

# Retrieving Data



Deborah Kurata

@deborahkurata | [blogs.msmvps.com/deborahk/](https://blogs.msmvps.com/deborahk/)

# Touch Points



- Michelangelo Buonarroti via Wikimedia Commons

<https://www.flickr.com/photos/34409164@N06/4277702120>

Retrieving Data

Filtering, Shaping, Querying Data

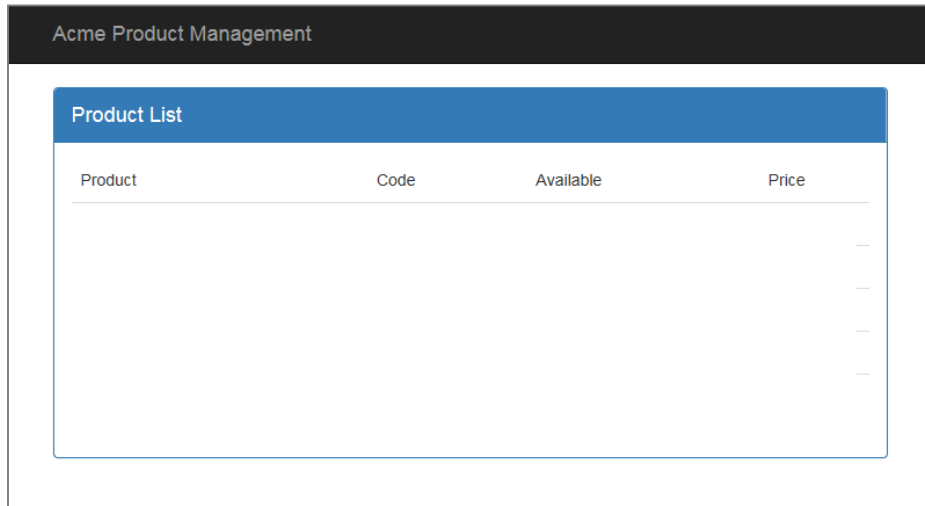
Saving Data

Error Handling

User Authentication

User Authorization

# Retrieving Data



Product	Code	Available	Price

GET Request  
api /products

API

```
public class ProductsController : ApiController
{
    public IEnumerable<Product> Get()
    {
        var productRepository = new ProductRepository();
        return productRepository.Retrieve();
    }
}
```

# Retrieving Data

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



```
[  
  {"description": "Leaf rake with 48-inch wooden handle.",  
   "price": 19.95,  
   "productCode": "GDN-0011",  
   "productId": 1,  
   "productName": "Leaf Rake",  
   "releaseDate": "2009-03-19T00:00:00"},  
  {"description": "15 gallon capacity rolling garden cart",  
   "price": 32.99,  
   "productCode": "GDN-023",  
   "productId": 2,  
   "productName": "Garden Cart",  
   "releaseDate": "2010-03-18T00:00:00"},  
  ... ]
```

# Module Objectives

Build the  
Model

Build the  
Repository

Build the  
Controller

Compare \$http and  
\$resource

Call the Web API  
from Angular

# Building the Web API Model

What is a model?

Why do we need one?

What does it look like?

# What Is a Web API Model?

- Models the data exposed by the Web API
- Specifies the shape and elements of that data

```
"productId": 1,  
"productName": "Leaf Rake",  
"productCode": "GDN-0011",  
"releaseDate": "March 19, 2009",  
"description": "Leaf rake with 48-inch wooden handle.",  
"price": 19.95
```

- Defines the return type for the GET operations

```
// GET api/values  
public IEnumerable<string> Get()  
{  
    return new string[] { "value1", "value2" };  
}
```

## Why Do We Need a Model?

To define the properties returned from a GET request



# What Does a Model Look Like?

```
public class Product
{
    public string Description { get; set; }
    public decimal Price { get; set; }
    public string ProductCode { get; set; }
    public int ProductId { get; set; }
    public string ProductName { get; set; }
    public DateTime ReleaseDate { get; set; }
}
```

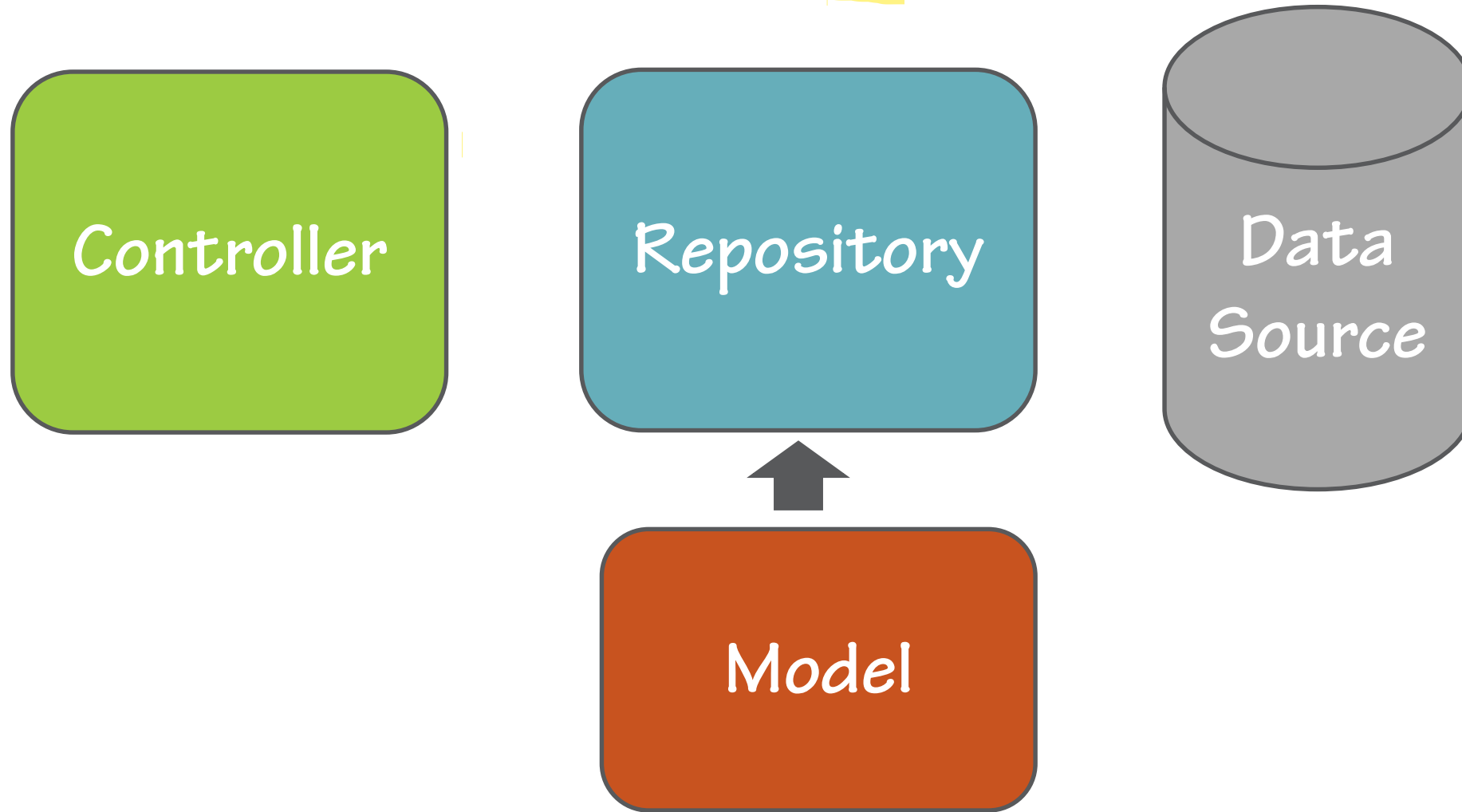
# Building a Repository

What is a repository?

How is data obtained?

What does a repository look like?

# What Is a Repository?



# How Is Data Obtained?

- A relational database
  - ADO.NET
  - Entity Framework
- A file
  - JSON
  - XML
- Any other data source

# What Does a Repository Look Like?

- Retrieve
  - Retrieve the data
  - Populate a set of objects using the model
  - Return those objects
- Save
  - Take in a new or updated item
  - Updates the data source

# Building the Web API Controller

- Inherits from ApiController
- Defines the methods associated with the action verbs
  - GET
  - PUT
  - etc.

# Example Controller

```
public class ValuesController : ApiController
{
    // GET api/values
    public IEnumerable<string> Get()
    {
        return new string[] { "value1", "value2" };
    }
    . . .
}
```

# Angular Services for Calling a Web Service

\$http

\$resource



```
$http.get("/api/products/")  
  .then(function(response) {  
    vm.products = response.data;  
  });
```

## \$http

Built into Angular Core

Facilitates communication with a back-end service

PUT GET etc

Asynchronous

the call returns a promise not an immediate data

# REST

- REpresentational State Transfer
- Requests and responses involve the transfer of **resources**
- Resources are identified with a URL  
**/api/products/**
- Requests utilize a standard set of HTTP verbs  
**GET | POST | PUT | DELETE**
- For more information: "REST Fundamentals"

Resources can be html pages images or a set of data

```
function productResource($resource) {  
    return $resource("/api/products/:id")  
}
```

```
productResource.query(function(data) {  
    vm.products = data;  
});
```

Query : retrieve array of data  
GET : get a single resource  
SAVE: to save a specific resource

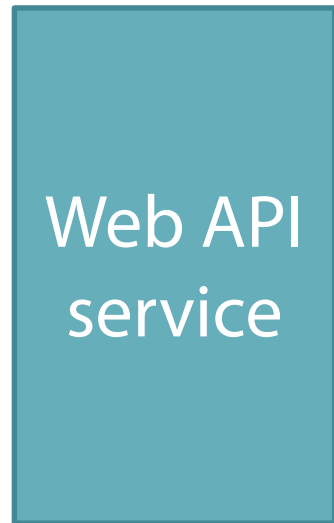
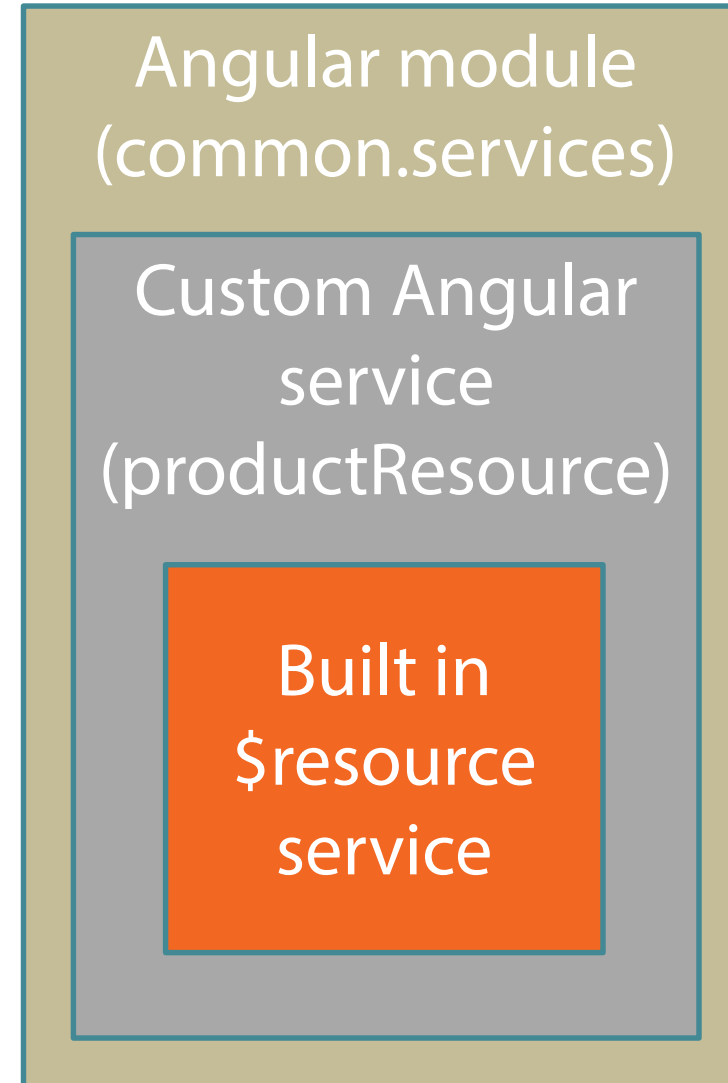
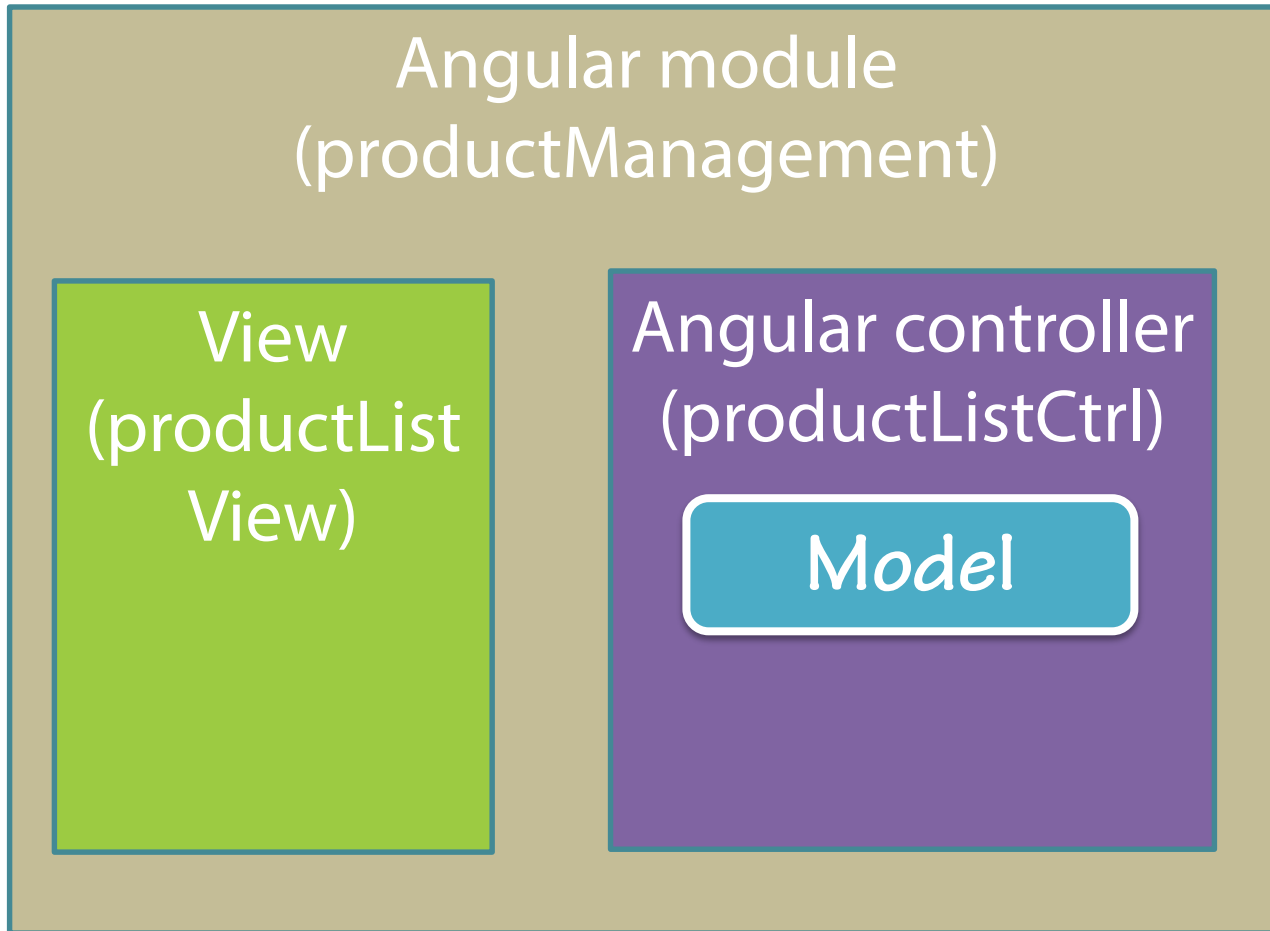
## \$resource

Separate Angular component: angular-resource

Abstraction on top of \$http for calling RESTful services

Requires less code

# Calling a Web API from Angular



# Touch Points



- Michelangelo Buonarroti via Wikimedia Commons  
<https://www.flickr.com/photos/34409164@N06/4277702120>



Retrieving Data

Filtering, Shaping, Querying Data

Saving Data

Error Handling

User Authentication

User Authorization

# This Module Covered



## Web API

- Building the Model
- Building the Repository
- Building the Controller

## Angular

- Comparing \$http and \$resource
- Calling the Web API from Angular