

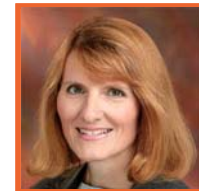
Building the First Page

Deborah Kurata

<http://blogs.msmvps.com/deborahk/>

@DeborahKurata

deborahk@insteptech.com



pluralsight 
hardcore dev and IT training

Module Objectives

Turn business requirements into a View, Model and Controller

Organize the Project Structure

Build the View

Define the Module

Build the Controller

Define the Model

Implement the Methods

Style the View with Bootstrap

As a product manager, I can ...

Browse the list of products


See the product images

Select a product to view details

Select a product to edit details

Product List View

Product List

<div>Hide Image</div>	Product	Code	Available	Price	
	Leaf Rake	GDN-0011	March 19, 2009	\$19.95	Edit
	Garden Cart	GDN-0023	March 18, 2010	\$32.99	Edit
	Hammer	TBX-0048	May 21, 2013	\$8.99	Edit
	Saw	TBX-0022	May 15, 2009	\$11.55	Edit
	Video Game Controller	GMG-0042	October 15, 2002	\$35.95	Edit

Product Model (JSON)

```
[
  {
    "productId": 1,
    "productName": "Leaf Rake",
    "productCode": "GDN-0011",
    "releaseDate": "March 19, 2009",
    "description": "Leaf rake with 48-inch handle.",
    "cost": 9.00,
    "price": 19.95,
    "category": "garden",
    "tags": [ "leaf", "tool" ],
    "imageUrl": "http://opencart.org/image/
                300px/svg_to_png/26215/Anonymous_Leaf_Rake.png"
  },
  {
    ...
  }
]
```

Product List Controller

Define the Model

- Retrieve the Products

Implement Methods

- Hide/Show Images

Angular Application

ProductListView

```
<body ng-app>  
  <div ng-controller=  
    "ProductListCtrl">
```

```
</div>  
</body>
```

ProductListCtrl

Model

Methods

Organizing the Project Structure

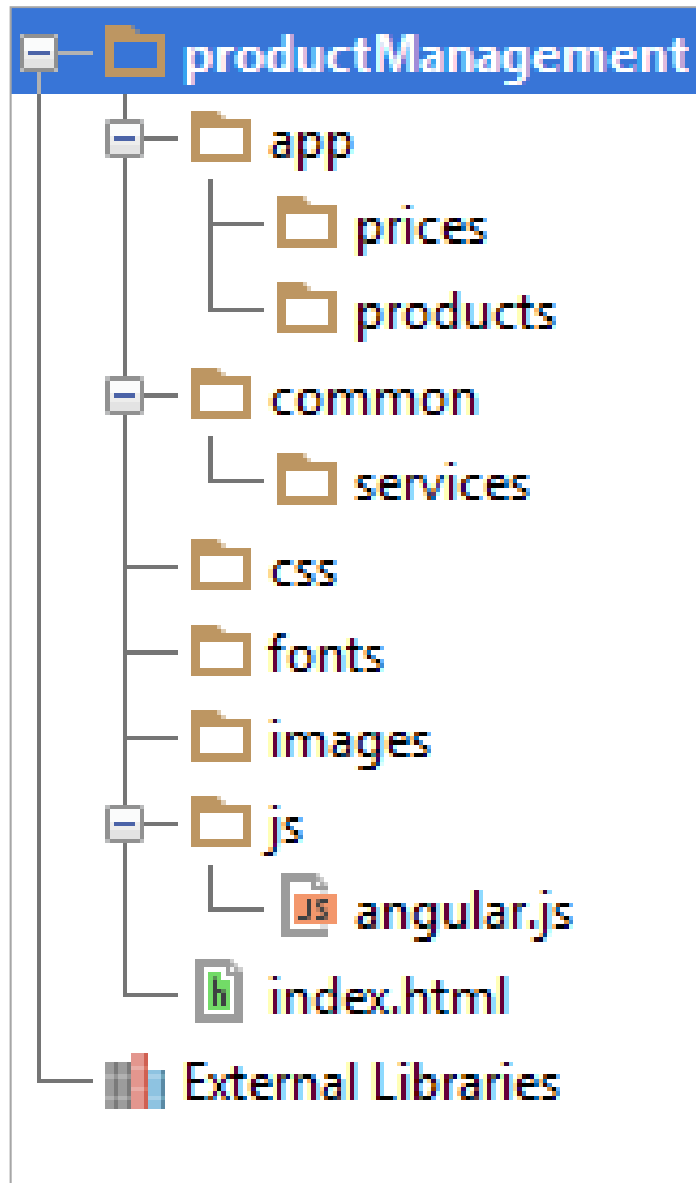
**Everyone knows
where everything
goes**

**Everyone knows
where everything
is**

**The result is
extensible**

**Pluralsight Course:
“AngularJS
Patterns: Clean
Code”**

Product Management Application Structure







View

```
<table class="table">
  <thead>
    <tr>
      <td style="vertical-align:bottom">Image</td>
      <td style="vertical-align:bottom">Product</td>
      <td style="vertical-align:bottom">Code</td>
      <td style="vertical-align:bottom">Available</td>
      <td style="vertical-align:bottom">Price</td>
    </tr>
  </thead>
  <tbody>
    <tr ng-repeat="product in vm.products">
      <td></td>
      <td>{{ product.productName }}</td>
      <td>{{ product.productCode }}</td>
      <td>{{ product.releaseDate | date }}</td>
      <td>{{ product.price | currency }}</td>
      <td><a class="btn btn-primary">Edit</a></td>
    </tr>
  </tbody>
</table>
```

Product List

[Hide Image](#)

	Product	Code	Available	Price	
	Garden Cart	GDN-0023	Mar 18, 2010	\$32.99	Edit
	Hand Cultivator	GDN-0035	Mar 18, 2012	\$9.99	Edit
	Hand Trowel	GDN-0067	Mar 22, 2013	\$6.99	Edit
	Leaf Rake	GDN-0011	Mar 19, 2009	\$19.95	Edit

Template

- HTML that defines visual elements
- And directives

View

- Load and rendered template
- Also called the “live template”

Module

**Defines an
Angular
application**

**Tracks all of the
application code**

**Tracks all of the
dependencies for
the application**

**Keeps the
application
modularized**

Angular Module Method

Setter Method

```
angular.module("productManagement", []);
```

Getter Method

```
angular.module("productManagement");
```

The IIFE JavaScript Pattern

- Immediately-Invoked Function Expression (IIFE)
 - AKA: Self-executing anonymous function
- JavaScript var declarations have **global** scope
- JavaScript variables and functions defined within a function have **local** scope

```
(function () {  
    // Define the main module  
    var app = angular.module("productManagement", []);  
  
})();
```

Controller Code

Defines the Model

- Products

Implements actions with Methods

- Hide/Show images

Controller

- **Controller Function**
 - Global namespace
 - Or registered with a module
- **ng-controller Directive**

productManagement

ProductListView

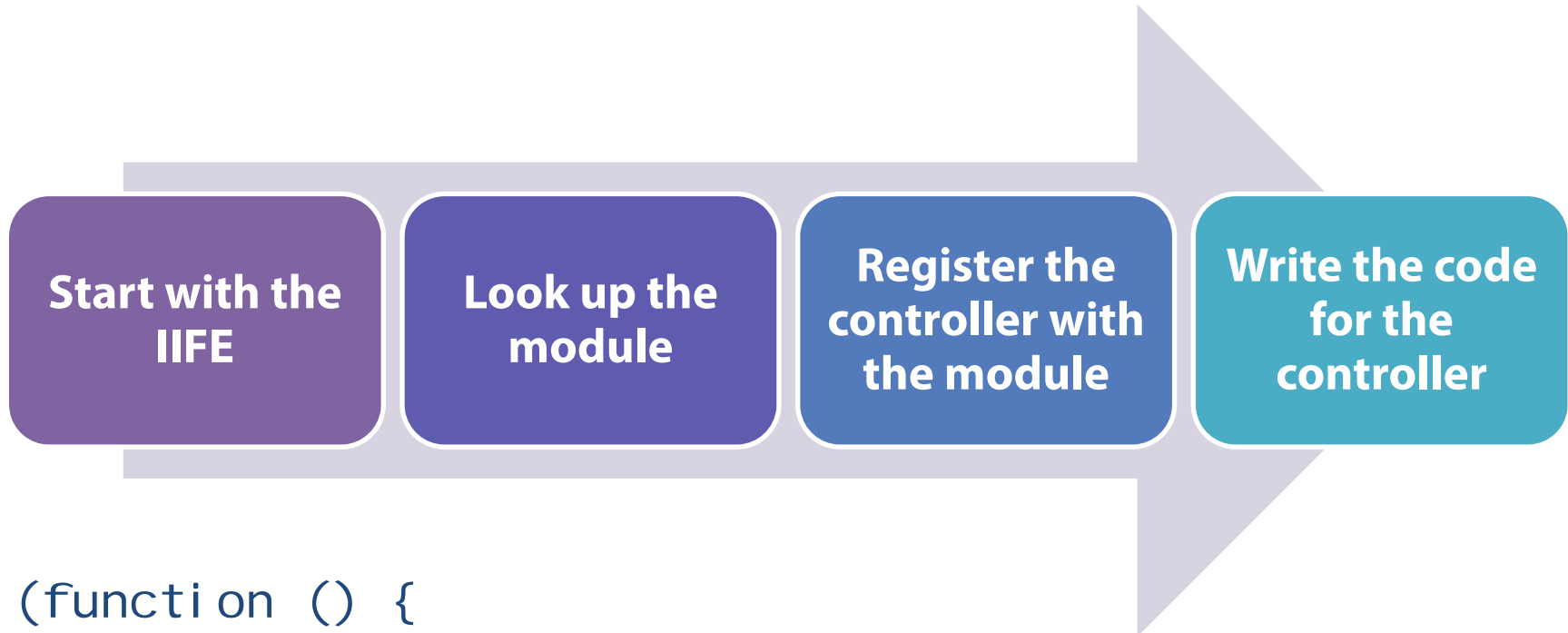
```
<body ng-app>  
  <div ng-controller=  
    "ProductListCtrl">  
  
  </div>  
</body>
```

ProductListCtrl

Model

Methods

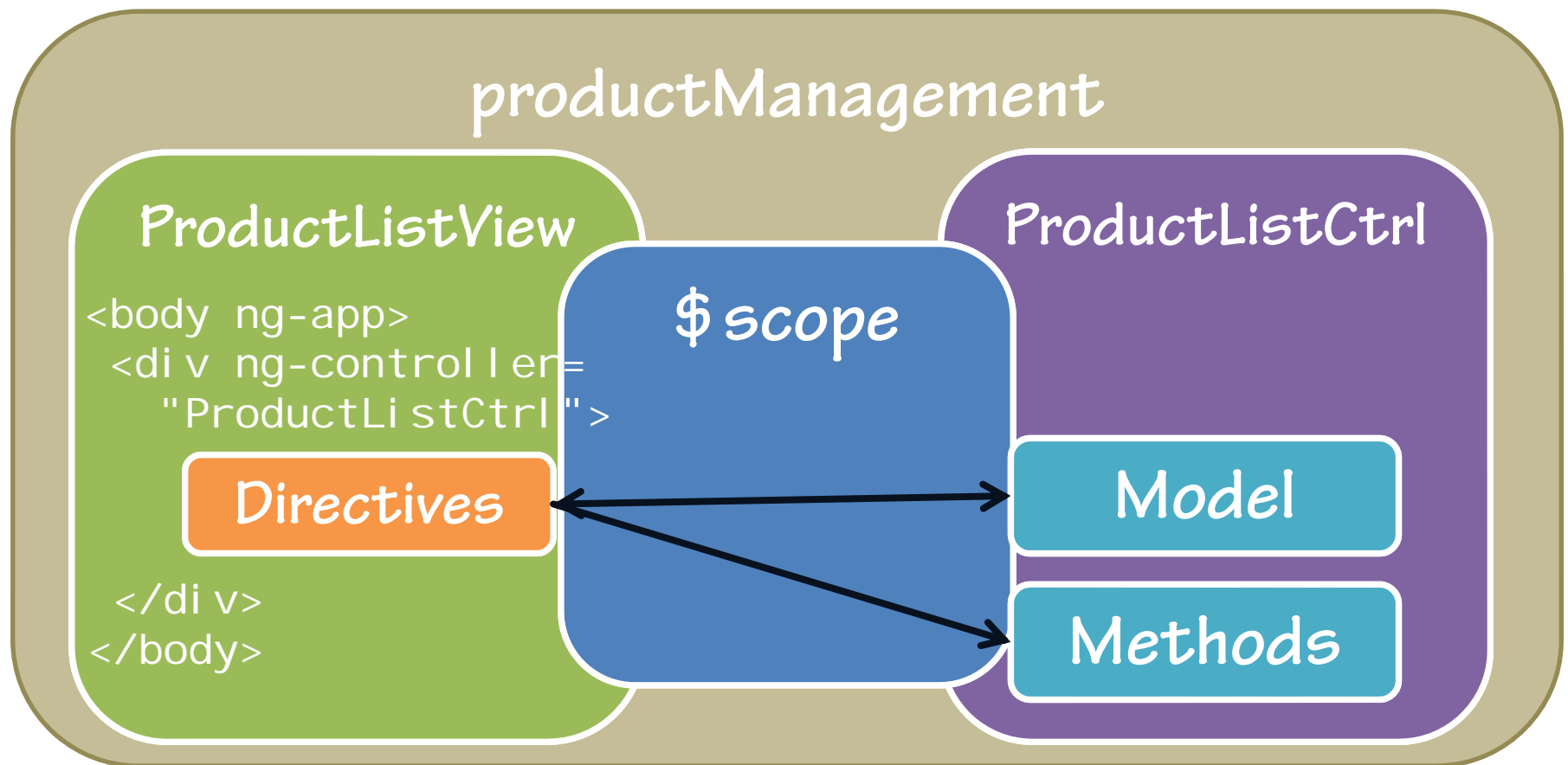
Steps for Building a Controller



```
(function () {  
    angular  
        .module("productManagement")  
        .controller("ProductListCtrl",  
            ["$scope",  
            function($scope) {  
                // Code here  
            }]);  
})();
```


\$scope

- Communication mechanism between the View and Controller
- Angular injects \$scope into the controller
- Controller augments the scope



“Controller As”

Angular 1.2 simplified \$scope handling

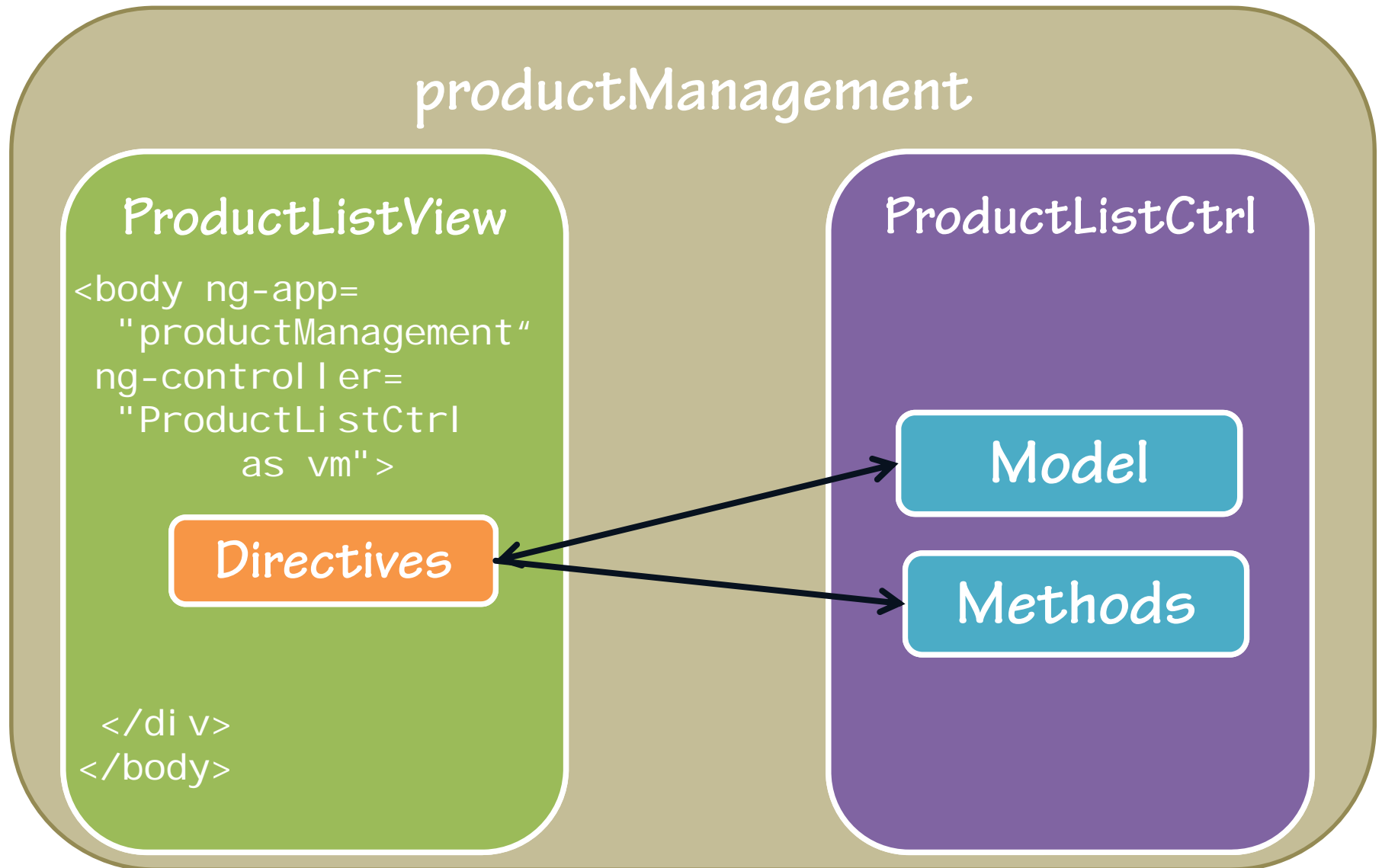
\$scope is not required as a parameter

Model and methods are defined on the controller itself

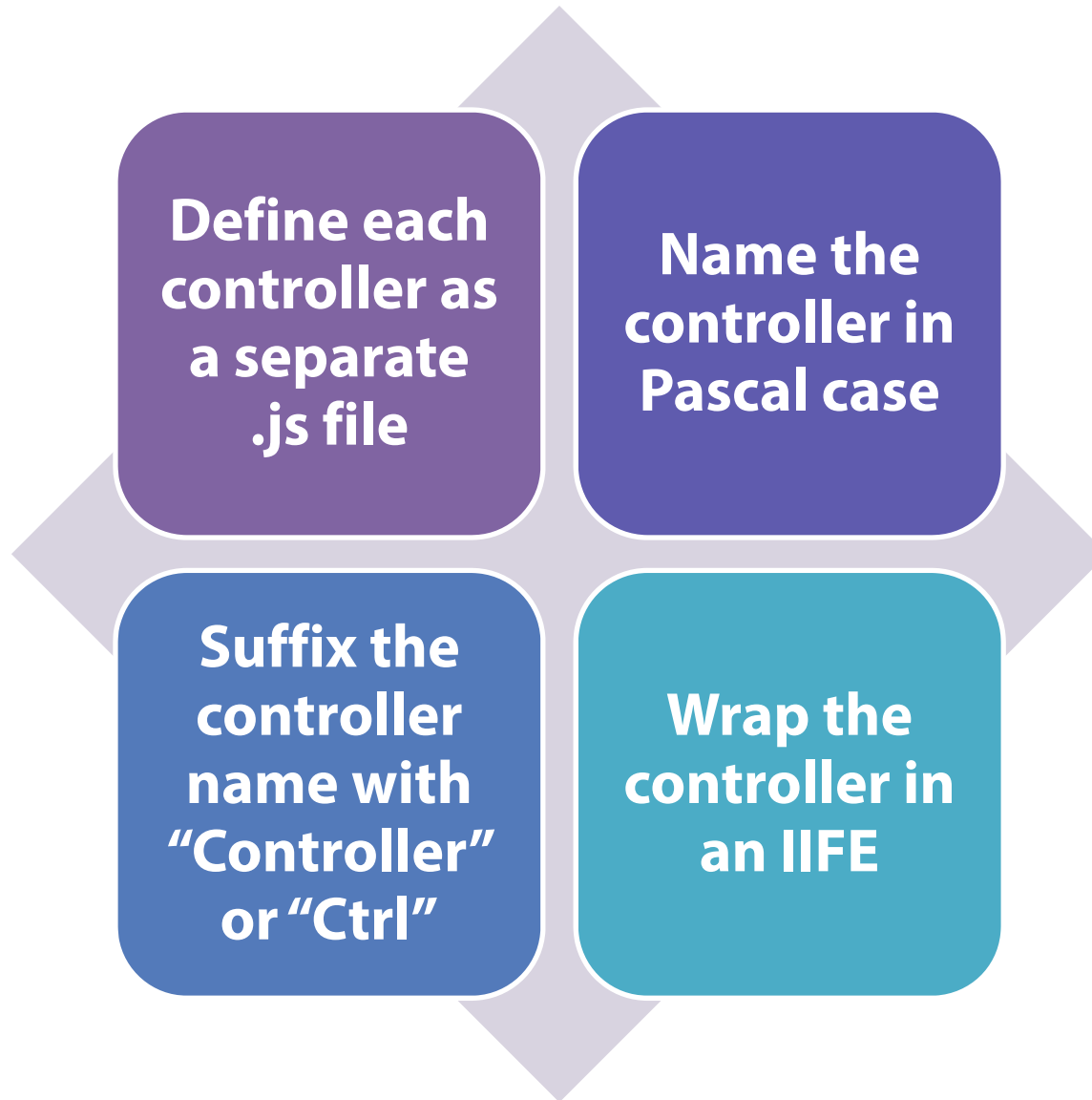
In the view, the model and methods are referenced using an alias defined in the ng-controller



The \$scope still exists, but it is behind the scenes





"Controller As"



Controller Best Practices

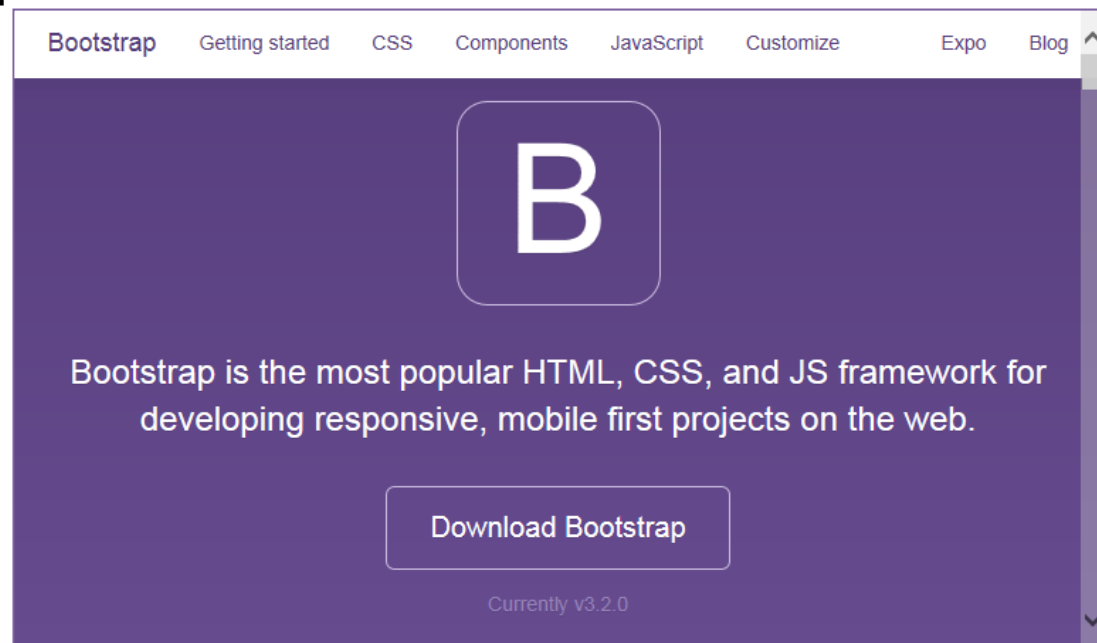


Hide Image	Product	Code	Available	Price
	Leaf Rake	GDN-0011	March 19, 2009	\$19.95
	Hammer	TBX-0048	May 21, 2013	\$8.99

Product List					
Hide Image	Product	Code	Available	Price	
	Garden Cart	GDN-0023	March 18, 2010	\$32.99	Edit
	Hammer	TBX-0048	May 21, 2013	\$8.99	Edit
	Leaf Rake	GDN-0011	March 19, 2009	\$19.95	Edit
	Saw	TBX-0022	May 15, 2009	\$11.55	Edit



- Framework for **prettifying** Web pages
- Helps you build **responsive** Web applications
- Handles **browser differences**
- Developed by **Twitter**
 - <http://getbootstrap.com/>
- **Large third party community**



Bootstrap Grid System

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Module Objectives



Turn business requirements into a View, Model and Controller



Build the Controller



Organize the Project Structure



Defined the Model



Build the View



Implement the Methods

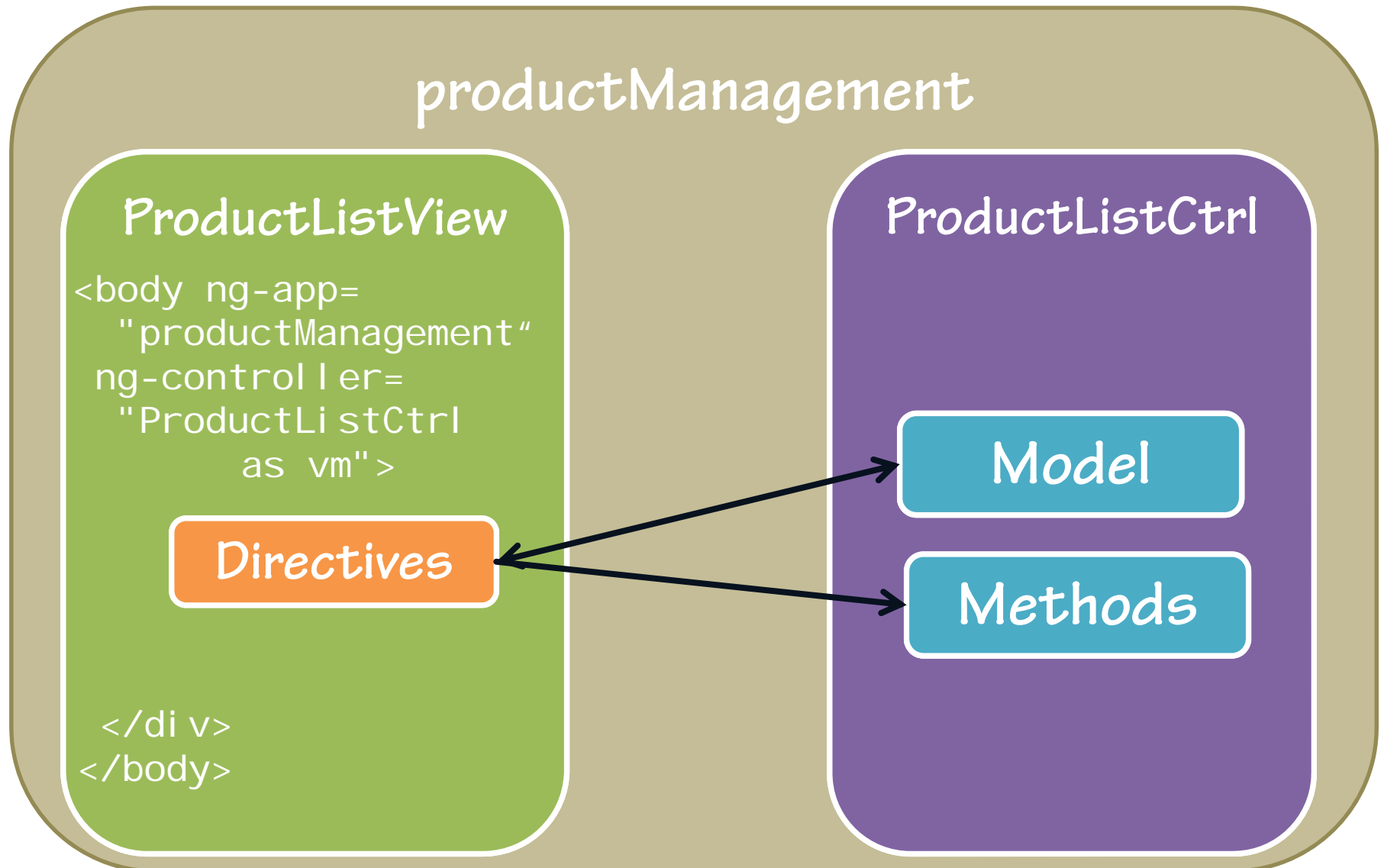


Define the Module



Style the View with Bootstrap

Summary



Angular

Line of Business Applications

Business Requirements



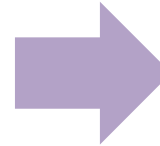
- Application Features
- Design Considerations
- Architecture

Layout and Navigation



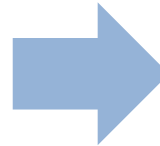
- Page Layout (View)
- Style (Bootstrap)
- Navigation (Routing)

Data Access



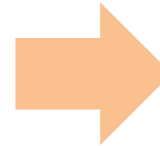
- Retrieving data
- Saving data

Data Entry Forms



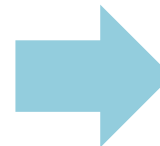
- Form layout
- Validation
- Submittal

Business Logic



- Services

Data Visualization



- Charts