Angular 101

Deborah Kurata
Developer | Author | Consultant | MVP | GDE
InStep Technologies, Inc

Deborah Kurata

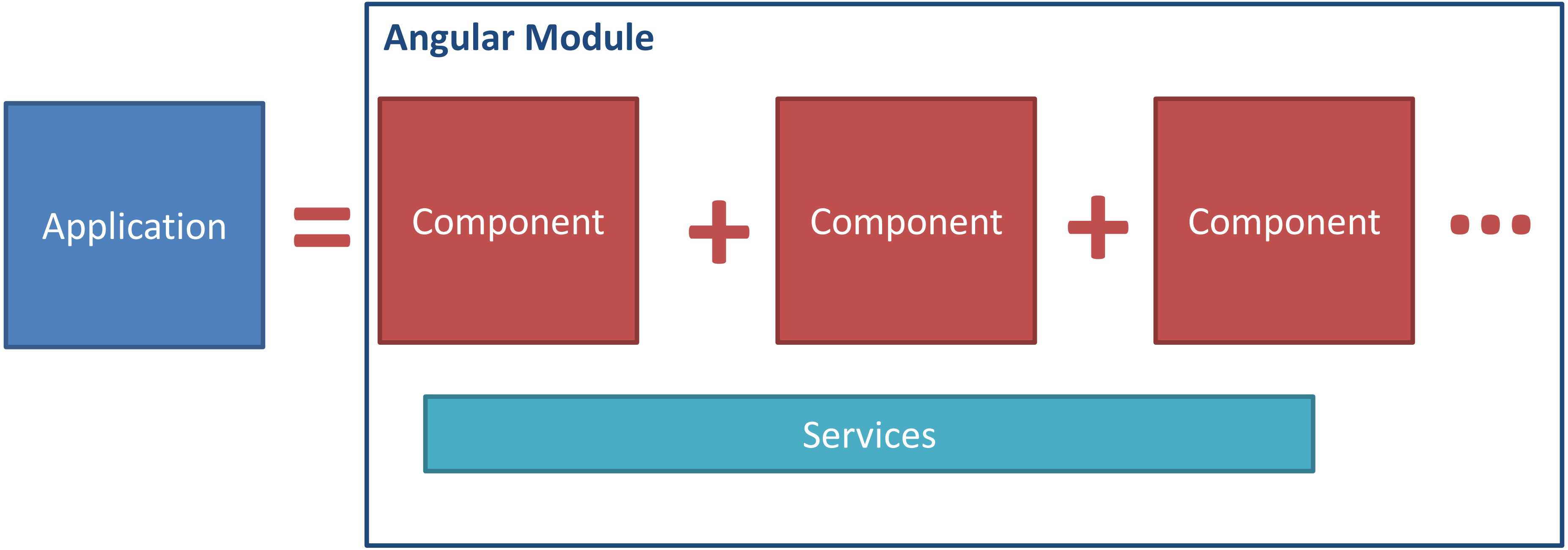
- Developer
- Angular v2 Documentation Team Member (angular.io)
- Pluralsight Author
  - Angular 2: Getting Started
  - Angular 2: Reactive Forms
  - Angular Front to Back with Web API
  - C# Best Practices
- Microsoft Most Valuable Professional (MVP)
- Google Developer Expert (GDE)

https://github.com/DeborahK/MovieHunter

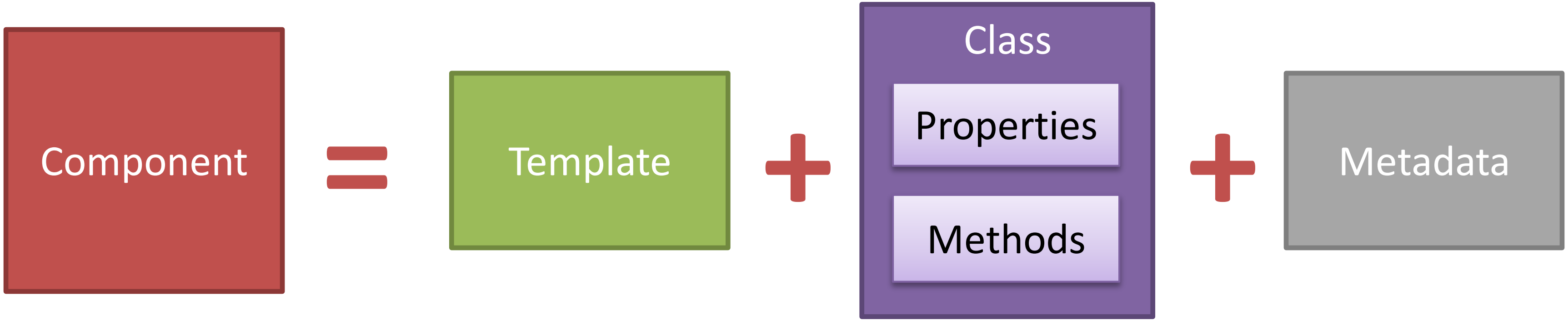
Rate Yourself on Angular

What is Angular?

Angular Application



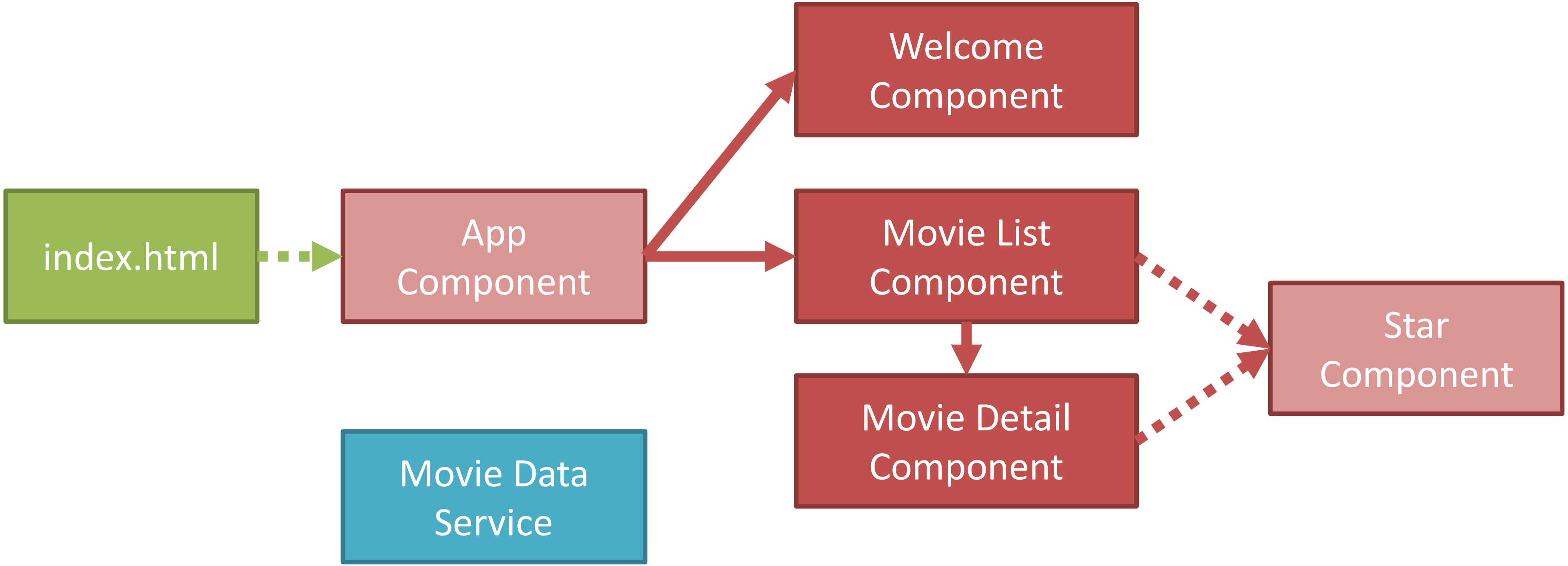
Component



Demo

MovieHunter
Sample Application

Sample Application Architecture



Component

app.component.ts

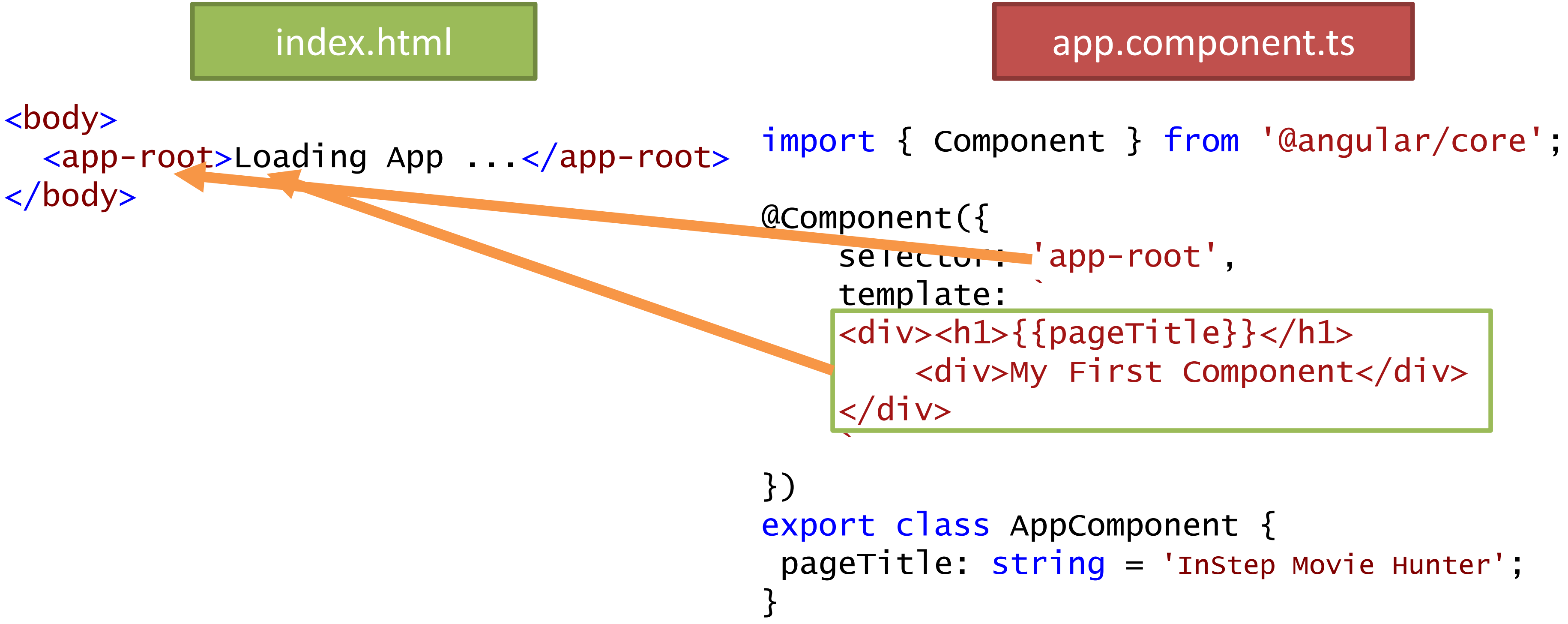
```
import { Component } from '@angular/core';

@Component({
    selector: 'app-root',
    template: `
    <div><h1>{{pageTitle}}</h1>
        <div>My First Component</div>
    </div>
    `
})
export class AppComponent {
    pageTitle: string = 'InStep Movie Hunter';
}
```

ES 2015
Import Syntax

Metadata:
Component
Decorator

ES 2015
Class Syntax

Hosting the Application



index.html

```
<body>
  <app-root>Loading App ...</app-root>
</body>
```

app.component.ts

```
import { Component } from '@angular/core';

@Component({
    selector: 'app-root',
    template: `
    <div><h1>{{pageTitle}}</h1>
        <div>My First Component</div>
    </div>
    `
})
export class AppComponent {
 pageTitle: string = 'InStep Movie Hunter';
}
```

Template

app.component.ts

```
import { Component } from '@angular/core';

@Component({
    selector: 'app-root',
    template: `
    <div><h1>{{pageTitle}}</h1>
        <div>My First Component</div>
    </div>
    `
})
export class AppComponent {
    pageTitle: string = 'InStep Movie Hunter';
}
```

Template

Simple HTML

```
template:
"<h1>{{pageTitle}}</h1>"
```

Multi-line HTML

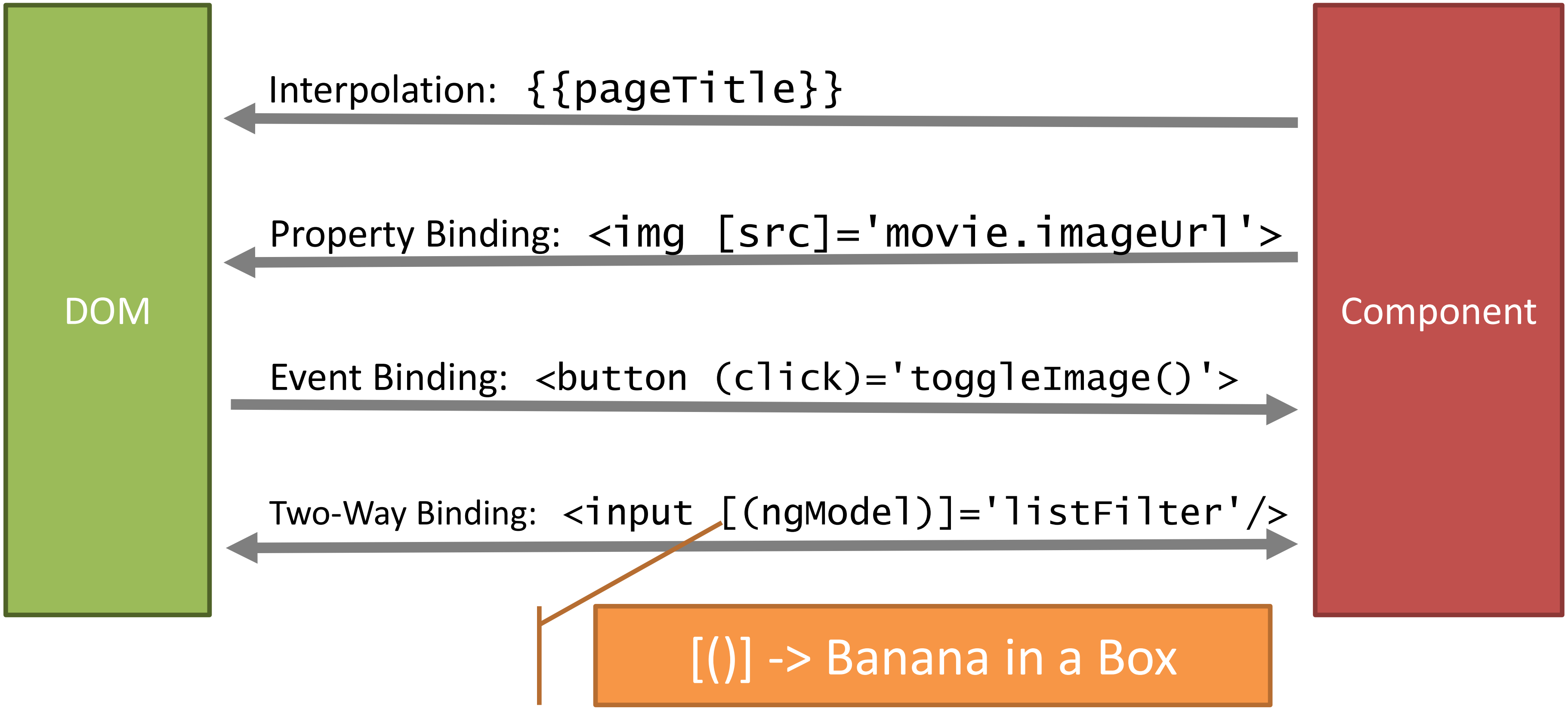
```
template: `
<div>
 <h1>{{pageTitle}}</h1>
 <div>
    My First Component
 </div>
</div>
`
```

ES 2015
Back Ticks
Template Literal

URL

```
templateUrl:
"app/movies/movie-
list.component.html"
```

Data Binding



DOM

Interpolation: {{pageTitle}}

Property Binding: <img [src]='movie.imageUrl'>

Event Binding: <button (click)='toggleImage()'>

Two-Way Binding: <input [(ngModel)]='listFilter'/>

[()] -> Banana in a Box

Component

Directives

Custom Directives

- Defined by our components

Structural Directives

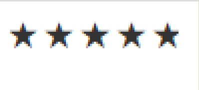
```
<table *ngIf="movies.length">

<tr *ngFor="let movie of movies">
```

Demo

Movie List Template

Using a Component

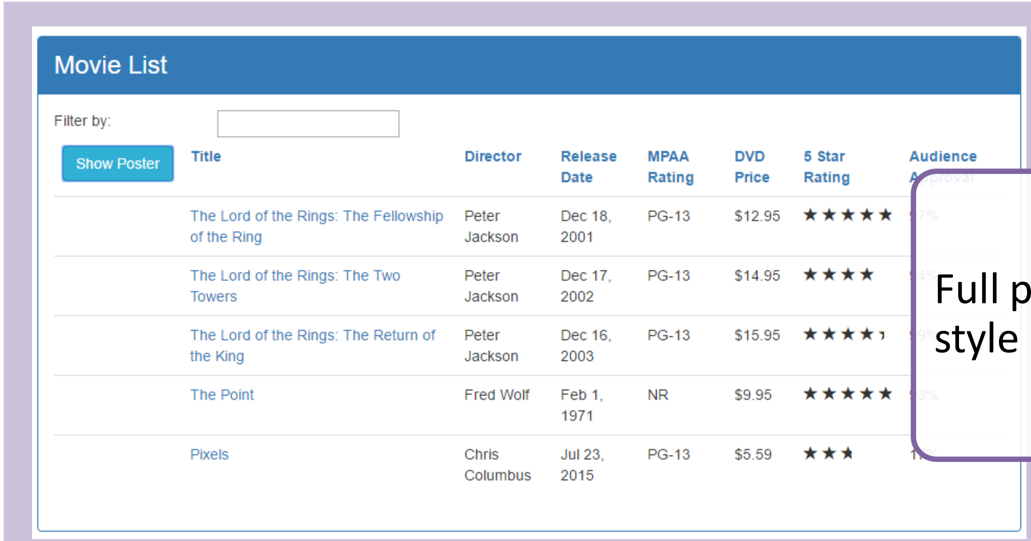
As a Directive



★★★★★

App Component OR Nested Component

As a Routing target



Movie List

Filter by:

Show Poster

	Title	Director	Release Date	MPAA Rating	DVD Price	5 Star Rating	Audience Approval
	The Lord of the Rings: The Fellowship of the Ring	Peter Jackson	Dec 18, 2001	PG-13	$12.95	★★★★★	97%
	The Lord of the Rings: The Two Towers	Peter Jackson	Dec 17, 2002	PG-13	$14.95	★★★★	94%
	The Lord of the Rings: The Return of the King	Peter Jackson	Dec 16, 2003	PG-13	$15.95	★★★★	99%
	The Point	Fred Wolf	Feb 1, 1971	NR	$9.95	★★★★★	93%
	Pixels	Chris Columbus	Jul 23, 2015	PG-13	$5.59	★★★	17%

Full page style view

Using a Component

As a Directive

- Component decorator must specify a selector
- Selector used as a custom directive

```
<body>
  <app-root>Loading App ...</app-root>
</body>
```

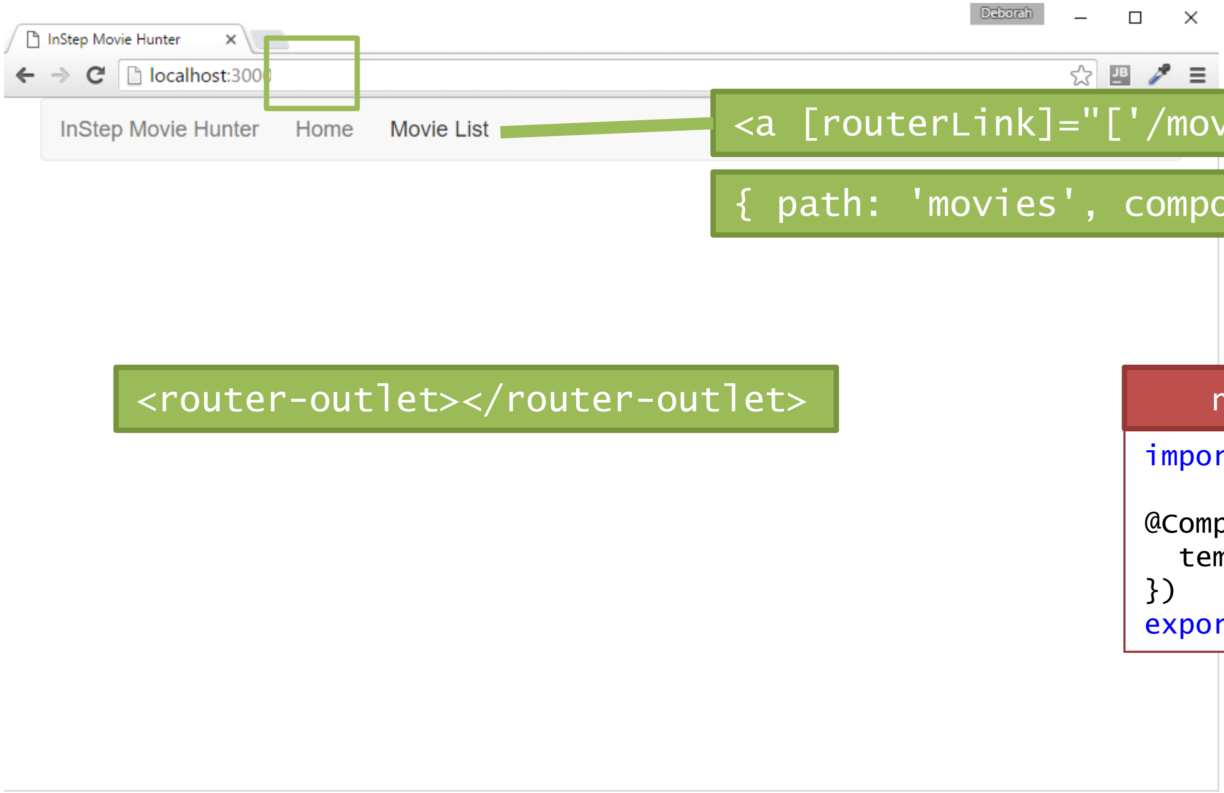
```
<td>
  <app-star [rating]="movie.starRating">
  </app-star>
</td>
```

As a Routing Target

- No selector required
- Must specify a route

```
{ path: 'movies',
  component: MovieListComponent }
```

How Routing Works



```
<a [routerLink]="['/movies']">Movie List</a>
```

```
{ path: 'movies', component: MovieListComponent }
```

```
<router-outlet></router-outlet>
```

movie-list.component.ts

```
import { Component } from '@angular/core';

@Component({
  templateUrl: 'movie-list.component.html'
})
export class MovieListComponent { }
```

Configuring Routes

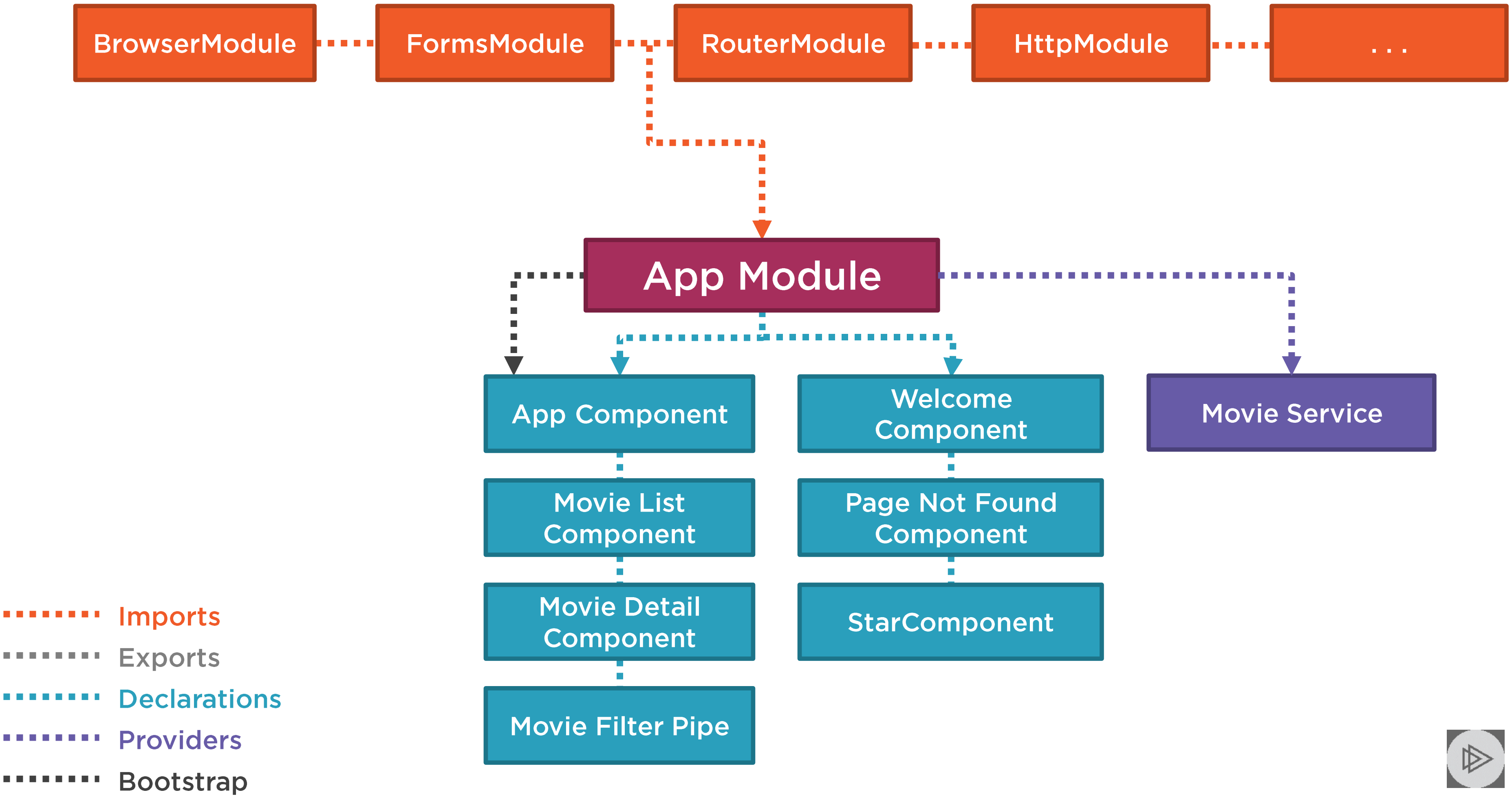
app.module.ts

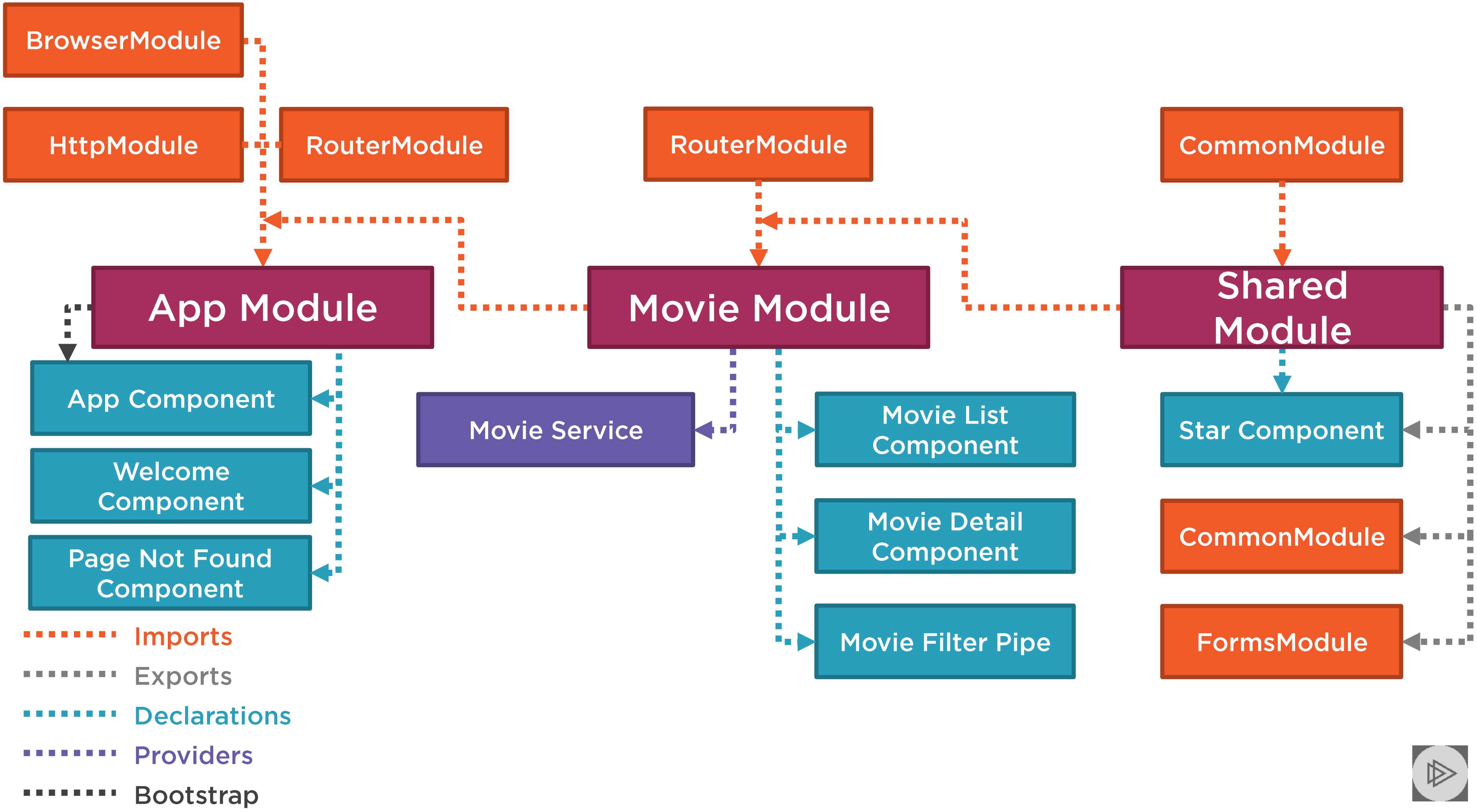
```
[
 { path: 'welcome', component: WelcomeComponent },
 { path: '', redirectTo: 'welcome', pathMatch: 'full' },
 { path: '**', component: PageNotFoundComponent }
]
```

Configuring Routes

movie.module.ts

```
[
 { path: 'movies', component: MovieListComponent },
 { path: 'movies/:id', component: MovieDetailComponent }
]
```

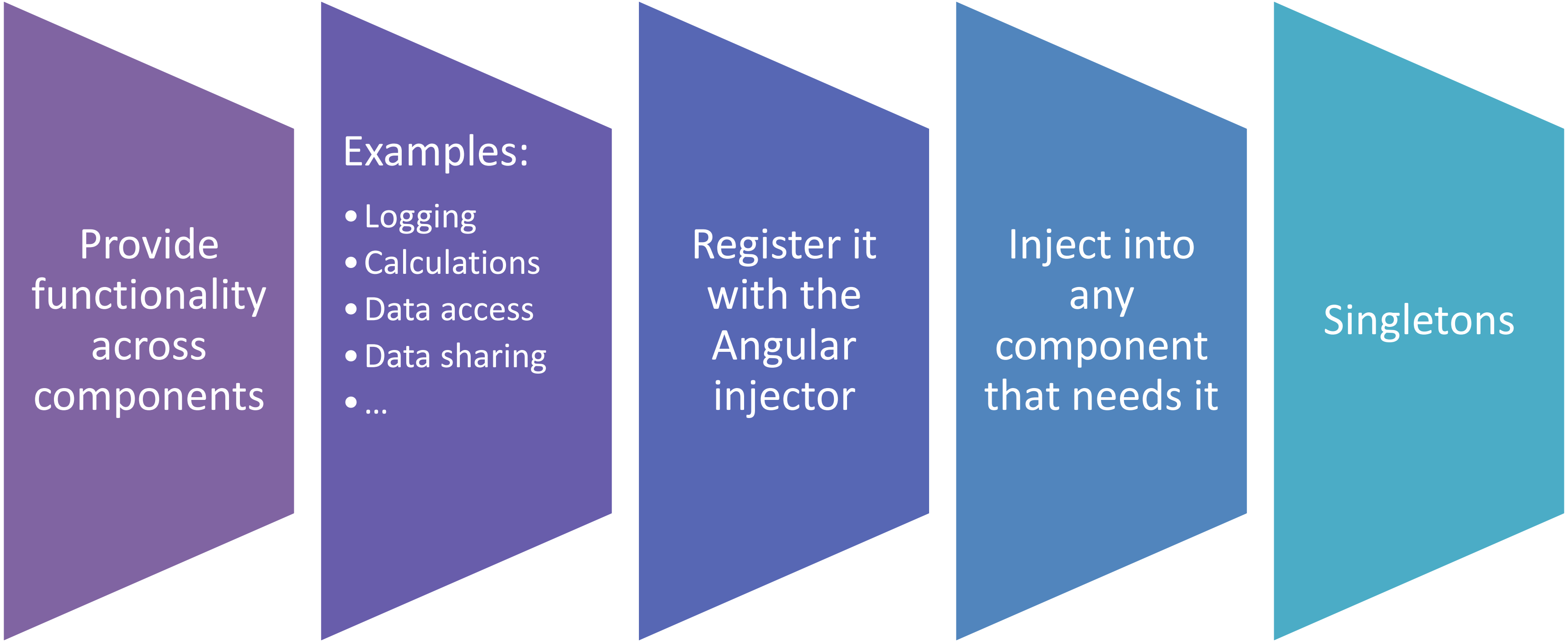

BrowserModule
FormsModule
RouterModule
HttpModule
. . .
App Module
App Component
Movie List Component
Movie Detail Component
Movie Filter Pipe
Welcome Component
Page Not Found Component
StarComponent
Movie Service
Imports
Exports
Declarations
Providers
Bootstrap


BrowserModule
HttpModule
RouterModule
RouterModule
CommonModule
App Module
Movie Module
Shared Module
App Component
Welcome Component
Page Not Found Component
Movie Service
Movie List Component
Movie Detail Component
Movie Filter Pipe
Star Component
CommonModule
FormsModule
Imports
Exports
Declarations
Providers
Bootstrap

Demo

Routing

Services



Provide functionality across components

Examples:

- Logging
- Calculations
- Data access
- Data sharing
- …

Register it with the Angular injector

Inject into any component that needs it

Singletons

Building a Service

movie.service.ts

```
import { Injectable }
        from '@angular/core';

@Injectable()
export class MovieService {
  getMovies() { }
}
```

Registering a Service

app.module.ts

```
...

@NgModule({
 imports: [ BrowserModule,
            FormsModule,
            RouterModule.forRoot(...) ],
  declarations: [ AppComponent,
                  MovieListComponent ],
  providers: [ MovieService ],
  bootstrap: [ AppComponent ]
})
export class MovieModule { }
```

movie.service.ts

```
import { Injectable }
       from '@angular/core';

@Injectable()
export class MovieService {
  getMovies() { }
}
```

Using a Service

movie-list.component.ts

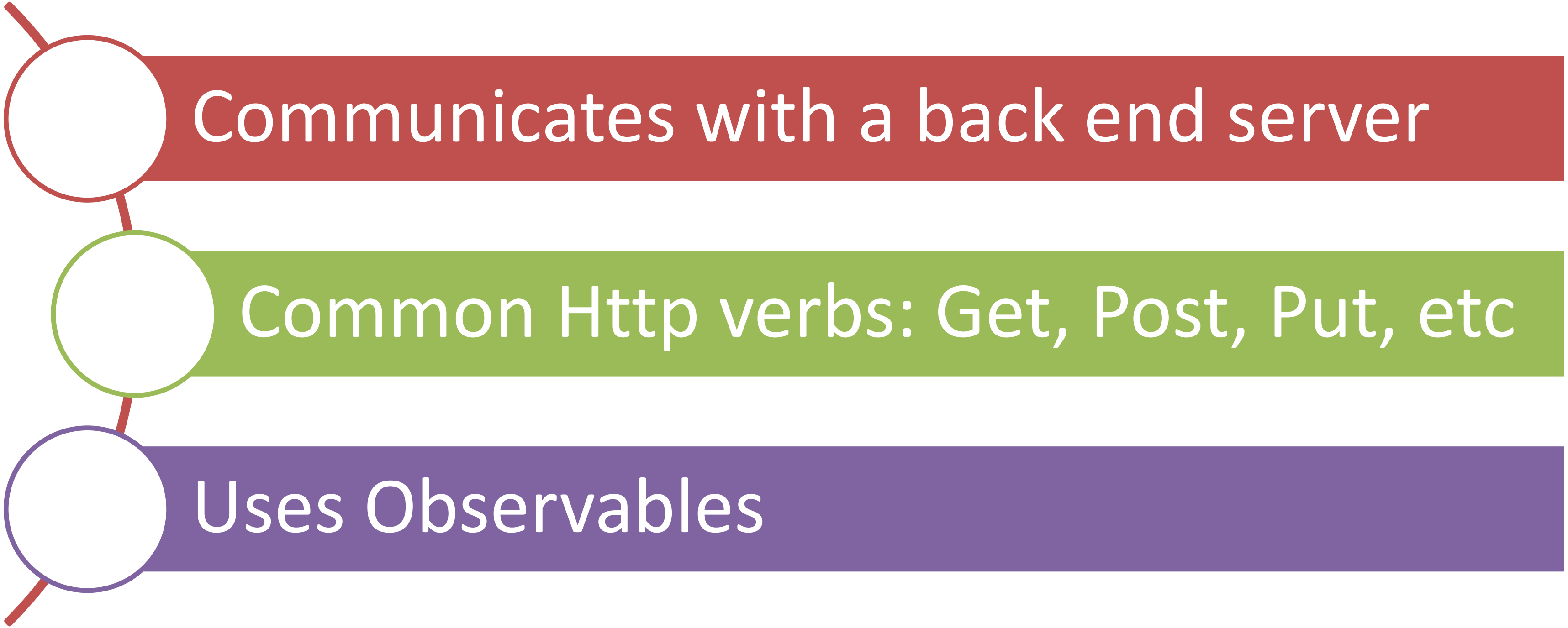
```
import { Component } from '@angular/core';
import { MovieService } from "./movie.service";

@Component({
  templateURL: "app/movies/movie-list.component.html"
})
export class MovieListComponent {
 constructor(private movieService: MovieService) { }
}
```

```
private movieService: MovieService;
constructor(myMovieService: MovieService) {
   this.movieService =  myMovieService;
}
```

movie.service.ts

```
import { Injectable }
        from '@angular/core';

@Injectable()
export class MovieService {
  getMovies() { }
}
```

Http



- Communicates with a back end server
- Common Http verbs: Get, Post, Put, etc
- Uses Observables

An **Observable** is an array of items that may come in the future

Demo

Data Access Service

Building an Angular Application

- From scratch
- ASP.NET Core + Angular 2 template for Visual Studio (http://blog.stevensanderson.com/2016/10/04/angular2-template-for-visual-studio/)
- Plunker (angular.io)
- QuickStart seed (github.com/angular/quickstart)
- Angular cli (github.com/angular/angular-cli)

Angular-cli

- Command line tool for generating, testing, and deploying an Angular application

```
ng new MovieHunter
ng generate component movie-list
ng build --prod
...
```

Key Concepts

- Great for building client-side web apps
- App = Components + Services + Module(s)
- Components
  - Template
  - Class
  - Metadata
- Angular Modules
- Routing
- Services
- Http and Observables

Thank You!

@deborahkurata

deborahk@insteptech.com

http://blogs.msmvps.com/deborahk

https://github.com/DeborahK/MovieHunter

http://bit.ly/Angular2-GettingStarted