CKEDITOR.dom.element('td');row2.append(cell31);var cell32=new CKEDITOR.dom.element('td');row2.append(cell32);var cell23=new CKEDITOR.dom.element('td');row1.append(cell23);var cell24=new CKEDITOR.dom.element('td');row1.append(cell24);var cell33=new CKEDITOR.dom.element('td');row2.append(cell33);var cell34=new CKEDITOR.dom.element('td');row2.append(cell34);var plusIcon=new CKEDITOR.dom.element('img');plusIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Plus.png");plusIcon.setStyle('width', '3em');plusIcon.setStyle('height', '3em');var minusIcon=new CKEDITOR.dom.element('img');minusIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Minus.png");minusIcon.setStyle('width', '3em');minusIcon.setStyle('height', '3em');var vorIcon=new CKEDITOR.dom.element('img');vorIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Voraussetzung.png");vorIcon.setStyle('width', '3em');vorIcon.setStyle('height', '3em');var empfIcon=new CKEDITOR.dom.element('img');empfIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/Empfohlen.png");empfIcon.setStyle('width', '3em');empfIcon.setStyle('height', '3em');var ectsIcon=new CKEDITOR.dom.element('img');ectsIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/ECTS.png");ectsIcon.setStyle('width', '3em');ectsIcon.setStyle('height', '3em');var voIcon=new CKEDITOR.dom.element('img');voIcon.setAttribute('src', drupal_root+"sites/all/modules/stukowin/images/V300.png");voIcon.setStyle('width', '3em');voIcon.setStyle('height', '3em');cell11.append(plusIcon);cell12.appendText("Aufklappen");cell13.append(empfIcon);cell14.appendText("Empfohlene Voraussetzungen anzeigen");cell21.append(minusIcon);cell22.appendText("Zuklappen");cell23.append(ectsIcon);cell24.appendText("ECTS");cell31.append(vorIcon);cell32.appendText("Verpflichtende Voraussetzungen anzeigen");cell33.append(voIcon);cell34.appendText("LVA-Typ");div.append(curr_div);curr_div.append(table);editor.insertElement(div);}};}
)