

STAYING SHARP

*with
AngularJS*

Directives Level 2

\$scope Section 1

STAYING SHARP with AngularJS



NOTES

+ New Note

Thinking of Large Data

NgModel BP

Our Current Code

Every time we need to print a note, we are using the same code over and over.

notes.html

```
...  
  <div class="note-wrapper">  
    <a class="card-notes" ng-repeat="note in notes"  
      ng-href="#/notes/{{note.id}}">  
      <div class="card">  
        <h2 class="h3">{{card.header}}</h2>  
      </div>  
    </a>  
  </div>  
...
```


Extracting Into a Directive Template

notes.html

```
...  
<div class="note-wrapper">  
  <a class="card-notes" ng-repeat="note in notes"  
    ng-href="#/notes/{{note.id}}">  
  
  </a>  
</div>  
...
```

nw-card.html

```
<div class="card">  
  <h2 class="h3">{{card.header}}</h2>  
</div>
```


Creating a New nwCard Directive

notes.html

```
...  
<div class="note-wrapper">  
  <a class="card-notes" ng-repeat="note in notes" ... >  
    <nw-card></nw-card>  
  </a>  
</div>  
...
```

nw-card.js

```
angular.module("NoteWrangler")  
  .directive("nwCard", function() {  
    return {  
      restrict: "E",  
      templateUrl: "templates/directives/nw-card.html"  
    };  
  });
```


Creating a New nwCard Directive

nw-card.js

```
angular.module("NoteWrangler")
.directive("nwCard", function() {
  return {
    restrict: "E",
    templateUrl: "templates/directives/nw-card.html"
  };
});
```


Giving nwCard's Controller an Alias

nw-card.js

```
angular.module("NoteWrangler")
.directive("nwCard", function() {
  return {
    restrict: "E",
    templateUrl: "templates/directives/nw-card.html",
    controller: function(){

    },
    controllerAs: "card"
  };
});
```


Setting a Header With controllerAs Syntax

nw-card.js

```
angular.module("NoteWrangler")
.directive("nwCard", function() {
  return {
    restrict: "E",
    templateUrl: "templates/directives/nw-card.html",
    controller: function(){
      this.header = "Note Title";
    },
    controllerAs: "card"
  };
});
```

nw-card.html

```
<div class="card">
  <h2 class="h3">{{card.header}}</h2>
</div>
```

Using "this" to bind
data for the template

Note Title

Setting a Header With \$scope Syntax

nw-card.js

```
angular.module("NoteWrangler")
.directive("nwCard", function() {
  return {
    restrict: "E",
    templateUrl: "templates/directives/nw-card.html",
    controller: function($scope){
      $scope.header = "Note Title";
    },
    controllerAs: "card"
  };
});
```

nw-card.html

```
<div class="card">
  <h2 class="h3">{{card.header}}</h2>
</div>
```

Can get rid of
controllerAs and alias

Binding Header to \$scope

nw-card.js

```
angular.module("NoteWrangler")
.directive("nwCard", function() {
  return {
    restrict: "E",
    templateUrl: "templates/directives/nw-card.html",
    controller: function($scope){
      $scope.header = "Note Title";
    }
  };
});
```

nw-card.html

```
<div class="card">
  <h2 class="h3">{{header}}</h2>
</div>
```

Note Title

Both Syntaxes Attach to \$scope

Uses a card alias to reference the controller and this to reference \$scope behind the scenes.

References our controller's scope, which our directive already has access to, so no alias needed.

nw-card.js

```
controller: function($scope) {  
  this.header = "Note Title";  
},  
controllerAs: "card"
```

nw-card.js

```
$ controller: function($scope) {  
  $scope.header = "Note Title";  
}
```

nw-card.html

```
<div class="card">  
  <h2 class="h3">{{card.header}}</h2>  
</div>
```

nw-card.html

```
<div class="card">  
  <h2 class="h3">{{header}}</h2>  
</div>
```


Either Syntax Will Work

nw-card.js

```
controller: function($scope) {  
  this.header = "Note Title";  
},  
controllerAs: "card"
```

nw-card.js

```
controller: function($scope) {  
  $scope.header = "Note Title";  
}
```

nw-card.html

```
<div class="card">  
  <h2 class="h3">{{card.header}}</h2>  
</div>
```

nw-card.html

```
<div class="card">  
  <h2 class="h3">{{header}}</h2>  
</div>
```

Same result

STAYING SHARP with AngularJS