fluid.demands

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.

fluid.demands()

Registers a specification for resolving the invocation of a particular function name in a given context.

```
fluid.demands(functionName, context, demandsSpec/argsArray);
```

File name: `FluidIoC.js`

Parameters

`functionName`	(String) the name of the function demand which this specification matches.
`context`	(String or Array of Strings) the name(s) of the context(s) in which this demand specification applies. See #Contexts below for more information.
`demandsSpec/argsArray`	(Object)/(Array) the actual demands specification to be applied in the case this rule matches. If the functionName does not require to be replaced by another, it is sufficient just to list an Array of the replacement arguments. Otherwise, this argument can be a full #Demands Specification

Return Value

None	

See Also

- fluid.initDependents
- fluid.defaults
- Demands Specifications
- API

Notes

Contexts

Components or functions in general may have different requirements depending on the context in which they are operating. For example, a subcomponent might operate differently when running on a production server versus when testing locally off the file system, and differently still when operating in the context of automated tests. In a more fine-grained way, a component may behave differently when operating in a browser with different capabilities, or on behalf of a user who has expressed particular needs or preferences. The `context` parameter to `fluid.demands` specifies a name for particular context in which the supplied demands are intended to be valid.

For more detailed information about contexts, see Contexts.

Demands Specification

The third parameter to `fluid.demands()`, **spec**, is the demands specification that declares how the function invocation is to be disposed, in the case that the overall block matches. This parameter can take one of several possible forms:

`{ funcName: "functionName", args: [array of arguments] }`	The array of arguments can contain references to any value-resolved material, as may also appear in `fluid.defaults` blocks

Example

```
fluid.demands(
    "my.component",
    "my.application",
    {
        funcName: "my.componentImpl",
        args: ["{application}.model", "{application}.options.strings"],
    });
```

In this example, the component `my.component` is specifying its requirements when used in the context of `my.application`. The request is that the component be instantiated using a function called `my.componentImpl`, passed the `args` array as a parameter.

When the IoC system fulfills the demands, the `model` property and the `options.strings` property of the `my.application` object will be retrieved and inserted in the `args` array before the function is called.