

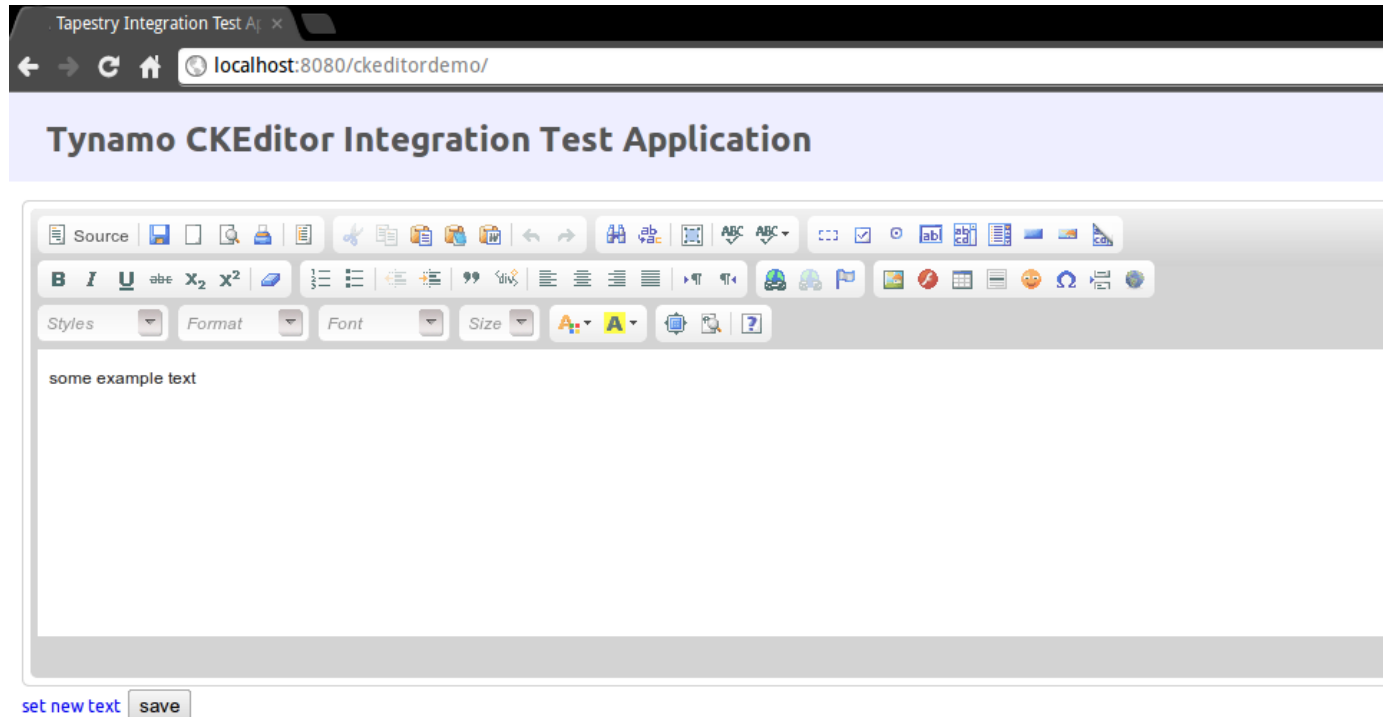
# tapestry-ckeditor guide

## Overview

The tapestry-ckeditor module integrates the powerful CKEditor rich text editor component with the Apache Tapestry5 web application framework.

The component is implemented as a tapestry mixin which can be applied only to the TextArea component. The integration with CKEditor is meant to be seamless such that the text area values are replaced with the ckeditor contents with no additional value submission and handling necessary.

Here's how it looks like:



## Usage

```
<dependency>
  <groupId>org.tynamo</groupId>
  <artifactId>tapestry-ckeditor</artifactId>
  <version>0.0.1</version>
</dependency>
```

An example page template:

### tapestry-ckeditor example

```
<t:form>
  <t:textarea t:value="contents" t:mixins="tynamo/ckeditor" parameters="{ 'toolbar':
  'Full' }" />
</t:form>
```

## Configuration

The module doesn't require any mandatory configuration. The configuration for the CKEditor is described in the [documentation](#) for the CKEDITOR.config namespace. There are two ways to make configuration changes for the tapestry-ckeditor mixin:

1. Per instance configuration: If there is need to make a change for a particular instance of the CKEditor component this can be achieved by using the **parameters** parameter of the mixin, which is of type Map<String, ?>. Example:

#### tapestry-ckeditor example

```
<t:form>
  <t:textarea t:mixins="tynamo/ckeditor" parameters="{ 'toolbar': 'Full' }" />
</t:form>
```

2. Global configuration: Changes of the CKEDITOR.config object are reflected in all editor instances, if not specified otherwise for a particular instance. the configuration can be made in a separate js file that is used by setting the **CKEDITOR.config.customConfig** property. Example:

#### config.js

```
CKEDITOR.editorConfig = function( config )
{
  // Define changes to default configuration here. For example:
  config.toolbar = 'Basic';
};
```

#### MyPage.tml

```
<t:form>
  <t:textarea t:mixins="tynamo/ckeditor" parameters="prop:CKEditorParameters" />
</t:form>
```

#### MyPage.java

```
public class MyPage
{
  @Inject
  @Path("config.js")
  private Asset config;

  public Map<String, Object> getCKEditorParameters()
  {
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("customConfig", config.toClientURL());
    return map;
  }
}
```

