WebSystique
learn together

AngularJS Custom-Directives link-function guide

Created on: December 20, 2015 | **Last updated on:** September 30, 2017 websystiqueadmin

AngularJS Directive's **link** key defines **link function** for the directive. Precisely, **using link function, we can define directive's API & functions that can then be used by directive to preform some business logic**. The link function is also responsible for **registering DOM listeners as well as updating the DOM**. It is executed after the template has been cloned.

This post is a part of AngularJS Directives Tutorial Series.

link function gets executed for each instance of directive so that each instance gets it's own business-logic without affecting others. If we need to add functionality to our instance of the directive, we can add it to the scope of the element we're working with. AngularJS passes several arguments to the link function, which looks like following:

```
link: function(scope, element, attrs) {
...
}
```

Recent Posts

- Spring Boot + AngularJS + Spring Data + JPA CRUD App Example
- Spring Boot Rest API Example
- Spring Boot WAR deployment example
- Spring Boot Introduction + hello world example
- Secure Spring REST API using OAuth2

scope : It is the scope of the element the directive is applied on. Note that scope is not injected but passed in.
element : It is the DOM element in HTML the directive is applied on.
attrs : These are the list of attributes as string available on the element in HTML.

There are other optional arguments available in link function for more advanced usage. Like:

```
link: function(scope, element, attrs, controller, transcludeFn) {
...
}
```

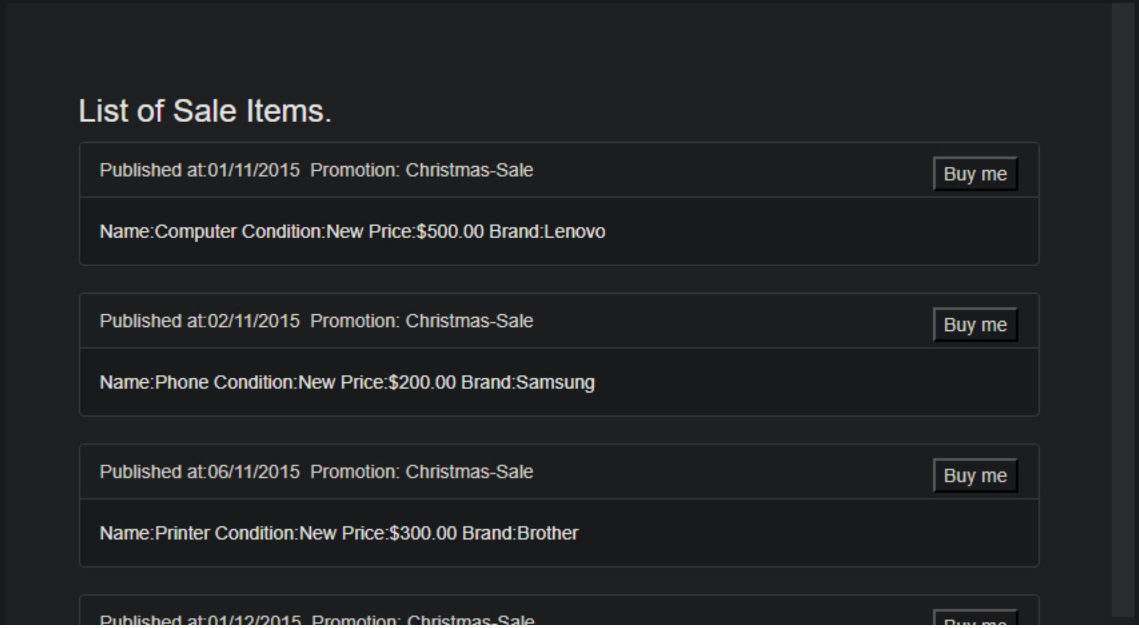
controller : It specifies the directive's required controller instance(s) or its own controller (if any). The exact value depends on the directive's require property. That controller itself is defined in directive function using `require` key.
transcludeFn : It is a transclude linking function pre-bound to the correct transclusion scope.

In this post, we will discuss the basic usages of link function. Advanced usage will be discussed in detail in post Directive's Controller & require option.

Let's see an example to learn in detail. We will build on the example from previous post.

Live Example:

This example shows how we can take advantage of **link function** to perform DOM manipulation via adding event listeners.



List of Sale Items.

Published at:01/11/2015 Promotion: Christmas-Sale

Name:Computer Condition:New Price:$500.00 Brand:Lenovo

Published at:02/11/2015 Promotion: Christmas-Sale

Name:Phone Condition:New Price:$200.00 Brand:Samsung

Published at:06/11/2015 Promotion: Christmas-Sale

Name:Printer Condition:New Price:$300.00 Brand:Brother

Published at:01/12/2015 Promotion: Christmas-Sale

Published at:01/12/2016 Promotion: Christmas-Sale Buy me

Name:Dishwasher Condition:Second-Hand Price:$250.00 Brand:WhirlPool

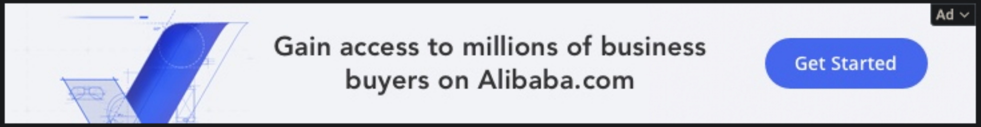
Now as soon as we click on an item, it gets replaced with a demo message[in addition to logging on console]. Simple DOM manipulation. Let's see how we do it.

```
.directive('itemWidget', [function() {
    return{
        ...
        link : function(scope, element, attrs){

                //Add event listener for 'click' event
            element.on('click', function(event) {
                    //Update DOM.
                element.html('Thanks for buying this item.');
                element.css({
                    color: 'green'
                });
          });
        }
    }
}]);
```

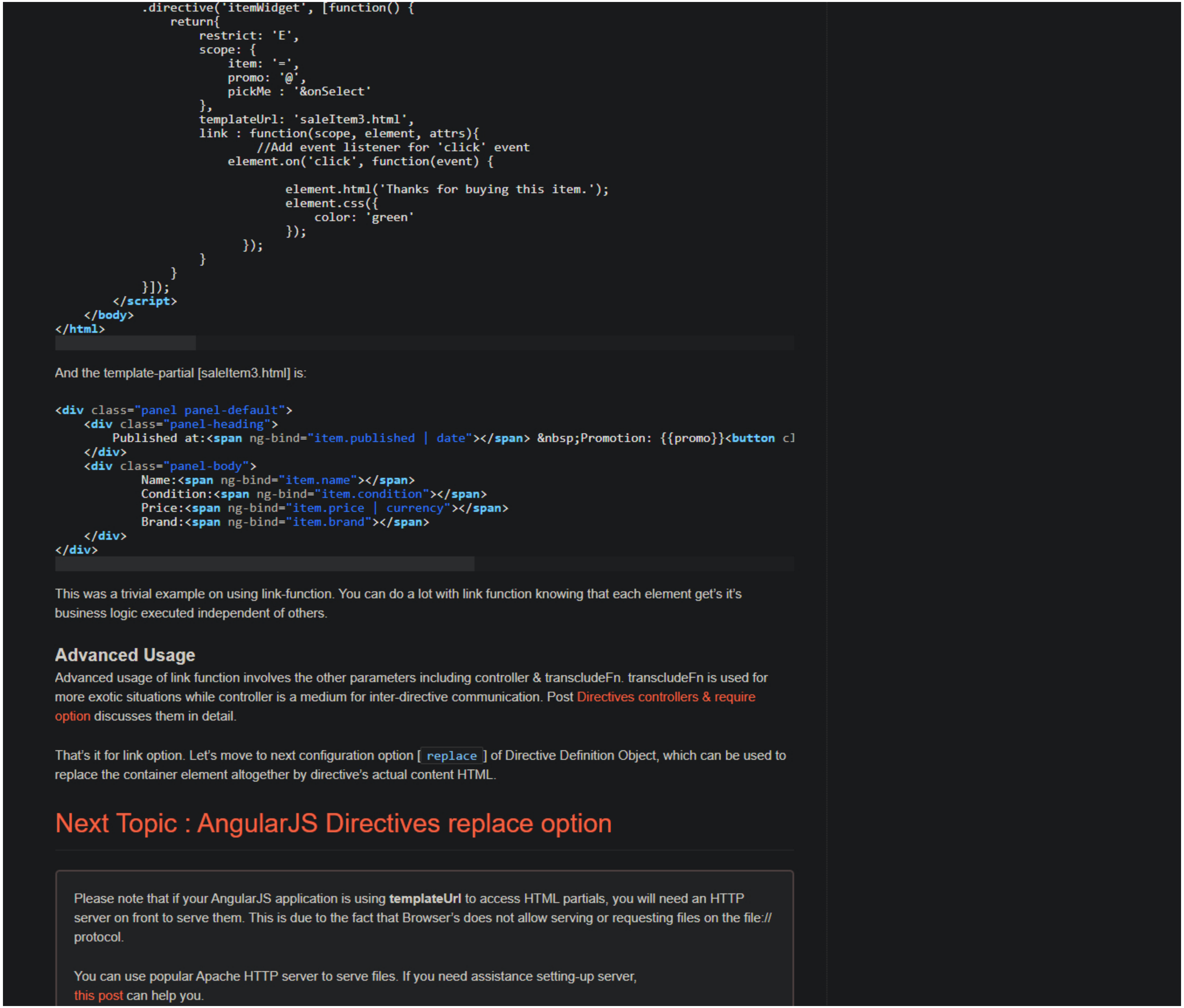
First thing to be noticed is the inclusion of **link function**. In this example, we are playing with DOM manipulation. We have kept the logic of DOM manipulation in our link function. Once an item is clicked, we apply CSS on that specific item and replace it's content by a dummy message. Interestingly, each element get's it's business logic executed independent of others.

Complete code for this example is shown below:



```
<html>
    <head>
        <title>Directive Demo</title>
        <link rel="stylesheet" href="<a class="vglnk" href="https://maxcdn.bootstrapcdn.com/bootstrap/3
    </head>
    <body class="jumbotron container" ng-app="myApp">
        <div ng-controller="AppController as ctrl">
            <h3>List of Sale Items.</h3>
            <div ng-repeat="bla in ctrl.items">
                <item-widget item="bla" promo="Christmas-Sale" on-select="ctrl.onItemSelect(selectedIte
            </div>
        </div>

        <script src="<a class="vglnk" href="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.4/angul
        </script>
        <script>
            angular.module('myApp', [])
            .controller('AppController', [function() {
                var self = this;
                self.items = [
                            {name: 'Computer', price: 500, condition:'New',brand : 'Lenovo', published:
                            {name: 'Phone', price: 200, condition:'New',brand : 'Samsung', published:'0
                            {name: 'Printer', price: 300, condition:'New',brand : 'Brother', published:
                            {name: 'Dishwasher', price: 250, condition:'Second-Hand',brand : 'WhirlPool
                            ];
                self.onItemSelect = function(name) {
                    console.log('Congrats. You have just bought a', name);
                };
            }])
```



```
            .directive('itemWidget', [function() {
                return{
                    restrict: 'E',
                    scope: {
                        item: '=',
                        promo: '@',
                        pickMe : '&onSelect'
                    },
                    templateUrl: 'saleItem3.html',
                    link : function(scope, element, attrs){
                            //Add event listener for 'click' event
                        element.on('click', function(event) {

                                element.html('Thanks for buying this item.');
                                element.css({
                                    color: 'green'
                                });
                          });
                    }
                }
            }]);
        </script>
    </body>
</html>
```

And the template-partial [saleItem3.html] is:

```
<div class="panel panel-default">
    <div class="panel-heading">
        Published at:<span ng-bind="item.published | date"></span> &nbsp;Promotion: {{promo}}<button cl
    </div>
    <div class="panel-body">
            Name:<span ng-bind="item.name"></span>
            Condition:<span ng-bind="item.condition"></span>
            Price:<span ng-bind="item.price | currency"></span>
            Brand:<span ng-bind="item.brand"></span>
    </div>
</div>
```

This was a trivial example on using link-function. You can do a lot with link function knowing that each element get's it's business logic executed independent of others.

Advanced Usage

Advanced usage of link function involves the other parameters including controller & transcludeFn. transcludeFn is used for more exotic situations while controller is a medium for inter-directive communication. Post Directives controllers & require option discusses them in detail.

That's it for link option. Let's move to next configuration option [`replace`] of Directive Definition Object, which can be used to replace the container element altogether by directive's actual content HTML.

Next Topic : AngularJS Directives replace option

Please note that if your AngularJS application is using **templateUrl** to access HTML partials, you will need an HTTP server on front to serve them. This is due to the fact that Browser's does not allow serving or requesting files on the file:// protocol.

You can use popular Apache HTTP server to serve files. If you need assistance setting-up server, this post can help you.

Download Source Code

Download Now!

References

- AngularJS Directive Official Reference Documentation
- AngularJS Official Reference Documentation

websystiqueadmin

If you like tutorials on this site, why not take a step further and connect me on Facebook , Google Plus & Twitter as well? I would love to hear your thoughts on these articles, it will help improve further our learning process.

Related Posts:

1. **AngularJS Custom-Directives controllers, require option guide**
2. **AngularJS Custom-Directives scope guide**
3. **AngularJS Custom-Directives restrict option guide**
4. **AngularJS Custom-Directives replace option guide**

angularjs. permalink.

← AngularJS Custom-Directives scope guide AngularJS Custom-Directives replace option guide →

2 Comments websystique

1 Login

♡ Recommend Tweet f Share

Sort by Best

Join the discussion…

LOG IN WITH

D f G

OR SIGN UP WITH DISQUS ?

Name

Beginner • 2 years ago • edited

what is the scope object purpose in directive and the symbols in scope object

1 ^ ˅ • Reply • Share ›

Pavani Manthena • a year ago

not very clear

^ ˅ • Reply • Share ›

Subscribe Add Disqus to your site Disqus' Privacy Policy DISQUS