EXT: CSV Connector Service

Extension Key: svconnector_csv
Language: en
Keywords: forDevelopers, forIntermediates

Table of Contents

EXT: CSV Connector Service....1
Introduction....3
Questions and support....3
Keeping the developer happy....3
Installation....4
Configuration....5
Developer's manual....6
To-Do list....7
Changelog....8

Introduction

This extension implements a specific connector service for reading flat files, in particular in CSV format. Although connector services were originally designed to connect to remote sources, using them to read flat files provides a developer with a standardized way of accessing such files.

Questions and support

If you have any questions about this extension, please ask them in the TYPO3 English mailing list, so that others can benefit from the answers. Please use the bug tracker on forge.typo3.org to report problem or suggest features (http://forge.typo3.org/projects/extension-svconnector_csv/issues).

Keeping the developer happy

If you like this extension, do not hesitate to rate it. Go the Extension Repository, search for this extension, click on its title to go to the details view, then click on the "Ratings" tab and vote (you need to be logged in). Every new vote keeps the developer ticking. So just do it!

You may also take a step back and reflect about the beauty of sharing. Think about how much you are benefiting and how much yourself is giving back to the community.

Installation

Install this extension and you can start using it's API for reading flat files inside your own code. It requires extension "svconnector" which provides the base for all connector services.

Configuration

The various "fetch" methods of the CSV connector take the same parameters:

Parameter	Type	Description	Default
filename	string	This is the name of the file to read. It can be any file in the paths accepted by TYPO3. It can use the "EXT:..." syntax.	
delimiter	string	The character used to separate the various fields on each line of the file	, (comma)
text_qualifier	string	The character used to wrap text fields	" (double quote)
encoding	string	Encoding of the data found in the file. This value must match any of the encoding values or their synonyms found in class t3lib_cs. Note that this means pretty much all the usual encodings. If unsure look at array t3lib_cs::synonyms.	
skip_rows	int	The number of rows to ignore at the beginning of the file. This has an additional special meaning if larger than 0. The CSV connector will take the first line and read column labels from it. These labels will then be used as keys in the array of data returned.	

The data is read using the PHP function `filegetcsv()` which takes care of the line endings. It also receives the delimiter and text_qualifier as parameters. Once the data is read it is converted from the encoding given as a parameter to the current encoding (either FE or BE, as stored in `lang::charSet`), if they are not the same. If the encoding parameter is left empty, no conversion takes place.

Developer's manual

Reading a flat file using the CSV connector service becomes a really easy task. The first step is to get the proper service object:

```
$services = t3lib_extMgm::findService('connector', 'csv');
if ($services === false) {
      // Issue an error
} else {
      $connector = t3lib_div::makeInstanceService('connector', 'csv');
}
```

On the first line, you get a list of all services that are of type "connector" and subtype "csv". If the result if false, it means no appropriate services were found and you probably want to issue an error message.

On the contrary you are assured that there's at least one valid service and you can get an instance of it by calling `t3lib_div::makeInstanceService()`.

The next step is simply to call the appropriate method from the API – with the right parameters – depending on which format you want to have in return. For a PHP array:

```
$parameters = array(
            'file' => 'path/to/your/file',
            'delimiter' => "\t",
            'text_qualifier' => '',
            'encoding' => 'utf-8',
            'skip_rows' => 1,
      );
$data = $connector->fetchArray($parameters);
```

In this example we declare the file as using tabs as delimiter and no text qualifier. Furthermore the file is declared as being encoded in UTF-8 and its first line should be ignored.

Let's assume the file looks something like this:

```
last_name   first_name
Wellington  Nicholas
Oshiro      Ki
Duncan      Dwayne
```

The resulting array stored in `$data` in the example above will be:

0	last_name	Wellington
	first_name	Nicholas
1	last_name	Oshiro
	first_name	Ki
2	last_name	Duncan
	first_name	Dwayne

The `fetchRaw()` method returns a two-dimensional array with one entry per line in the file and in each of these an entry per column. The `fetchXML()` method returns the array created by `fetchArray()` transformed to XML using `t3lib_div::array2xml_cs()`.

To-Do list

There is a roadmap on Forge for the continuing development of this extension:

http://forge.typo3.org/projects/extension-svconnector_csv/roadmap

Changelog

Version	Changes:
1.1.0	Added throwing of exceptions when reading the file fails
1.0.0	Updated to the modified svconnector base API Raised status to stable
0.1.0	First public release