

2.2 | Scope-a-dope 250 PTS

When using the `controllerAs` syntax, the controller's context (`this`) attaches things to the current `scope` behind the scenes.

☒ True

☐ False

2.3 | Inline Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 1/5

1 2 3 4 5

Inject the `$scope` service into the controller.

```
templateUrl: 'templates/directives/nw-card.html',
controller: function($scope) {
  this.header = 'Note Title';
  this.description = 'A lovely note description.';
},
```

2.3 | Inline Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 2/5

1 2 3 4 5

Delete the unnecessary alias for our controller.

```
templateUrl: 'templates/directives/nw-card.html',
controller: function($scope) {
  $scope.header = 'Note Title';
  this.description = 'A lovely note description.';
},
```

2.3 | Inline Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 3/5

1 2 3 4 5

Set the `header` property on the controller's `$scope` instead of `this`.

Note: We do not want to modify the string.

```
templateUrl: 'templates/directives/nw-card.html',
controller: function($scope) {
  $scope.header = 'Note Title';
  this.description = 'A lovely note description.';
},
```

2.3 | Inline Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 4/5

1 2 3 4 5

Set `description` property on the controller's `$scope` instead of `this`.

Note: We do not want to modify the string.

```
templateUrl: 'templates/directives/nw-card.html',
controller: function($scope) {
  $scope.header = 'Note Title';
  $scope.description = 'A lovely note description.';
},
```

2.3 | Inline Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 5/5

1 2 3 4 5

Display the `header` and `description` on the `nw-card.html` page.

js/directives/nw-card.js

templates/directives/nw-card.html



```
1 <div class="card" title="{{header}}">
2   <h2 class="h3">{{header}}</h2>
3   <p>{{description}}</p>
4 </div>
5
```

2.4 | Outside Controller & \$scope 250 PTS

Now that you are familiar with `$scope`, let's practice using it with an outside controller.

Help Me

Task 1/3

1 2 3

Attach `notes` to `$scope` instead of the controller (`this`).

```
1 angular.module('NoteWrangler')
2 .controller('NotesIndexController', ['$http', '$scope', function($http, $scope) {
3   var controller = $scope;
4
5   $http({method: 'GET', url: '/notes'}).success(function(data) {
6     controller.notes = data;
7   });
8 }]);
9 |
```

2.4 | Outside Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 2/3

1 2 3

Remove the `controllerAs` controller alias from `routes.js`.

2.4 | Outside Controller & \$scope 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 3/3

1 2 3

Update the `notes/index.html` template to work with `$scope`.

```
</div>

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

2.5 = slides

2.6 | Default Scope 250 PTS

By default, directives use their parents' scope.

☒ True

☐ False

2.7 | Setting on Scope 250 PTS

When setting `$scope.title` inside a directive using the default scope, we are actually setting a variable named `title` on the parent `$scope`.

```
angular.module('NoteWrangler', [])
.directive("nwCard", function() {
  return {
    restrict: "E",
    templateUrl: "templates/directives/nw-card.html",
    controller: function($scope) {
      $scope.title = 'Space Llamas';
    }
  };
});
```

☒ True

☐ False

2.8 | A Flexible Card Directive 250 PTS

In order to make the `nw-card` directive reusable, it needs to accept a few attributes.

Help Me

Task 1/2

1 2

Give the `nw-card` directive an `isolate` scope, and enable it to accept a `header` and `description` attribute with two-way bindings.

Rewatch Video

Solution (-10 pts)

```
return {
  restrict: 'E',
  templateUrl: 'templates/directives/nw-card.html',
  scope: {
    header: "=",
    description: "="
  }
}
```

2.8 | A Flexible Card Directive 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 2/2

1 2

Now make `nw-card.html` work for both Notes and Users by accessing the `header` and `description` directly from the scope.

nw-card.js

nw-card.html

notes/index.html


```
1 <div class="card" title="{{header}}">
2   <h2 class="h3">{{header}}</h2>
3   <p>{{description}}</p>
4 </div>
```

2.9 | Making Isolation Work for You 250 PTS

Now that the `nw-card` directive has an `isolate` scope, the note cards no longer have access to the variables in their expressions. Let's fix that and finish making this Directive reusable for both Notes and Users.

Help Me

Task 1/2

1 2

Add attributes for `header` and `description` for the `nw-card` directive in the `notes/index.html` template.

The note `title` should be used for the `header`, and the note `description` should be used for the `description`.

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

2.9 | Making Isolation Work for You 250 PTS

We have already separated out the duplicate note code into `nw-card.html`. Clean up the template using `$scope` syntax instead of `controllerAs`.

Help Me

Task 2/2

1 2

Add attributes for `header` and `description` for the `nw-card` directive in the `users/index.html` template.

The user `name` should be used for the `header`, and the user `bio` should be used for the `description`.

```
<div class="users-wrapper">
  <h1>Users</h1>
```

```
<div class="users-wrapper">
  <a class="card-users" ng-repeat="user in users">
    <nw-card header="user.name" description="user.bio"></nw-card>
  </a>
</div>
```

```
</div>
```