

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| **Document title** | **Storyline Table** |
| **Document reference number** | **UEL -** |
| **Project name** | **UNISA e-Content 20140363** |
| **Unisa Department** | **Department of Economics** |

Table of Contents

1 Description of the component 5

2 How to Install The component 5

3 how to use the component 5

Document sign-off

This document has been seen and accepted by the following people:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Date | Signature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Compiler | Purpose | Change Summary |
| <yyyymmdd> | Name Surname | Document Created,  Internal Review,  External Review,  etc | Initial release,  Updates based on internal review,  Updates to section xxx,  etc |
|  |  |  |  |
|  |  |  |  |

# Description of component

# Storyline Table is a feature that allows a user to create storyline by importing a CSV format of the story content.

# How to Install the Storyline Table Component

### Step 1.1

Install Laravel framework. The following command will install the latest version of Laravel with the project name LaravelDefault.

composer create-project laravel/laravel LaravelDefault

### Step 1.2

Download Zip from [StorylineTable repo.](https://github.com/EonConsulting/storyline-table)

## Step 2 - Configuration:

### Step 2.1

Create a folder called "Packages" inside of the root of the Laravel project.

### Step 2.2

Paste the folder from the Git repo (Step 1.2) inside the Packages folder.

### Step 2.3

Inside the root Laravel project, open the composer.json file. In that file, look for:

"autoload" : {

"psr-4": {

...

Inside of the psr-4 tag, load the following:

"EONConsulting\\Storyline\\Table\\": "packages/storyline-table/src/"

### Step 2.4 - Use as a global function instead of by namespace (Optional)

## Step 3 - Registering the Service Provider

### Step 3.1

Open up config/app.php. In the providers array, enter the following:

***\EONConsulting\Storyline\Table\StorylineTableServiceProvider::class****,*

## Step 4 - Tsugi

To be able to use the tsugi, the APP\_URL in your env file needs to be correct. So change it to the URL you are using for your testing environment.

## Step 5 - The Finale

In the command line / terminal, enter the following command in the root of the Laravel project:

composer dump-autoload -o

## How to use SMS- Email Component

***Config file***Package config can be found in csv.php file under config directory (after you published it).  
Config file contains default values for delimiter, enclosure and escape parameters. You can set default values here and skip passing  
additional parameters to open and create methods (we discuss them later).  
  
***Convert encoding***Common issue when working with CSV files generated by Excel is encoding. Excel exports CSV file encoded with windows-1250 character set  
while most of PHP applications use UTF-8. To solve this issue, you can set encoding option in config file. You set your encoding  
preferences separately for reader and writer.  
  
 *'encoding' => [  
 'reader' => [  
 'enabled' => true,  
 'from' => 'CP1250',  
 'to' => 'UTF-8'  
 ],  
 'writer' => [  
 'enabled' => true,  
 'from' => 'UTF-8',  
 'to' => 'CP1250'  
 ]  
 ]*As you can see in the example above, Reader will convert windows-1250 encoding to UTF-8, while Writer will do this in opposite way.  
You don't have to use both options. You can set encoding conversion only for one class - reader or writer.  
  
***Reader***First you need to open CSV file.  
  
 *$reader = CsvReader::open('/path/to/file.csv');*If you need to change delimiter, enclosure or escape you can do it by passing proper values to <code>open</code> method.  
More information about these values can be found here - [http://php.net/manual/en/function.fgetcsv.php](*http://php.net/manual/en/function.fgetcsv.php*).  
  
 *$reader = CsvReader::open('/path/to/file.csv', ';', '\'', '\\\\');*Having your CSV file opened you can read it line after line  
  
 *while (($line = $reader->readLine()) !== false) {  
 print\_r($line);  
 }*or you could read whole file at once  
  
 *print\_r($reader->readAll());*If your CSV file contains header line, you can convert it into array keys for each line.  
  
 *$reader = CsvReader::open($file, ';');  
 $header = $reader->getHeader();  
 print\_r($header);  
 print\_r($reader->readAll());*Don't forget to close file after you're done with your work.  
  
 *$reader->close();****Writer***At first you need to create new file  
   
 *$writer = CsvWriter::create('/path/where/your/file/will/be/saved.csv');*File path is optional. If you won't provide it, CsvWriter will use memory as storege for added content.  
  
If you need to change delimiter, enclosure or escape you can do it by passing proper values to <code>create</code> method.  
More information about these values can be found here - [http://php.net/manual/en/function.fputcsv.php](*http://php.net/manual/en/function.fputcsv.php*).  
  
 *$writer = CsvWriter::create('/path/to/file.csv', ';', '\'', '\\\\');*To add data into CSV file you can use <code>writeLine</code> or <code>writeAll</code> methods.  
  
 *$writer->writeLine(['some', 'data']);  
 $writer->writeLine(['another', 'line']);  
   
 $writer->writeAll([  
 ['some', 'data'],  
 ['another', 'line'],  
 ]);*To display data added to CSV file, you can use flush method.  
  
 *echo $writer->flush();*Don't forget to close file after you're done with your work.  
  
 *$writer->close();*