

BUILDING AN ANGULAR WEB APP FROM THE GROUND UP - PART 2

Deborah Kurata

Consultant

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

deborahk@insteptech.com

Deborah Kurata

- ◎ Independent Consultant | Developer | Mentor
 - Web (Angular), .NET
- ◎ Pluralsight Author
 - AngularJS Line of Business Applications
 - Angular Front to Back with Web API
 - Object-Oriented Programming Fundamentals in C#
- ◎ Microsoft MVP

Session Materials & Sample Code

<https://github.com/DeborahK/SDD2015>

Why Angular?

Expressive
HTML

Powerful
Data Binding

Modularity

Rule-based
Navigation

Easily Call
HTTP Service

Testable

It's Popular

Overview - Part 2



Introduction to SPA

Rule-Based Navigation / Routing

Routing Frameworks (ngRoute/ngNewRouter)

JavaScript Promise

Accessing Data by Calling an HTTP service

Reusable Code: Building a Service

DEMO:

Angular Apps

ROUTING:

Rule-based Navigation

Rate Yourself on Routing

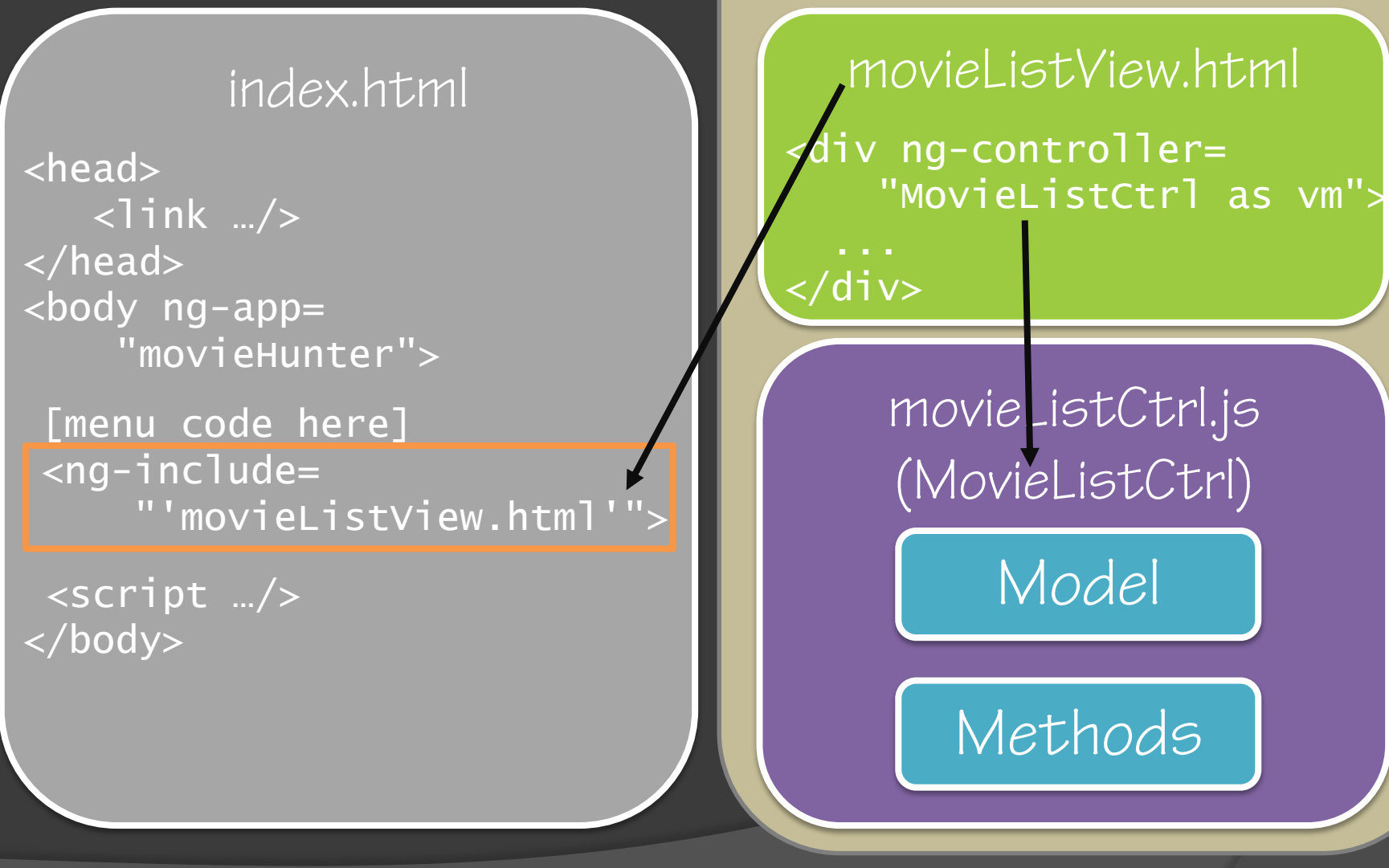
- ⦿ New to Angular - **no** proficiency
- ⦿ Just starting out - **limited** proficiency
- ⦿ Doing it, not fully understanding it - **working** proficiency
- ⦿ Been there, done that, can help others - **full** proficiency

index.html

```
<head>
  <link .../>
</head>
<body ng-app=
  "movieHunter">

  [menu code here]
  <ng-include=
    "'movieListView.html'">

  <script .../>
</body>
```



app.js (movieHunter)

```
movieListView.html
<div ng-controller=
  "MovieListCtrl as vm">
  ...
</div>
```

```
movieListCtrl.js
(MovieListCtrl)
```

Model

Methods

SPA

⦿ Single Page Application

⦿ Only one page?

- Only one full HTML page
- All other pages are HTML Fragments called templates
- And are displayed within the one HTML page

⦿ Goals:

- Enhance user experience
- Reduce round tripping

Multiple Views

InStep Movie Hunter

Search by Title



| Search by Movie Title | | | | |
|-----------------------------|--|----------------|--------------|--------|
| Filter by: | <input type="text"/> | | | |
| Show Poster | Title | Director | Release Date | Rating |
| | Pirates Of The Caribbean: Curse Of The Black Pearl | Gore Verbinski | Jul 9, 2003 | PG-13 |
| | Pirates Of The Caribbean: Dead Man's Chest | Gore Verbinski | Jul 7, 2006 | PG-13 |
| | The Lord of the Rings: The Fellowship of the Ring | Peter Jackson | Dec 19, 2001 | PG-13 |
| | The Lord of the Rings: The Return of the King | Peter Jackson | Dec 17, 2003 | PG-13 |
| | The Lord of the Rings: The Two Towers | Peter Jackson | Dec 18, 2002 | PG-13 |
| | The Point | Fred Wolf | Feb 2, 1971 | NR |

The Fellowship of the Ring

Description: A meek hobbit of the Shire and eight companions set out on a journey to Mount Doom to destroy the One Ring and the dark lord Sauron.

Director: Jackson

Release Date: Dec 19, 2001

Rating: PG-13

Routing

Rule-based
Navigation

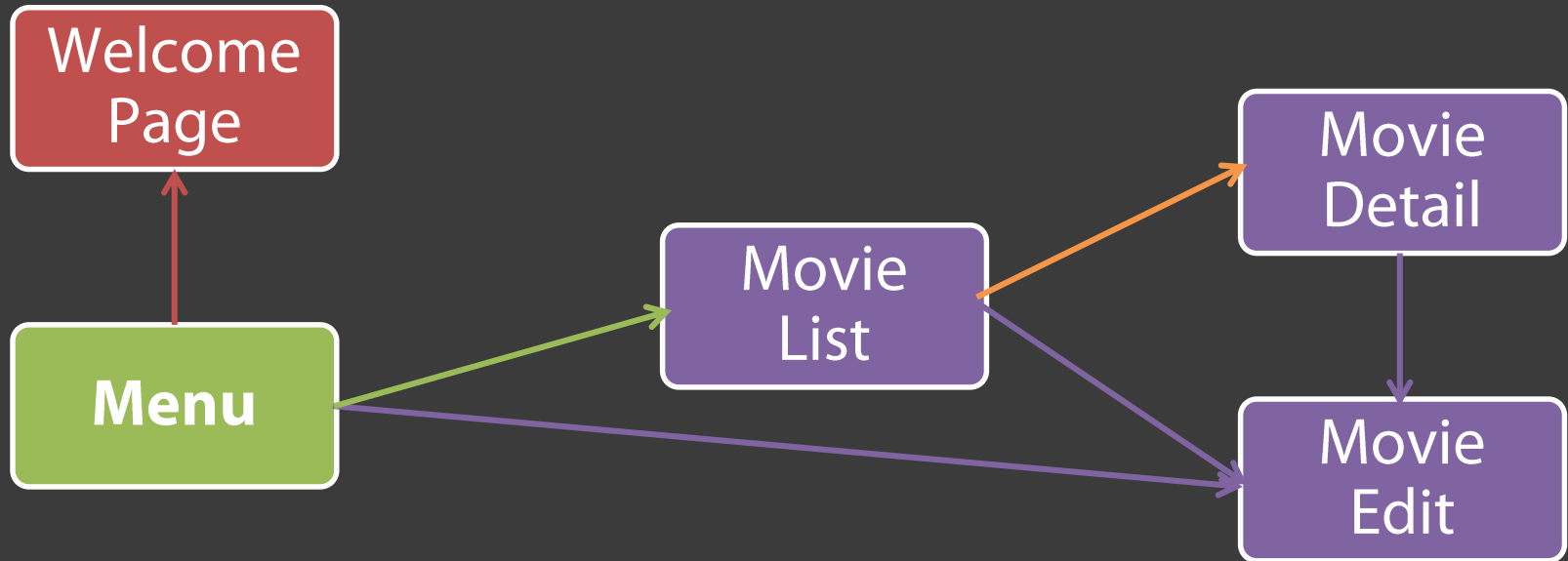
Technique for
navigating
between views

Routes defined
by rules

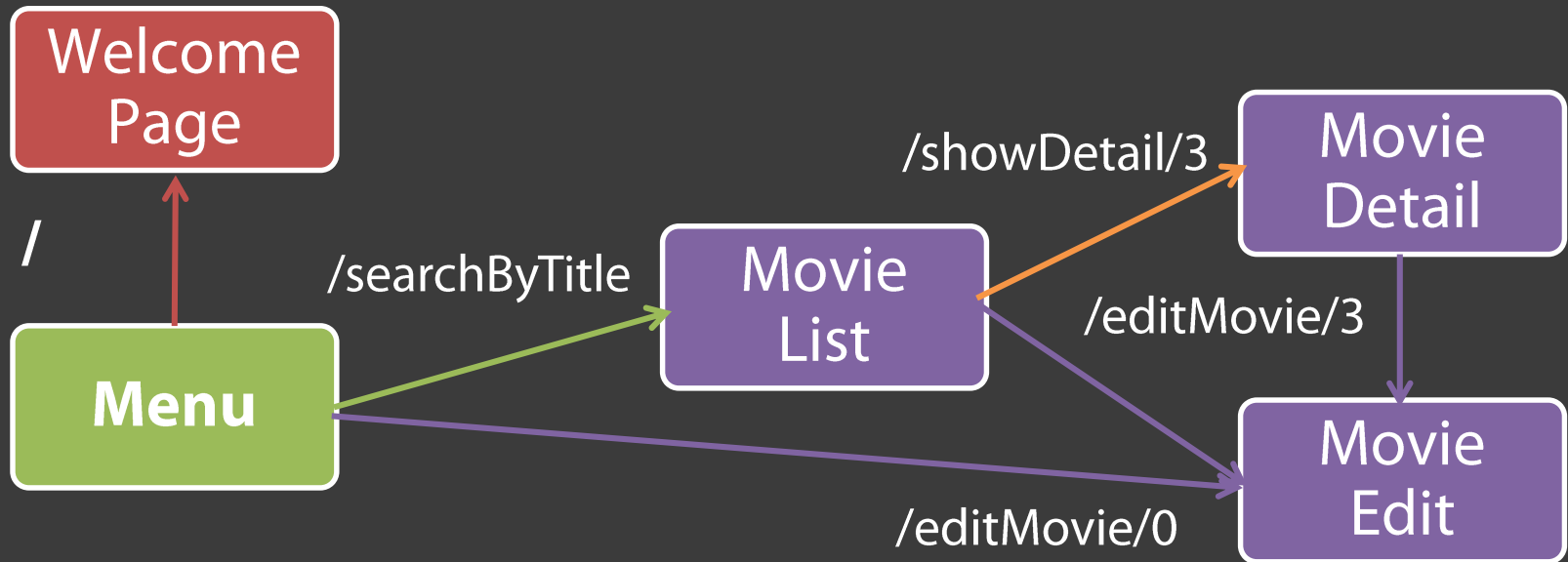
Each route
represents a
specific view

Activating a
route navigates
to that view

Site Map



Routes

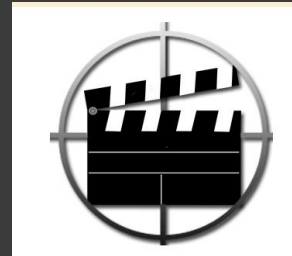


Routing URL Fragments

<mysite>/index.html#/showDetail/3

InStep Movie Hunter

Search by Title



Search by Movie Title

Filter by:

Show Poster

| Title | Director | Release Date | Rating |
|--|----------------|--------------|--------|
| Pirates Of The Caribbean: Curse Of The Black Pearl | Gore Verbinski | Jul 9, 2003 | PG-13 |
| Pirates Of The Caribbean: Dead Man's Chest | Gore Verbinski | Jul 7, 2006 | PG-13 |
| The Lord of the Rings: The Fellowship of the Ring | Peter Jackson | Dec 19, 2001 | PG-13 |
| The Lord of the Rings: The Return of the King | Peter Jackson | Dec 17, 2003 | PG-13 |
| The Lord of the Rings: The Two Towers | Peter Jackson | Dec 18, 2002 | PG-13 |
| The Point | Fred Wolf | Feb 2, 1971 | NR |

The Fellowship of the Ring

Description: A meek hobbit of the Shire and eight companions set out on a journey to Mount Doom to destroy the One Ring and the dark lord Sauron.

Director: Jackson

Release Date: Dec 19, 2001

Rating: PG-13

Routing Frameworks

ngRoute

UI Router

ngNewRouter

ngRoute

Available
from

- Angular Web Site
- <https://angularjs.org/>
- angular-route.js

Based on

- URL fragment identifiers
- <http://mysite/index.html#/searchByTitle>

Each route
has

- URL fragment identifier
- View template
- Optional Controller

UI Router

Available
from

- Third-party Angular UI Library
- <https://github.com/angular-ui/ui-router>
- angular-ui-router.js

Based on

- Application States

Each state
has

- Optional URL fragment identifier
- View template
- Optional Controller

ngNewRouter

Available
from

- Angular 1.4
- <https://github.com/angular/router>
- Router.es5.js

Based on

- URL fragment identifiers
- <http://mysite/index.html#/searchByTitle>

Each route
has

- URL fragment identifier (path)
- Component

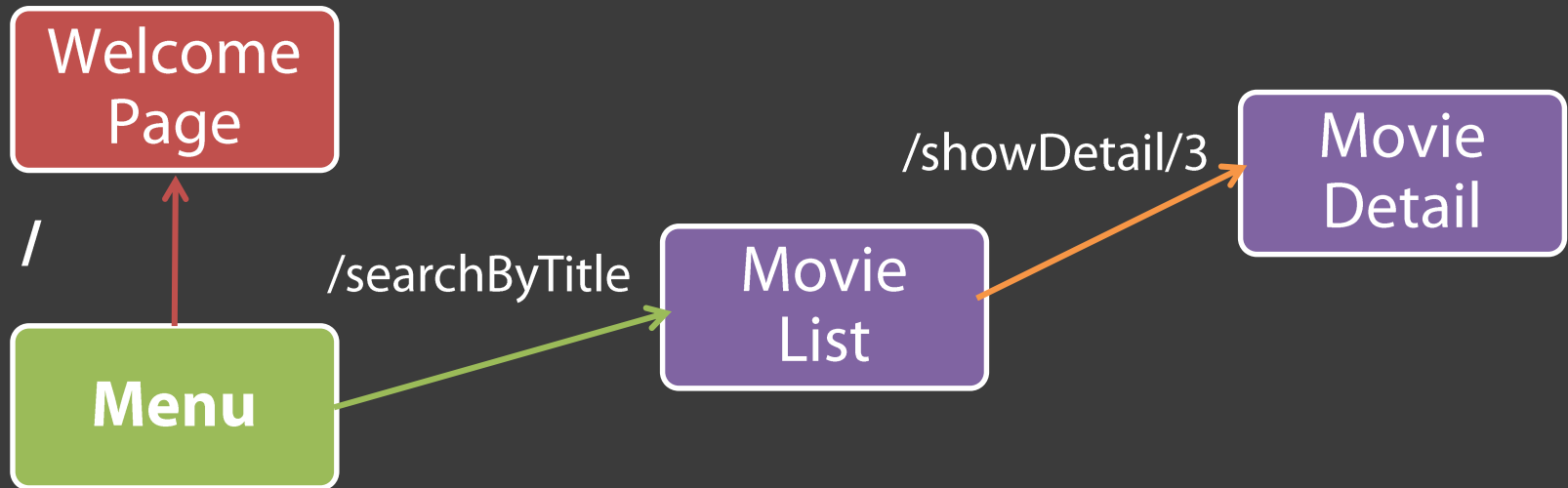
ngRoute

<https://docs.angularjs.org/api/ngRoute>

ngRoute: index.html

```
<div class="container">  
  <div ng-view></div>  
</div>
```

Routes



ngRoute: app.js

```
var app = angular.module("movieHunter", ["ngRoute"]);
```

```
app.config(['$routeProvider',  
  function ($routeProvider) {
```

```
    $routeProvider
```

```
      .when("/", {  
        templateUrl: "app/welcomeView.html"  
      })
```

```
      .when("/searchByTitle", {  
        templateUrl: "app/movies/movieListView.html",  
        controller: "MovieListCtrl as vm"  
      })
```

```
      .when("/showDetail/:movieId", {  
        templateUrl: "app/movies/movieDetailView.html",  
        controller: "MovieDetailCtrl as vm"  
      })
```

```
      .otherwise("/");
```

```
  });
```

Navigation

⦿ Anchor tags

- Link to the appropriate route
- `<a ng-href="#searchByTitle">`

⦿ \$location service

- `$location.path("/searchByTitle");`

ngRoute: index.html

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li>
      <a ng-href="#searchByTitle">Search by Title</a>
    </li>
  </ul>
</div>
```

ngRoute: movieDetailView.html

```
<div class="panel-footer">  
  <a class="btn btn-default"  
    ng-href="#searchByTitle"  
    style="width:80px">  
    <i class="glyphicon glyphicon-chevron-left"></i>  
    Back  
  </a>  
</div>
```

DEMO:

Navigation & Routing

ngNewRouter

<https://github.com/angular/router>

<https://angular.io/>

ngNewRouter: index.html

```
<div class="container">  
  <div ng-viewport></div>  
</div>
```

ngNewRouter : app.js

```
var app = angular.module("movieHunter", ["ngNewRouter"]);  
app.controller('AppController', ['$router', AppController]);  
  
function AppController ($router) {  
  $router.Config([  
    { path: '/', component: 'welcome' },  
    { path: '/searchByTitle', component: 'movieList' },  
    { path: '/showDetail/:movieId', component: 'movieDetail' }  
  ]),  
}
```

Routable Component in 1.4+

- ⦿ Defined by the directory structure
- ⦿ Or configure with
 - `$componentLoader`

`components/`

`welcome/
welcome.html`

`movieList/
movieList.html
movieList.js`

`movieDetail/
movieDetail.html
movieDetail.js`

ngNewRouter: movieDetailView.html

```
<div class="panel-footer">  
  <a class="btn btn-default"  
    ng-link="movieList"  
    style="width:80px">  
    <i class="glyphicon glyphicon-chevron-left"></i>  
    Back  
  </a>  
</div>
```


Advanced Features

- ⦿ Multiple viewports
 - Sibling Components
- ⦿ Child Views
 - Tab-based UI
- ⦿ Lifecycle Hooks
 - canActivate | activate
 - canDeactivate | deactivate

Routing

Rule-based Navigation

- ⦿ SPA: Single Page Application
- ⦿ Use routing to display the views
- ⦿ Routing provides rule-based navigation
- ⦿ Define the routes using `.config`
- ⦿ Navigate to a route using `<a>`
- ⦿ Consider using the new 1.4 router

DATA ACCESS:

Calling an HTTP Service

Rate Yourself on Data Access

- ⦿ New to Angular - no proficiency
- ⦿ Just starting out - limited proficiency
- ⦿ Doing it, not fully understanding it - working proficiency
- ⦿ Been there, done that, can help others - full proficiency

HTTP Service

Easily Call
HTTP Service

- ⦿ What is an HTTP service?
- ⦿ Why call one from our Angular app?



Web Browser

Web Server

URL Request (www.mysite.com)

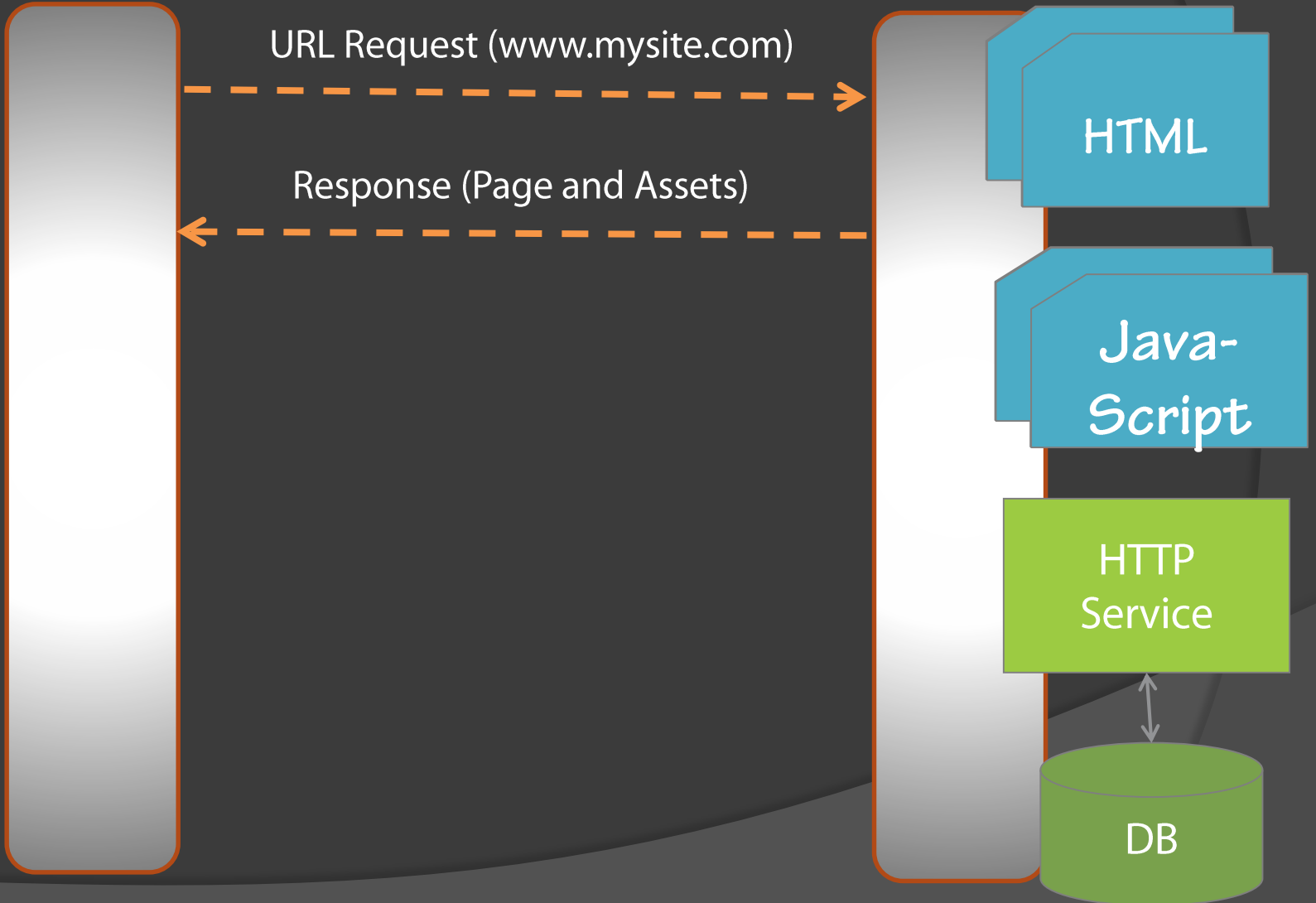
Response (Page and Assets)

HTML

Java-
Script

HTTP
Service

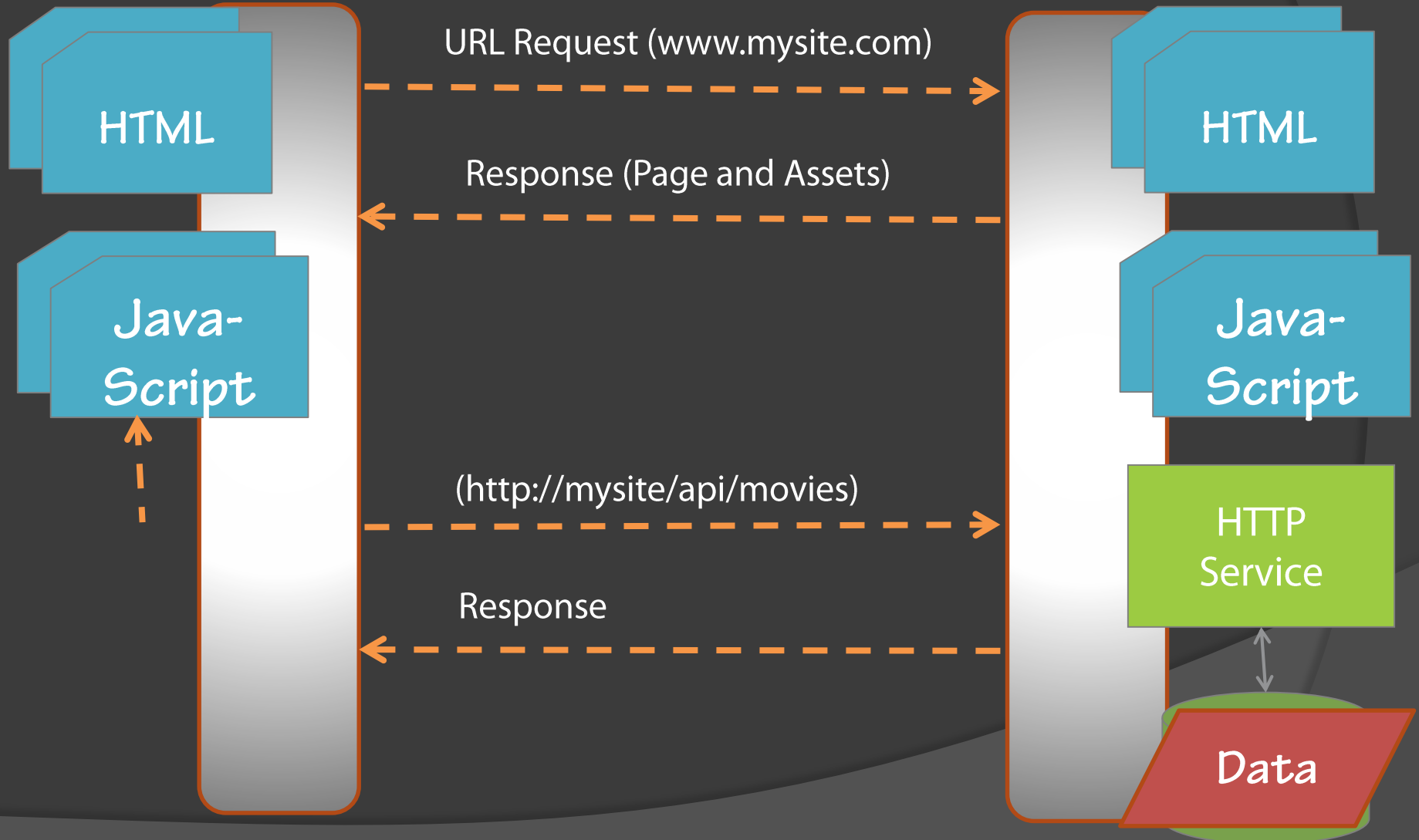
DB



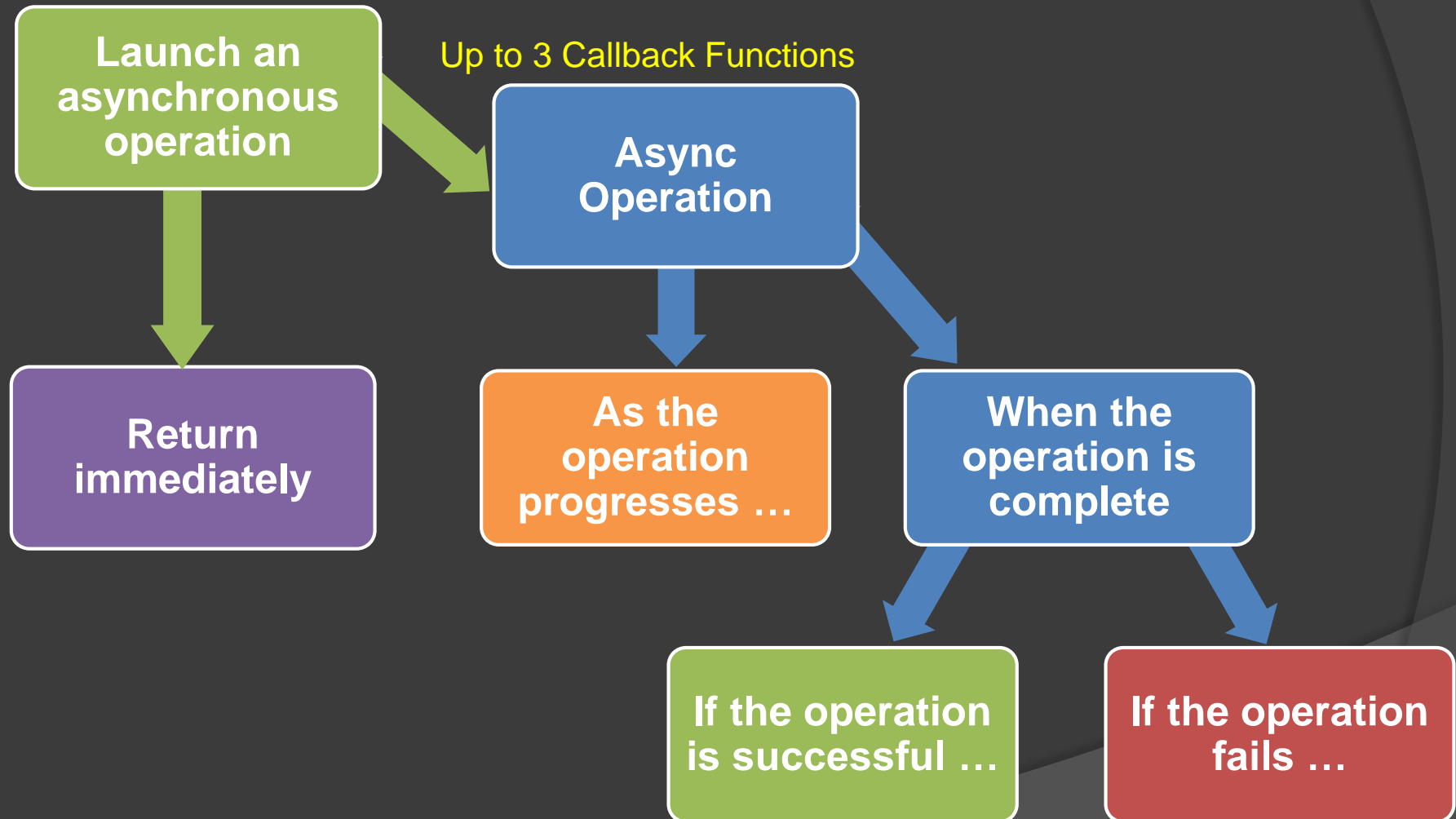


Web Browser

Web Server



JavaScript Promise



Calling HTTP Service

\$http

\$resource

\$httpBackend

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

\$http

Built into Angular Core

Facilitates communication with a back-end service

get, post, put, delete, ...

Asynchronous

REST

- ⦿ REpresentational State Transfer
- ⦿ Requests and responses involve the transfer of resources
- ⦿ Resources are identified with a URL
 - ⦿ `/api/movies/`
- ⦿ Requests utilize a standard set of HTTP verbs
 - ⦿ `GET` | `POST` | `PUT` | `DELETE`
- ⦿ And much more!

```
function movieResource($resource) {  
    return $resource("/api/movies/:movieId")  
}
```

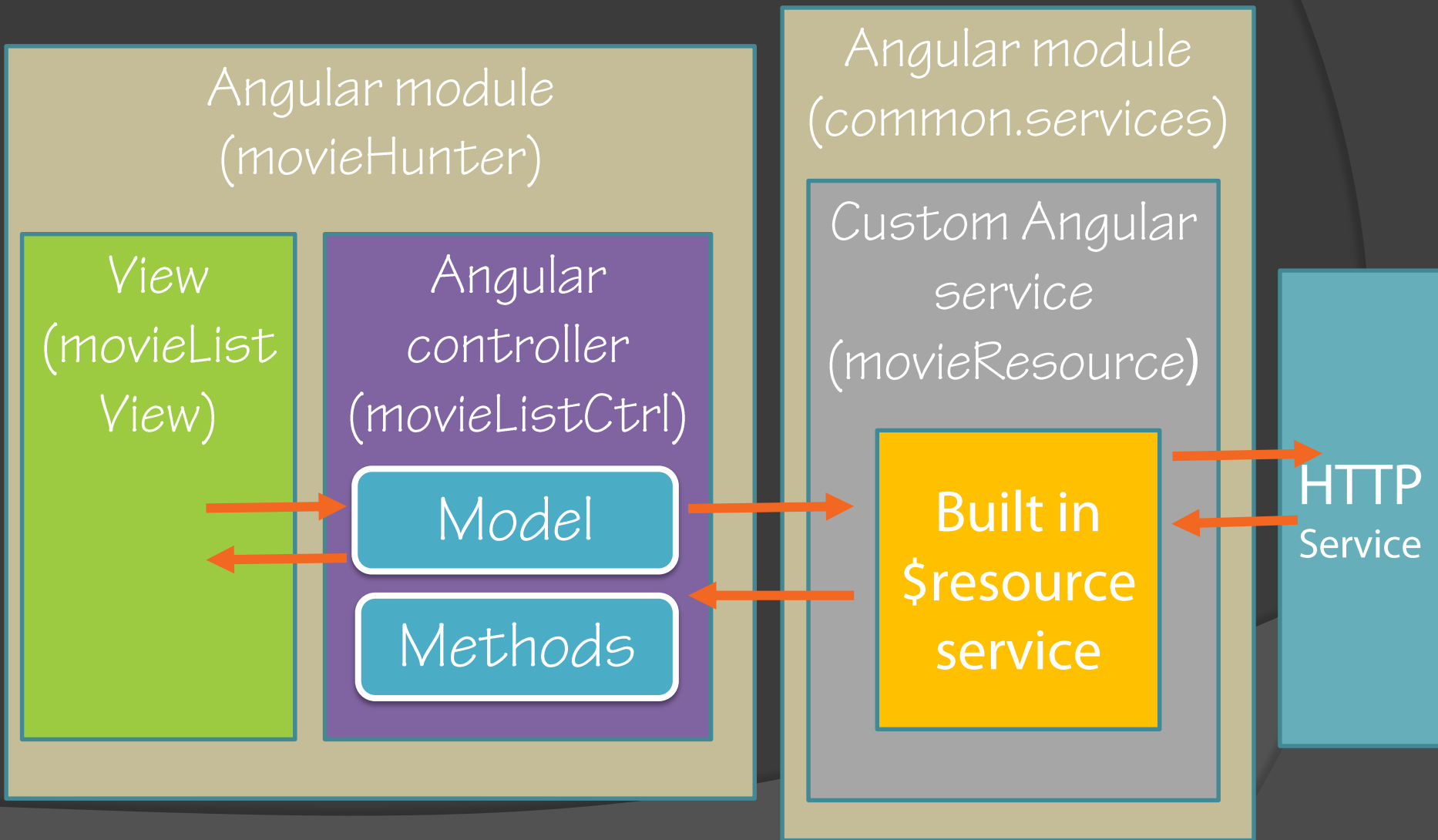
\$resource

Separate Angular component: angular-resource
ngResource

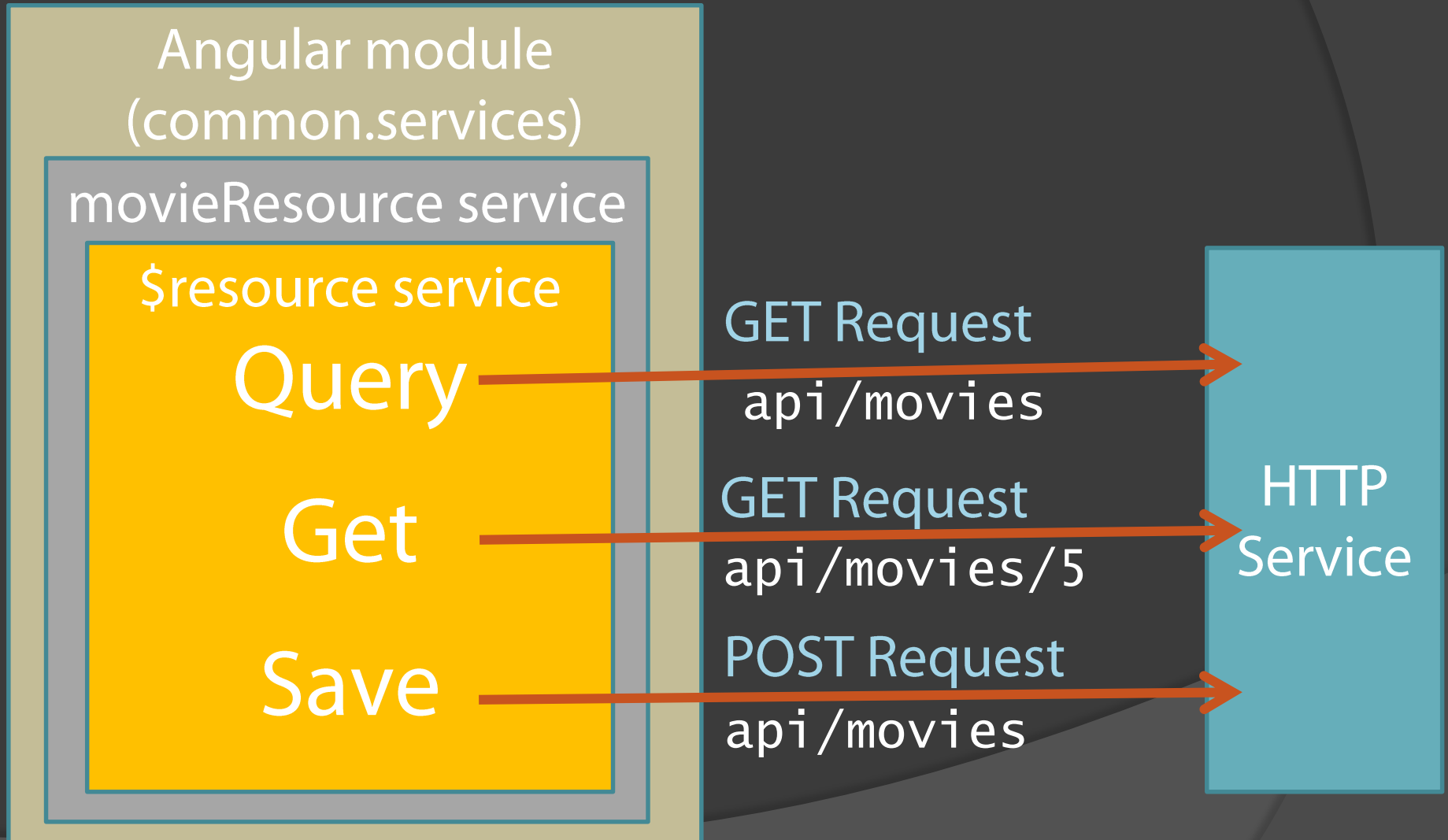
Abstraction on top of \$http for calling RESTful
services

Requires less code

Reusable Data Access Service



Built-In \$resource Methods



\$resource Methods

| Method Name | URL | Description |
|-------------|---------------|-----------------------------------|
| query | /api/movies | Retrieves all of the movies |
| get | /api/movies/5 | Retrieves the specified movie |
| save | /api/movies | Saves modifications to the movies |

```
movieResource.query(function (data) {  
    vm.movies = data;  
});
```

DEMO:

Retrieving Data

DEMO:

Saving Data

HTTP Service

Easily Call
HTTP Service

- ⦿ Easily call an HTTP service using \$resource
- ⦿ Steps:
 - Create a new common module
 - Add a movieResource service
 - Inject the service into a controller
 - Call the \$resource methods
 - query, get, save

Take Away

Expressive
HTML

Powerful
Data Binding

Modularity

Rule-based
Navigation

Easily Call
HTTP Service

Testable

It's Popular

Thank You!

- @deborahkurata
- deborahk@insteptech.com
- <http://blogs.msmvps.com/deborahk>
- <https://github.com/DeborahK/SDD2015>