Table of Contents

RenderPlugin **1**
- Description 1
- Syntax 1
- Usage 2
- Examples 2
  - Fetching a message 2
  - Previewing TML 2
- Installation Instructions 2
- Plugin Info 3

RenderPlugin

Render WikiApplications asynchronously

Description


Powered by
WikiRing Consultants

This plugin implements three simple REST handlers to get content from Foswiki. For example, these can be used to easily replace a server-side INCLUDE with a functional equivalent using an ajax call to one of these REST handlers.

This comes in handy to asynchronously load content in a tabbed interface, loading subsequent content for a treeview, a menu widget, or tooltips.

You may also use it to just render a snippet of TopicMarkup on the server and return a HTML preview back to the browser.

The REST handlers of this plugin have been designed to be rather generic. So, besides some very basic parameter checking, most error handling is up to the callback analysing the return value of the REST handler. The main purpose is best described as being an intermediate transport device.

Compare this to other plugins that implement more specific REST handlers. As such, they are rather tailored to one specific service and thus will be dealing with error cases on the server side in much more detail.

The advantage of using this plugin's REST handler is that you don't have to implement service backends using perl. Instead, you can write TopicFunctions that contain the WikiApplication to be executed when it is called by an appropriate ajax request. This basically is the client side equivalent of a parametrized `INCLUDE`, as it is used on server side WikiApplications (see examples below).

Syntax

The available REST handlers are:

Handler	Parameters	Description
`expand`	`text`	expands common variables in the submitted `text` fragment
`render`	`text`	same as `expand` but also renders TopicMarkup converting it to HTML
`tag`	`name`, `param`, `<named-params>*`, `render`	this is a convenience function to expand one specific variable instead of having to build up a snippet of code that you want to `expand` or `render`; The `param` parameter specifies the default parameters to this tag; the `named-params` are additional named parameters; the `render` flag indicates if the result is to be expanded only or if it should be renderd into HTML; default is off, that is only expand common tags but don't interpret TopicMarkup
`template`	`name`, `expand`, `render`	this handler loads the template `name` and expands the macro given in `expand`; the `render` flag indicates whether the result is to be rendered in addition to common tags being interpreted

Usage

The REST handlers are called like this:

```
%SCRIPTURLPATH{"rest"}/RenderPlugin/render?text=%ENCODE{"This is _very_ cool."}%


%SCRIPTURLPATH{"rest"}/RenderPlugin/tag?name=INCLUDE;param=Applications.RenderWebTagCloud;
```

(equivalent to a `%INCLUDE{"Applications.RenderWebTagCloud" search="screenshot"}%`)

This can be schedule using jQuery like this:

```
<div id="container"></div>

<script type="text/javascript">
$(function() {
  $("#container").load("<url-to-rest-handler->");
});
</script>
```

Examples

Fetching a message

```
%STARTSECTION{"message"}%
*Hello World!* :)
%ENDSECTION{"message"}%
```

function getTheMessage() { $("#message").load("/bin/rest/RenderPlugin/tag", { name: 'INCLUDE', param: 'System.RenderPlugin', section: 'message', render: true }, function () { $(this).show() }); } function clearTheMessage() { $("#message").empty().hide(); }

Previewing TML

function preview() { var text = $("#mytext").val(); $("#mypreview").load("/bin/rest/RenderPlugin/render", { 'text': text, 'topic': 'System.RenderPlugin' }); }

Okay, your turn to create an example calling SEARCH.

Installation Instructions

You do not need to install anything in the browser to use this extension. The following instructions are for the administrator who installs the extension on the server.

Open configure, and open the "Extensions" section. Use "Find More Extensions" to get a list of available

extensions. Select "Install".

If you have any problems, or if the extension isn't available in `configure`, then you can still install manually from the command-line. See http://foswiki.org/Support/ManuallyInstallingExtensions for more help.

Plugin Info

Plugin Author:	Michael Daum
Copyright ©:	2006-2009, Michael Daum http://michaeldaumconsulting.com
License:	GPL (GNU General Public License)
Plugin Version:	3.0
Change History:	
12 May 2009:	added `template` handler
24 Apr 2009:	converted to foswiki plugin
07 Jan 2009:	added upload rest handler; fixes for foswiki and FastCGIContrib
11 Jul 2008:	initial version
Dependencies:	**Name** / **Version** / **Description**: Foswiki::Plugins::JQueryPlugin / >=0.5 / Optional.
Plugin Home:	Foswiki:Extensions/RenderPlugin
Support:	Foswiki:Support/RenderPlugin

This topic: System > RenderPlugin
Topic revision: r3 - 16 Sep 2009 - 22:13:00 - AdminUser

FOSWIKI Copyright © by the contributing authors. All material on this site is the property of the contributing authors.
Ideas, requests, problems regarding Wiki? Send feedback