

## The Four Functions of the AngularJS Directive Life Cycle

There are many options that can be configured and how those options are related to each other is important. Each directive undergoes something similar to a life cycle as AngularJS compiles and links the DOM. The directive lifecycle begins and ends within the AngularJS bootstrapping process, before the page is rendered. In a directive's life cycle, there are four distinct functions that can execute if they are defined. Each enables the developer to control and customize the directive at different points of the life cycle.

The four functions are: *compile*, *controller*, *pre-link* and *post-Link*.

The **compile** function allows the directive to manipulate the DOM before it is compiled and linked thereby allowing it to add/remove/change directives, as well as, add/remove/change other DOM elements.

The **controller** function facilitates directive communication. Sibling and child directives can request the controller of their siblings and parents to communicate information.

The **pre-link** function allows for private **\$scope** manipulation before the post-link process begins.

The **post-link** method is the primary workhorse method of the directive.

In the directive, post-compilation DOM manipulation takes place, event handlers are configured, and so are watches and other things. In the declaration of the directive, the four functions are defined like this.

```
.directive("directiveName",function ()
{
    return
    {
        controller: function()
        {
            // controller code here...
        },
        compile:
        {
            // compile code here...
            return {
                pre: function()
                {
                    // pre-link code here...
                },
                post: function()
                {
                    // post-link code here...
                }
            };
        }
    }
})
```

