Decorators

The Infusion Framework contains two categories of thing going by the name of *Decorators*. This naming is inspired by, but not a direct implementation of, the Gang of Four (GoF) Decorator pattern. Whereas GoF Decorators are explicitly an Object-Oriented technique based on interface interception and forwarding, since Infusion is not directly an Object-Oriented framework, its "Decorator" concepts are necessarily lighter and less intrusive on control flow.

Both of these categories of Decorator have in common with each other, and also with the GoF Decorator concept, that they represent an "orthogonal" modification of the base component - that is, firstly, that they do not represent a modification of the original component contract. Going further, decorators in Infusion represent a greater orthogonality in that the same decorator may be attached to any (suitable) component, modifying its behaviour in the same way. In an OO framework this concept of "suitability" would be expressed by derivation from a common base class – since Fluid (and, fundamentally, Javascript) is not class-oriented, this is not the case here.

Component Decorators

The first category of decorator is decorators to Infusion Components. These may be directed to be attached, as a Subcomponent to a top-level component by virtue of satisfying a (typically very light/none) instantiation contract - they will interact with its lifecycle, modifying its behaviour at key instances typically via the Fluid Event System. A classic example of a Component decorator is the Undo Decorator. This may successfully decorate any of the existing Inline Edit family of components, as well as a wider space of similar components satisfying the same very light model contract.

Renderer Decorators

The second category of decorator is decorators to (Fluid) Renderer Components. Since Renderer Components have no behaviour and consist purely of data, the intrusion on the base contract is necessarily even lighter. Every Renderer Component has a "slot" for an array of decorators, stored at its member `decorators`. When it encounters this list during rendering, the Fluid Renderer will honour them by either modifying the markup or issuing a function call, in a manner appropriate to the decorator, and independent of the nature of the decorated component.

The complete list and function of these decorators is listed at Renderer Decorators.