

Tapestry5AnyComponent

Note: This has now been enhanced to allow replacement of an element's body. Please see the examples.

"Any" does nothing (by default) but still evaluates attributes using binding prefixes (default is literal) and can use mixins etc. The results can often be duplicated by expansions, but Any is sometimes the only solution for browser-previewable/editable HTML, or simply less intrusive - see the examples.

Example use: (Please add more examples if you use Any)

```
<!-- a static class for preview; a dynamic class at runtime -->
<span t:type="Any" t:class="prop:someMethodToGetTheClass" class="previewDisplayClass">${message:someText}</span>

<!-- *NEW* demo body for preview; dynamic body at runtime -->
<h3 t:type="Any" t:body="item.itemName" class="itemClass">Example Item</h3>

<!-- or if there are no informals like class-->
<t:any t:body="item.itemName">Example Item</t:any>

<!-- using a mixin -->
<div t:type="Any" t:mixins="myLayout">My layout</div>
```

Source:

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

```
// Copyright 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.ClientElement;
import org.apache.tapestry.ComponentResources;
import org.apache.tapestry.MarkupWriter;
import org.apache.tapestry.PageRenderSupport;
import org.apache.tapestry.annotations.Environmental;
import org.apache.tapestry.annotations.Inject;
import org.apache.tapestry.annotations.Parameter;
import org.apache.tapestry.annotations.SupportsInformalParameters;

/**
 * The Any component is a swiss-army knife that emulates any arbitrary element. Renders an
 * element tag including an id attribute and any informal parameters. The id is provided by
 * {@link PageRenderSupport#allocateClientId(String)} (so it will be unique on the client side)
 * and is available after the component renders as {@link #getClientId()}. The Any component
 * has no template but does render its body (unless the body attribute is supplied).
 */
@SupportsInformalParameters
public class Any implements ClientElement
{
    /**
     * The element to be rendered by the component. Normally, this matches the element from the
     * template, but this can be overridden if necessary.
     */
    @Parameter(value = "prop:componentResources.elementName", defaultPrefix = "literal")
    private String _element;

    /**
```

```

    * The base value for the id attribute. If non-null, then a <em>unique</em> value is
    * obtained, and the <code>id</code> attribute is added to the element. The actual unique
    * id is available via the {@link #getClientId() clientId property}.
    */
@Parameter(defaultPrefix = "literal")
private String _id;

/**
 * Optional body attribute which will replace the element body from the template.
 */
@Parameter
private String _body;

@Inject
private ComponentResources _resources;

@Environmental
private PageRenderSupport _pageRenderSupport;

private String _clientId;

/**
 * Starts the element, and renders the id attribute (if the id parameter is non-null).
 */
void beginRender(MarkupWriter writer)
{
    if (_id != null)
    {
        _clientId = _pageRenderSupport.allocateClientId(_id);
    }

    if (_element != null)
    {
        writer.element(_element);

        if (_id != null)
        {
            writer.attributes("id", _clientId);
        }

        _resources.renderInformalParameters(writer);
    }
}

boolean beforeRenderBody()
{
    return (!_resources.isBound("body"));
}

/** Ends the element. */
void afterRender(MarkupWriter writer)
{
    if (_body != null)
    {
        /** Replace the body. */
        writer.writeRaw(_body);
    }

    writer.end();
}

/**
 * The page-unique id, rendered out as the <code>id</code> attribute of the tag. This will
 * be null if the component's id parameter is unbound or otherwise null.
 */
public String getClientId()
{
    return _clientId;
}
}

```

