Sitecore Website Prototype

A sitecore module to easily create and setup prototype for a website.

Version: 1.0.0.0

Contents

Author

Vipin Banka
vipbanka@yahoo.com
github:
https://github.com/vipin-banka/Sitecore-WebsitePrototype

Introduction

This sitecore module allows to create prototype of the website with less efforts. This module is useful in design and development phase of a sitecore website for development teams. Development teams can quickly start with it without additional steps and efforts.

Benefits

There are following benefits with this module.

1. Works on actual presentation set on the website pages and there is no need to setup separate templates and presentation in sitecore hence it saves a lots of time.
2. Defines a good physical folder structure that is more aligned to sitecore content structure and hence improves maintainability.
3. Works on presentation component item level. Supports prototype for layouts, sublayouts and renderings.
4. Only works if prototype for the component exists, if prototype for a component does not exist than it just renders the original layout or subalyout/rendering that is placed on the presentation.
5. Supports both ASP.NET Web Form and ASP.NET MVC.
6. Allows to open all website pages in prototype mode by default, useful for demo purpose.
7. Several options available to tune sitecore to show prototype only for specific websites, pages or subalyouts/renderings.
8. A fully functional website prototype can be created.
9. The efforts of creating the prototype will not be a waste when development starts for the original website.
10. Task can be divided easily between frontend and backend developers so a better utilization of the available skills is possible.

Concept

The concept here is to replace the presentation component on the page with the prototype of the component whenever the page is requested to show the prototype. The replacement of component in presentation only works at runtime for that page request only and it does not change and save anything to the sitecore items. The replacement also works only if the prototype for the component exist, if not than original component will be kept in the presentation.

Component Prototype

There are following two approaches this module supports to define the prototype for a component.

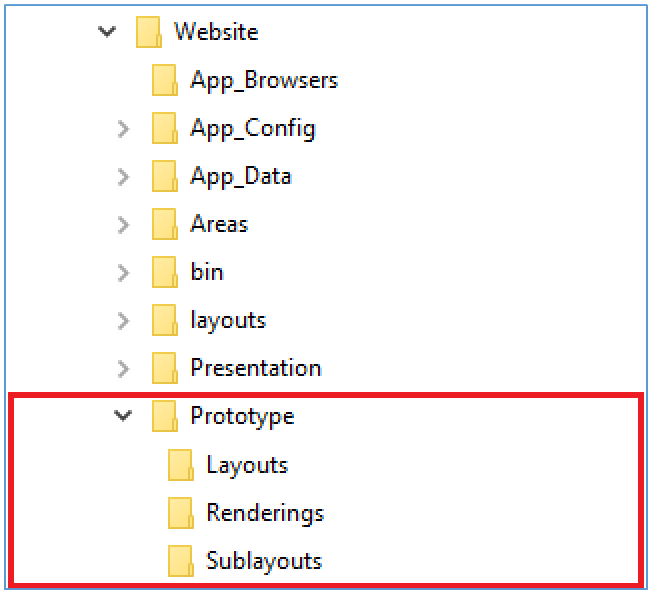
Html file
A html file with the component html markup can be created and saved at a physical file location by following some conventions as described in the Html File Convention section below. And That's it! With this html file approach nothing else needs to done.

Prototype Sitecore Component
A different sitecore component can be created in sitecore, this different component needs to be configured with mapping information. Configuration can be done using the modules settings under the system section. Basically you have setup the folder level paths mappings. You can also setup the mapping based on the ids. This approach will only be useful for very specific scenarios and for most of the cases html file approach will work. For example, let's say the technical architect has defined a rendering requirement where rendering contains some UI elements as well as one or more sitecore placeholders to add more renderings inside of it.

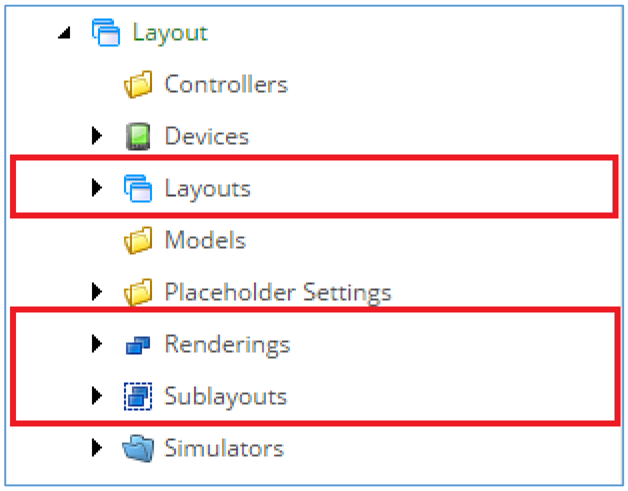
Html File Convention
The conventions are related to saving the html file.

1. **File Name:**
   The name of the file should be same as subalyout/rendering item name given in sitecore. for example if the rendering name is "Feature Callout" than file name should be "Feature Callout.html".

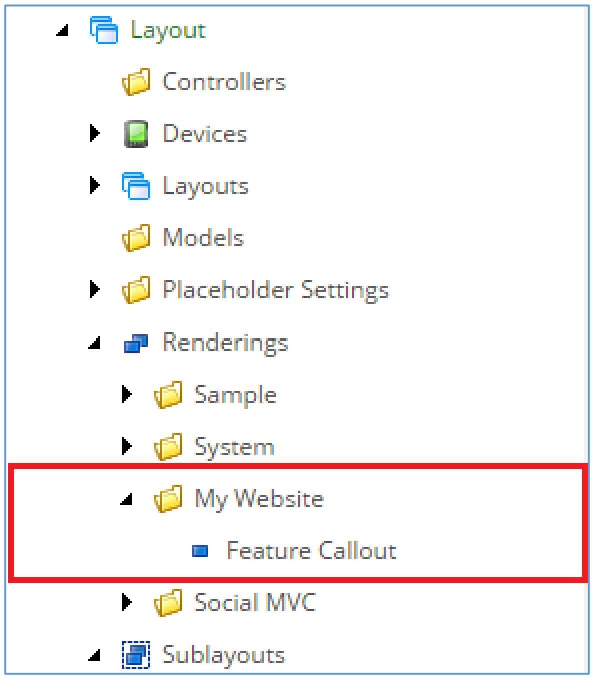
2. **File Location:**
   This module installs following folders in the root of the website:



   As you can see there is specific folder for each presentation component.
   This actually mirrors the sitecore item folders under Layout section in sitecore.



Folder for your file should be type of presentation component your html file belongs to. Rest of the file path should follow the same folder structure as created in the sitecore.
For example:
I have a rendering at following sitecore path
/sitecore/Layout/Renderings/My Website/Feature Callout



For this rendering the html file physical location should be
/Prototype/Renderings/My Website/Feature Callout.html

Demo

In this section we will simply create a prototype for a rendering using the html file approach. This demo is focused to give a demonstration on uses of the module only and there will not be any fancy thing.

Let's assume that we have to build following:

1. Create a rendering for Feature Callout, that should show an image, content, and link to a page.
2. Create a page to place this rendering.

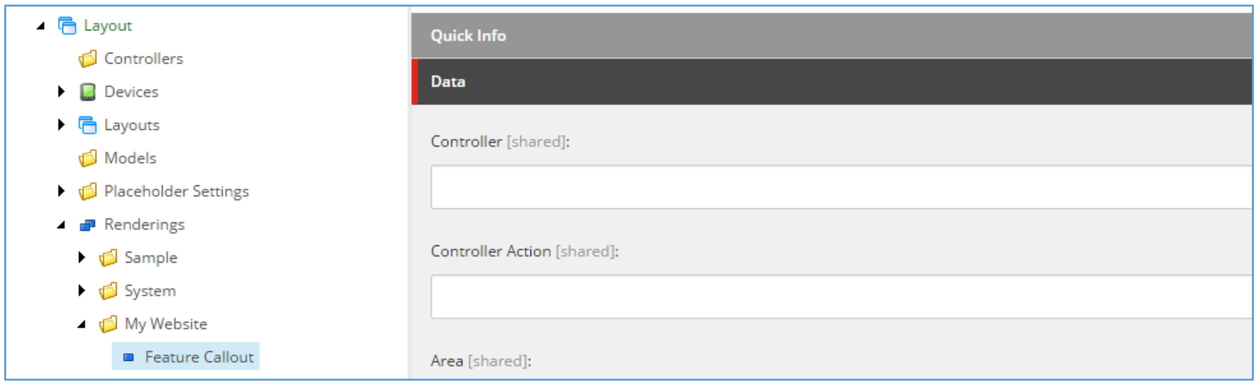
Follow below steps:

1. Install the module.
2. Publish the site changes.
3. Create a MVC layout item in sitecore. If you already have one you can use that.
4. For this demo purpose I am going to use the simple layout for now so use the below html markup for your MVC layout if you are creating a new one.

```
@using Sitecore.Data.Items
@using Sitecore.Mvc

<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <div>
    @Html.Sitecore().Placeholder("main")
  </div>
</body>
</html>
```

5. Create a sitecore item for view rendering.



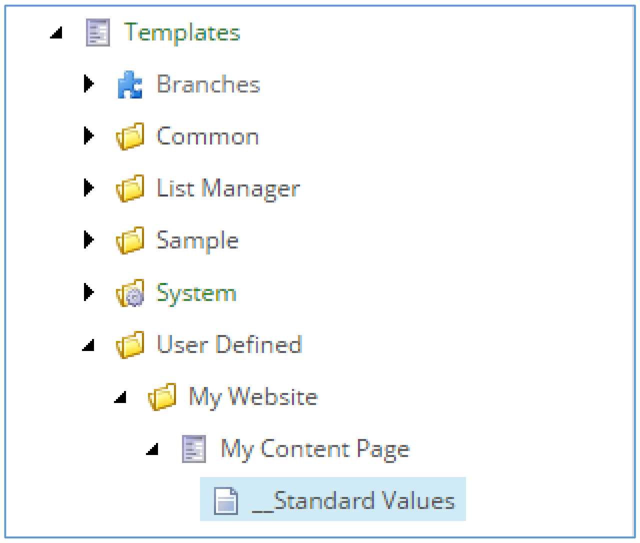
Controller [shared]:

Controller Action [shared]:

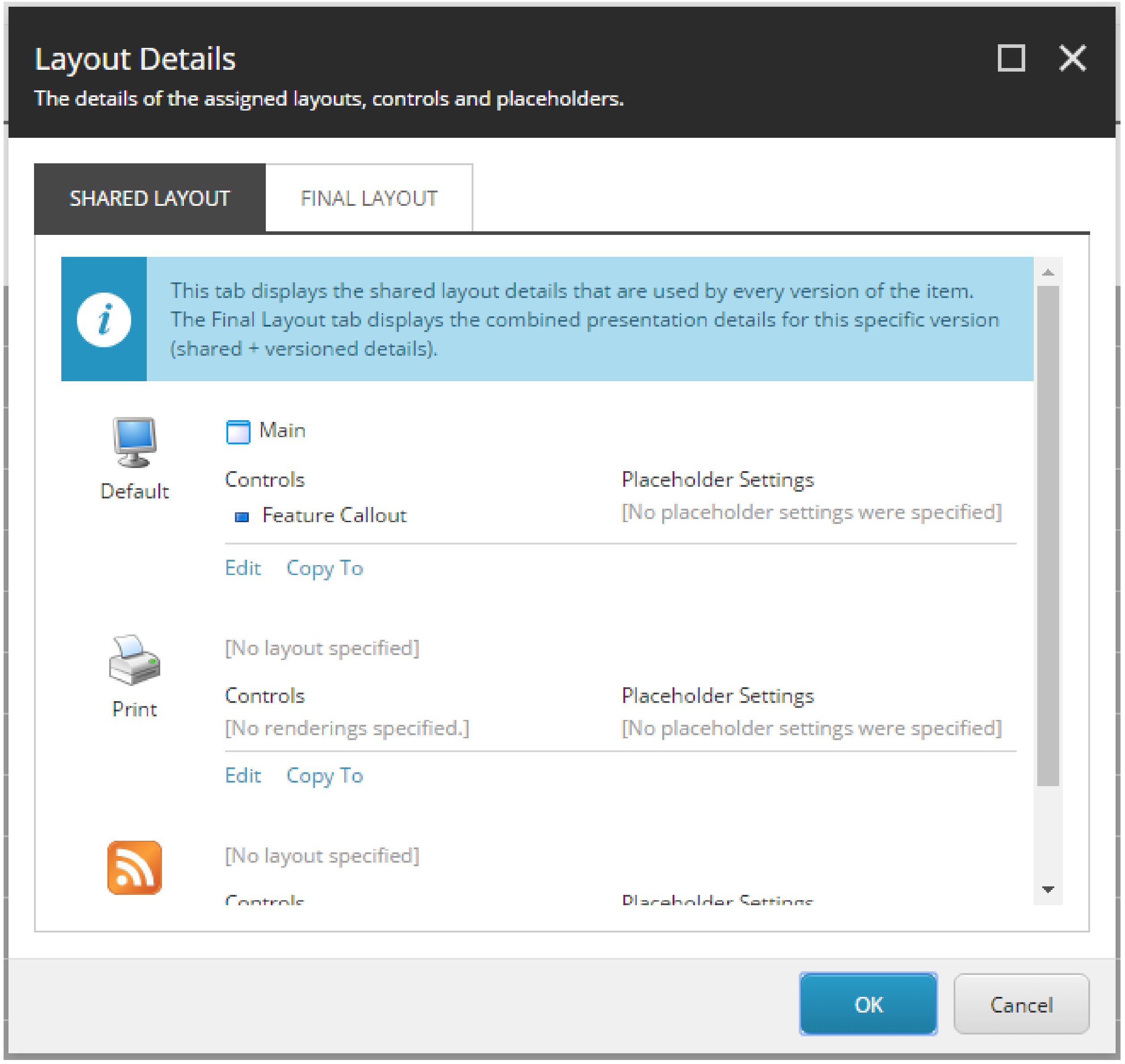
Area [shared]:

Notice that we have not given any controller and action information as the purpose is to create the prototype only now but utilize the effort when actual development starts. Once the development start the team can fill the required details.

6. Create a page template in sitecore.



7. Set the presentation for the standard template value, in the presentation select the rendering created in above steps.



Layout Details

The details of the assigned layouts, controls and placeholders.

This tab displays the shared layout details that are used by every version of the item. The Final Layout tab displays the combined presentation details for this specific version (shared + versioned details).

Default

Main

Controls: Feature Callout

Placeholder Settings: [No placeholder settings were specified]

Print

[No layout specified]

Controls: [No renderings specified.]

Placeholder Settings: [No placeholder settings were specified]

[No layout specified]

You have to use the correct placeholder names while adding the presentation component.

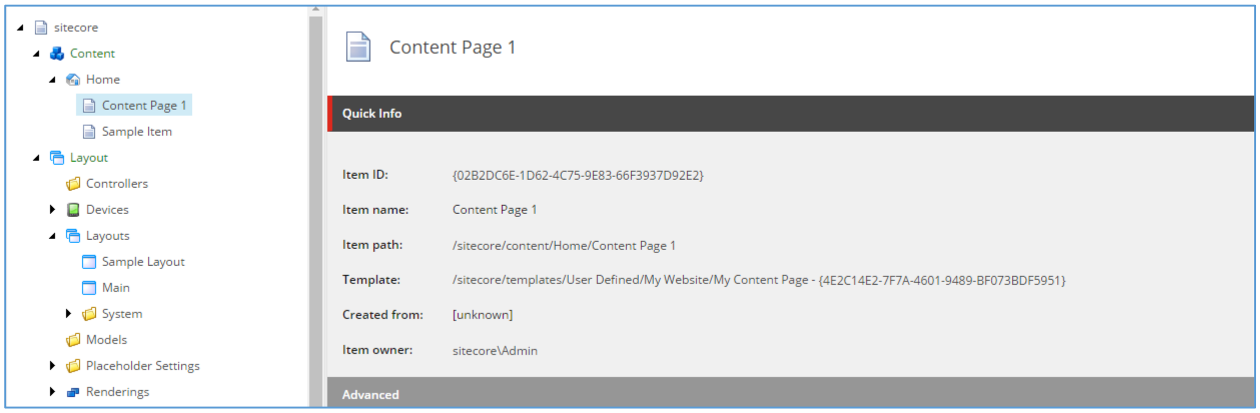
8. Create the prototype html file for the rendering.

```
<h1>Hello from Feature Callout Component Prototype</h1>
```

For the sake of the demo I am just adding a sample content.

9. Save it under the /Prototype folder as per the convention.
as per convention our file should be saved in "/Prototype/Renderings/My Website" folder with name "Feature Callout.html".

10. Create the page item under the website.

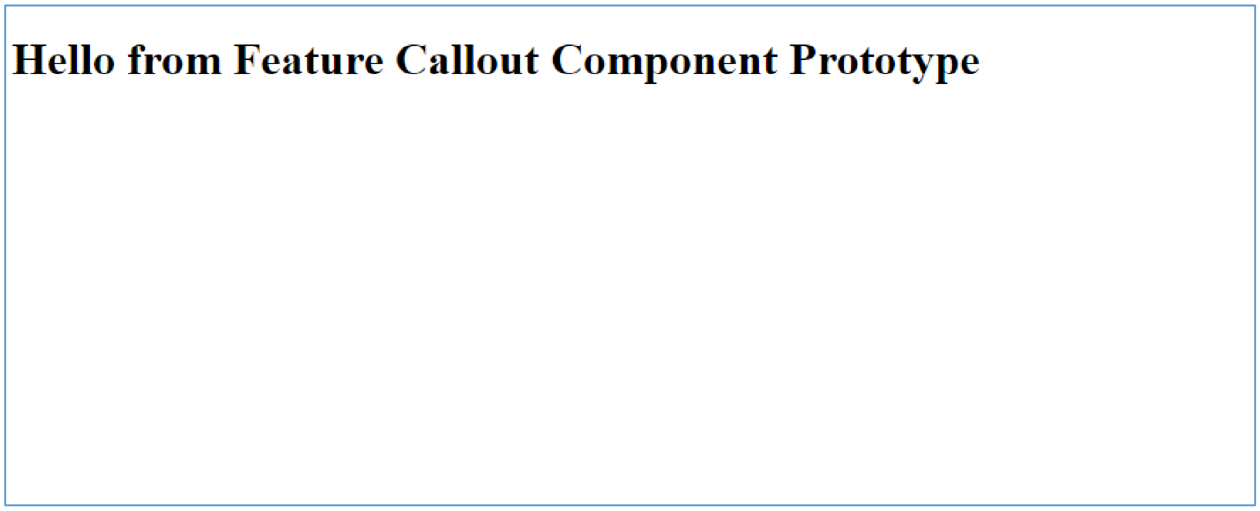


Item ID: {02B2DC6E-1D62-4C75-9E83-66F3937D92E2}
Item name: Content Page 1
Item path: /sitecore/content/Home/Content Page 1
Template: /sitecore/templates/User Defined/My Website/My Content Page - {4E2C14E2-7F7A-4601-9489-BF073BDF5951}
Created from: [unknown]
Item owner: sitecore\Admin

11. Publish sitecore changes and make sure the html files are deployed to the website correctly.

12. Open the page in browser with a query string value “r_mode=prototype”.
    e.g.
    http://module.sitecore.prototype.local/content%20page%201?r_mode=prototype

13. You will see the prototype content of the html file.



Hello from Feature Callout Component Prototype

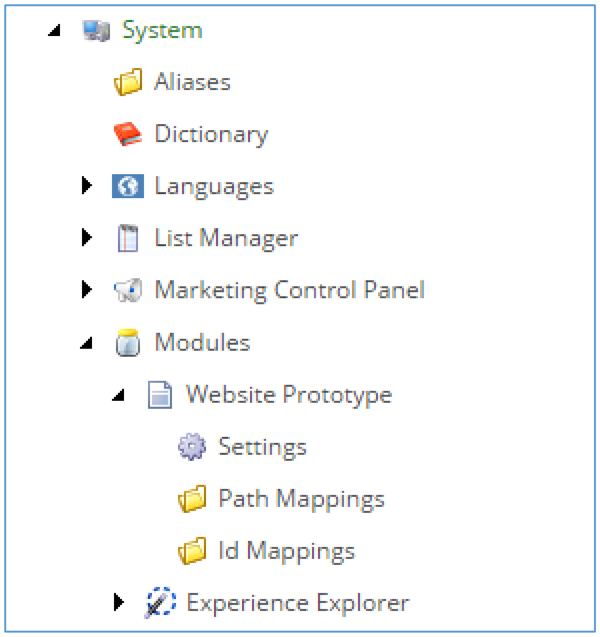
Hey! How can I get rid of the query string?

You can change the module settings in that case as described below:

14. Go to sitecore item “/sitecore/system/Modules/Website Prototype/Settings”.
15. In Enable section select the checkbox for “Enable for All Websites” field.
    Note: Please see Settings section below to understand the settings and their usage.
16. Save item and publish.
17. Open the page in browser and do not use any query string value this time.
18. You will see the prototype content of the html file.

Settings

This module has some settings and you can open the following item in sitecore to see the settings:
/sitecore/system/Modules/Website Prototype/Settings



The settings for this module contains following sections:

Enable Section

The section defines the settings to enable the prototype content rendering implicitly for several areas. Here implicitly means that when you open the page than it will always be prototype content and you do not need to add the query string value in the page url. This is specifically useful for giving demos to managers, stakeholders and client. There are several options to control the implicit behavior for different items:

1. **Enable for All Websites:**
   If true this means that sitecore will always render the prototype content of the pages for all websites.

2. **Enable for Websites:**
   This facilitates to limit the implicit behavior for the specific websites only. You have to select the website home page items to make this work.

3. **Enable for Paths:**
   This facilitates to limit the implicit behavior for the specific sections for one or more websites. You have to select the specific website page or folder to show prototype content for children of item or folder.

4. **Enable for Pages:**
   This further restrict the implicit behavior for specific pages only. You have to select the sitecore page items.

Mappings

This section is useful only if you are creating separate prototype components for the actual website components and you are not defining the html file for the component.

Id Mappings

This section is to define the one to one item mapping for the presentation component. You have to follow the below Steps:

1. Create actual and prototype presentation component in sitecore.
2. Under Id Mappings folder create a new Mapping item and select the original presentation component and prototype presentation component.
3. In the module settings go to the Id Mappings section and select the defined setting.

Path Mappings

This section is to define mapping for a group of actual and prototype presentation components. The matching of item is done based on the component name and the relative path under the group. You have to follow the below Steps:

1. Create one or more actual and prototype presentation components in sitecore and group them logically.
2. Under Path Mappings folder create a new Mapping item and select the original presentation component group item and prototype presentation component group item. Here group item means the component folder in the layout section.

3. In the module settings go to the Path Mappings section and select the defined setting.