

Tapestry5ImageComponent

The old "Img" component required that you specified a domain like this: `src="context:path/image.gif"`. For template previewability it's better to separate the domain out into another parameter and default it to context. Here's the result, which I've renamed "Image".

Example use:

```
<!-- previewable image that is still an asset (localized etc) at runtime -->

```

Source:

(Modified from T5.0.4. Put it into yourapp.components package)

```
// Copyright 2006, 2007 The Apache Software Foundation
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

package yourapp.components;

import org.apache.tapestry.Asset;
import org.apache.tapestry.ComponentResources;
import org.apache.tapestry.MarkupWriter;
import org.apache.tapestry.PageRenderSupport;
import org.apache.tapestry.annotations.AfterRender;
import org.apache.tapestry.annotations.BeforeRenderBody;
import org.apache.tapestry.annotations.BeginRender;
import org.apache.tapestry.annotations.Environmental;
import org.apache.tapestry.annotations.Inject;
import org.apache.tapestry.annotations.Parameter;
import org.apache.tapestry.annotations.SupportsInformalParameters;
import org.apache.tapestry.services.AssetSource;

/**
 * Renders an HTML img element using a supplied image source path. The path domain defaults to
 * context but any other tapestry domain can be specified, e.g. "classpath".
 * <p>
 * Renders an id attribute (a {@link PageRenderSupport#allocateClientId(String) unique} version
 * of the component's id) as well as informal parameters.
 */
@SupportsInformalParameters
public class Image
{
    @Environmental
    private PageRenderSupport _support;

    @Inject
    private ComponentResources _resources;

    /**
     * The image asset to render.
     */
    @Parameter(required = true, defaultPrefix = "literal")
    private String _src;

    @Parameter(required = false, defaultPrefix = "literal", value = "context")
    private String _domain;
```

```
@Inject
private AssetSource assetSource;

@BeginRender
void begin(MarkupWriter writer)
{
    String clientId = _support.allocateClientId(_resources.getId());

    Asset image = assetSource.findAsset(null, _domain + ":" + _src, null);
    writer.element("img", "src", image.toClientURL(), "id", clientId);

    _resources.renderInformalParameters(writer);
}

@BeforeRenderBody
boolean beforeRenderBody()
{
    return false;
}

@AfterRender
void after(MarkupWriter writer)
{
    writer.end();
}
}
```