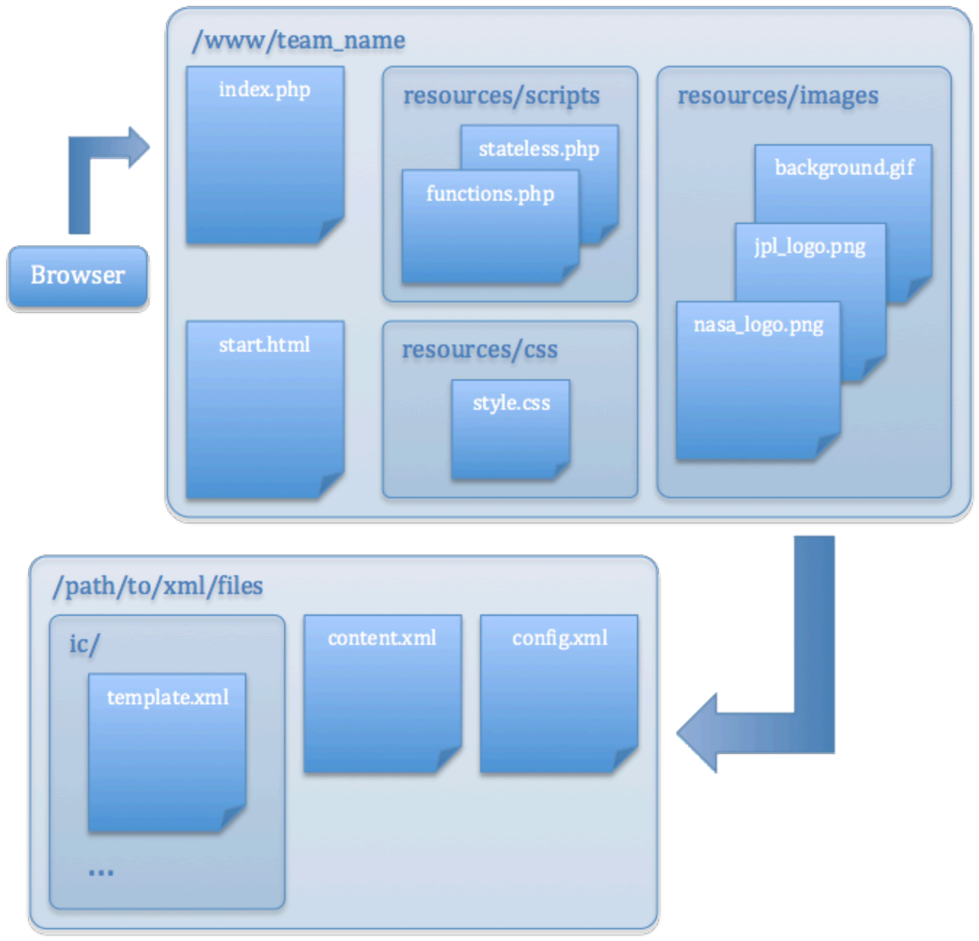
Cassini Structured Homepages (CASH)

Summer 2013
Andrew Darwin and Delvison Castillo

Overview

Built on PHP 5.2.9, the Cassini Structured Homepages Framework is designed to facilitate dynamic generation of webpages tailored specifically to the user. The main goal of the framework is to grant the site administrator the ability to hide specific content according to the current user.



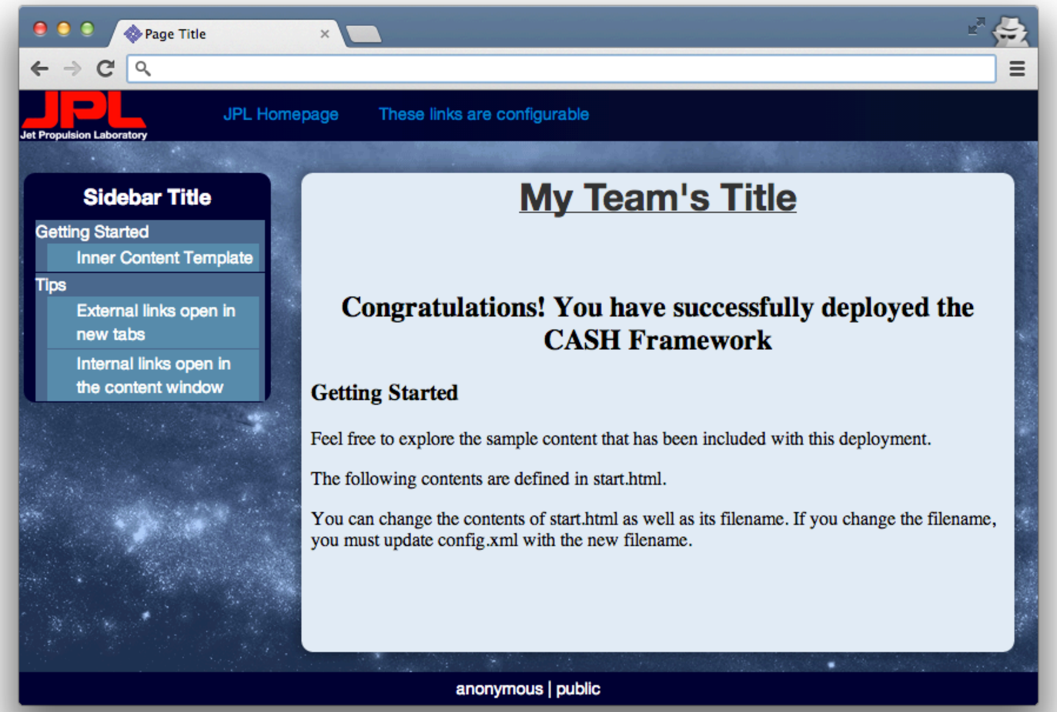
Directory Structure

~/index.php	→ Point your browser to this file to utilize the framework
~/start.html	→ This file is displayed in the center pane when you first load the framework
~/resources/scripts	→ You shouldn't have to bother with this directory.
~/resources/images/	→ This is a good place to put any images you add
~/resources/css/	→ This is a good place to add custom CSS classes.
/path/to/config.xml	→ Modify this file for security and framework configurations
/path/to/content.xml	→ Modify this file for main page content
/path/to/ic/	→ Place inner content files in here

Getting Started

Installation

1. Copy *cash.bin* and *install.sh* to the server at the desired location (e.g. /www/team_name).
2. Run *install.sh* (`sh install.sh`), to install the framework. Follow the prompts. **Note that the install script gives you the opportunity to specify where to place the xml files. While you can choose anywhere you want, it is highly recommended that you place them outside of the public domain (i.e. not /www or any subdirectory). Any xml files that exist in the public domain will be directly viewable through the web browser. This poses a major problem when the xml files contain private content such as LDAP server location and port number.**
3. If installation completed successfully, *cash.bin* and *install.sh* should no longer exist. In their place should be the web framework itself. You can now point your browser to http(s)://project.domain/team_name.


Page Title
JPL
Jet Propulsion Laboratory
JPL Homepage
These links are configurable
Sidebar Title
Getting Started
Inner Content Template
Tips
External links open in new tabs
Internal links open in the content window
anonymous | public

My Team's Title

Congratulations! You have successfully deployed the CASH Framework

Getting Started

Feel free to explore the sample content that has been included with this deployment.

The following contents are defined in start.html.

You can change the contents of start.html as well as its filename. If you change the filename, you must update config.xml with the new filename.

Upgrading from a Previous Version

1. Copy *cash.bin* and *install.sh* to the server at the location of your current CASH installation.
2. Run *install.sh* (`sh install.sh`) to install the new version of the framework over top your existing installation. Doing so will automatically

generate backups of all important files (i.e. **.xml*, *start.html*, *resources/**), unless you specify otherwise during installation.

3. If installation completed successfully, *cash.bin* and *install.sh* should no longer exist. Additionally, all xml files, *start.html*, *index.php*, and the *resources* directory should have "_backup" appended to their names, unless you specified otherwise during installation.
4. At this point, you essentially have a fresh installation of the CASH framework. In order to reimplement the changes to the previous version of CASH, simply replace the newly created xml files with those that you have modified. **If you are upgrading from version 1.0, you will need to manually transfer your modifications from *config.xml* to the new installation. Since version 1.0, the structure of *config.xml* has changed. Replacing the new *config.xml* with your old copy will render CASH unable to parse *config.xml*.**

Usage

The first time you point your browser to http(s)://project.domain/*MyTeamName*, you'll notice there's not much there. Have a look around at the minimal content that is shipped with the deployed framework. You should find it fairly self-documenting.

Configuration

The configuration file, *config.xml*, is your home for all configuration changes. **Note that the 'Access Control Lists' and 'Restricted Directories' sections of config.xml allow you to specify access control lists (ACLs) for specific directories. There is a clear distinction to be made between these ACLs and Unix ACLs. Unix ACLs apply security to files and directories. The ACLs defined in config.xml only affect CASH and what content (e.g. hyperlinks) it shows to users. If an unprivileged user were to come across a link to a file he or she wasn't supposed to see, there would be nothing preventing that user from viewing the contents of the linked file, unless Unix ACLs have been specified to restrict such access.**

Access Control Lists

Access Control Lists are declared with the following xml snippet.

```
<acl>
   <name>secret</name>
   <group>group_a</group>
   <group>group_b</group>
</acl>
```

The value of the `<name>` tag is arbitrary. We have called it *secret* here purely for example purposes. Note also that you can have as many `<group>` tags as needed. These ACL declarations don't accomplish much by themselves. They must be paired with restricted directories.

Restricted Directories

Restricted directories are declared with the following xml snippet:

```
<restricted_directory>
   <path>/www/team-name/secret_folder</path>
   <acl_name>secret</acl_name>
</restricted_directory>
```

The `<path>` tag refers to a physical directory on the server. The `<acl_name>` tag maps the path with the ACL named (in this case) “secret”. Declaring restricted directories in this way defines which groups can view content stored in a particular directory.
If you have multiple restricted directories, each one must be declared with a separate `<restricted_directory>` tag.

Start Page

The start page is referenced from the following xml tag:

```
<start_page>start.html</start_page>
```

Main Content

The main content file is referenced from the following xml tag:

```
<main_content>content.xml</main_content>
```

Default Security

The default security requirement is declared with the following xml tag:

```
<default_security>secret</default_security>
```

The value of `<default_security>` should be one of your ACL names.

Domain Name

The domain name of your project is specified with the following xml tag:

```
<domain_name>project.domain.name</domain_name>
```

LDAP Configuration

LDAP connection information is specified with the following xml snippet:

```
<ldap_config>
   <address>ldap.domain.address</address>
   <port>123</port>
   <organizational_unit1>Organizational Unit 1</organizational_unit1>
   <organizational_unit2>Organizational Unit 2</organizational_unit2>
   <organization>Some Organization</organization>
   <country_naming>US</country_naming>
   <attribute>uniqueMember</attribute>
   <use_ldap>false</use_ldap>
   <default_security_in_ldap_absence>
      public
   </default_security_in_ldap_absence>
```

Adding Content

Content for the main page should be added to *content.xml*

Defining Colors

CSS colors have been separated from the rest of the built in CSS styles and placed in the header of *index.php*. Simply modifying the hex color declarations in this file should help you achieve the color scheme you want.
For textual content, you have the option of attaching a CSS class, or embedding CSS.

Defining Styles

Styles can be defined either by referencing a CSS class with the "css_class" attribute, or by embedding the desired CSS code with "css_embed".

Hyperlinks

Hyperlinks are typically specified via the 'url' attribute, like so:

```
<li url=https://cassini.jpl.nasa.gov>
   Cassini Portal
</li>
```

You can have multiple links per line by using the <link> tag. An example of this is as follows:

```
<p>Go to
   <link url="http://www.github.com">this page</link>
   to view our
   <link url="https://en.wikipedia.org/wiki/code">
   code
   </link>
</p>
```

Lists

Lists are declared just like they are in html, using <ul> and <li> tags. The following xml snippet shows the syntax for a nested list

```
<ul>
  <li>Item 1</li>
  <li>Item 2
    <ul>
      <li>Item 2.a</li>
      <li>Item 2.b</li>
    </ul>
  </li>
  <li>Item 3</li>
</ul>
```

XML Tag Index

Supported Tags	Description
Structural	
<content>	Distinguishes the content section of xml files within the 'ic' directory.
<document>	Used as the root element of each xml file. This is required by the xml specification.
<header>	Distinguishes the header area. Any content in between these two tags is considered part of the header area on the webpage.
<navbar>	Distinguishes the navbar area. Any content in between these two tags is considered part of the navbar area. (By default, this will be the navigation bar directly on top of the webpage).
<navigation>	Distinguishes the navigation area. Any content in between these two tags is considered part of the navigation area on the webpage. (By default, this will be the sidebar on the left).
Content	
<column>	This tag designates a column of content. It is used to set up multiple columns.
<embed>	This tag allows you to embed standard XHTML code.
<h1>, <h2>, … <h6>	These are just like the html heading tags
<img />	Declares an image. Must be used with the "src" attribute
<li>	Declares a list item. Note that list items must be children of unordered lists (<ul>).
<link>	Indicates a hyperlink
<p>	Indicates an html paragraph element
<sidebar_title>	This xml tag defines the title text for the side navigation bar
<ul>	Defines an unordered list
Tables	
<cell>	Content within this tag will be translated as a column belonging a parent table tag. (Translates to <td></td> in HTML).
<col_title>	Content within this tag will be translated as a column title belonging a parent table tag. (Translates to <th></th> in HTML).
<row>	Content within this tag will be translated as a row belonging a parent table tag. (Translates to <tr></tr> in HTML).
<table>	Content within this tag will be translated to a table.

XML Attribute Index

Attribute	Description	Example Syntax
css_class	Used for specifying a CSS class that will be applied to the contents within the tag that uses it.	<p **css_class="myclass"**></p>
css_embed	Used for supplying embedded CSS for a given xml element	<h1 **css_embed="color:blue"**></h1>
heading	Specifies an html heading (h1, h2, h3, etc.)	<sidebar_title **heading="h3"**>Title</sidebar_title>
height	Used to specify the height of an image.	<img src="image.png" **height="50px"** />
security	Used for defining the security level of the contents within the tag that uses it.	<li **security="SomeACLName"**></li>
src	Used for defining the file source for an image.	<img **src="image.png"** height="50px" />
style	Used as a quick means of applying decoration to text located between the tag that uses it. Values for the style tag can be bold, italic or underlined.	<h2 **style="bold, italic"**></h2>
target	Used to force hyperlinks to open in either a "new_tab" or the "same_tab"	<li url="http://jpl.nasa.gov" **target="new_tab"**></li>
url	Used for specifying what URL a link will point to.	<li **url="http://jpl.nasa.gov"** target="same_tab"></li>
width	Used to specify the width of an image.	<img src="image.png" **width="150px"** />

XML Tag to Attribute Mapping

*Notes: 1. None of the structural xml tags use attributes.

Tag/Attribute	css_class	css_embed	heading	height	security	src	style	target	url	width
Content										
<column>					✔					
<embed>					✔					
<h#>	✔	✔			✔		✔	✔	✔	
<img />	✔	✔		✔	✔	✔		✔	✔	✔
<li>	✔	✔			✔		✔	✔	✔	
<link>					✔		✔	✔	✔	
<p>	✔	✔			✔		✔	✔	✔	
<sidebar_title>			✔				✔			
<ul>										
Tables										
<cell>	✔	✔			✔		✔	✔	✔	
<col_title>	✔	✔			✔		✔			
<row>					✔					
<table>					✔					

Power Users

--**To add JavaScript functions** → Write JavaScript functions into *resources/scripts/js_functions.js* or add your own JavaScript files into *resources/scripts/*. All JavaScript files within *resources/css/* will automatically be included in the header your *index.php* webpage.
--**To add CSS classes** → Write CSS classes into *resources/css/style.css* or add your own style sheets into *resources/css/*. All CSS files with *resources/css/* will automatically be included into the header of your *index.php* webpage.

External Libraries

- JQuery v1.9.1