

# fluid.render

 This functionality is [Sneak Peek](#) status. This means that the **APIs may change**. We welcome your feedback, ideas, and code, but please use caution if you use this new functionality.

 This page is still incomplete.

## fluid.render()

A simple driver for single node self-templating. Parses the source into a template structure, renders it using the supplied component tree and options, replaces the markup in the target node with the rendered markup, and finally performs any required data binding.

```
fluid.render(source, target, tree, options);
```

**File name:** `Renderer.js`

## Parameters

<b>source</b>	(Object or String) Either a structure <code>{node: node, armouring: armourstyle}</code> or a string holding a literal template. See the <a href="#">source</a> section below for more information.
<b>target</b>	(Node) The node to receive the rendered markup
<b>tree</b>	(Object) The component tree to be rendered
<b>options</b>	(Object) An options structure to configure the rendering and binding process. See the <a href="#">options</a> section below for more information.

## Return Value

<b>Object</b>	A templates structure, suitable for a further call to <a href="#">fluid.reRender</a> or <a href="#">fluid.renderTemplates</a>
---------------	---

## See Also

- [fluid.initRendererComponent](#)

## Notes

### The source parameter

The `source` parameter can either be a string holding a literal template or an object with the following structure:

```
{
  node: <DOM node>,
  armouring: <boolean>
}
```

The properties in the object are:

<b>node</b>	This is either a pure DOM node or a jQuery object wrapping a DOM node.
<b>armouring</b>	This boolean flag controls whether or not the node is <b>treated as ...?</b> (default: <code>false</code> )

## Options

The `options` parameter is an optional object containing key/value pairs that can configure how the Renderer works. Supported options are:

Field	Description	Type	In/Out
<code>model</code>	Perhaps the most important parameter, contains the "data model" to which value bindings expressed within the tree will be expressed.	free Object	Mostly In, but the supplied model may be written to later as a result of servicing user actions, especially if the parameter <code>autoBind</code> is supplied.
<code>applier</code>	a <a href="#">ChangeApplier</a> object associated with the <code>model</code>	<code>ChangeApplier</code>	In
<code>autoBind</code>	If set, any user modification of fields with <code>valuebindings</code> set will immediately be reflected in the current state of the supplied <code>model</code>	boolean	In
<code>document</code>	If set, will cause the rendered ids to be uniquified against the supplied document, rather than the current one	document	In
<code>debugMode</code>	If set, mismatches between template and component tree will be highlighted in an unreasonable garish pink colour	boolean	In
<code>idMap</code>	This map operates in conjunction with the <code>identify</code> decorator which may be attached to nodes in the component tree. Whilst rendering takes place, this map will fill up with a lookup from the supplied nickname to the finally rendered id	free Object	In/Out
<code>messageLocator</code>	Configures the lookup from (I18N) messages referenced in the component tree, to some source of a message bundle	function (key, args) ->message	In
<code>messageSource</code>	Will construct a <code>messageLocator</code> from a raw bundle specification via a call to <code>fluid.resolveMessageSource</code>	<code>MessageSource</code> structure	In
<code>renderRaw</code>	Will XMLEncode the rendered markup before insertion into the document. Can be useful for debugging	boolean	In
<code>cutpoints</code>	This is properly a directive to the parser, rather than the renderer, but the options structure is shared. This contains a list of pairs of <code>id</code> , <code>selector</code> which will be used to impute an <code>rsf:id</code> structure onto a document, by means of matching the paired selector	Array of <code>Cutpoint</code> object	In

## Example

example here

In this example, description here...