Angular in 60-ish Minutes

Dan Wahlin

Dan Wahlin

https://blog.codewithdan.com

@DanWahlin

CODE WITH DAN

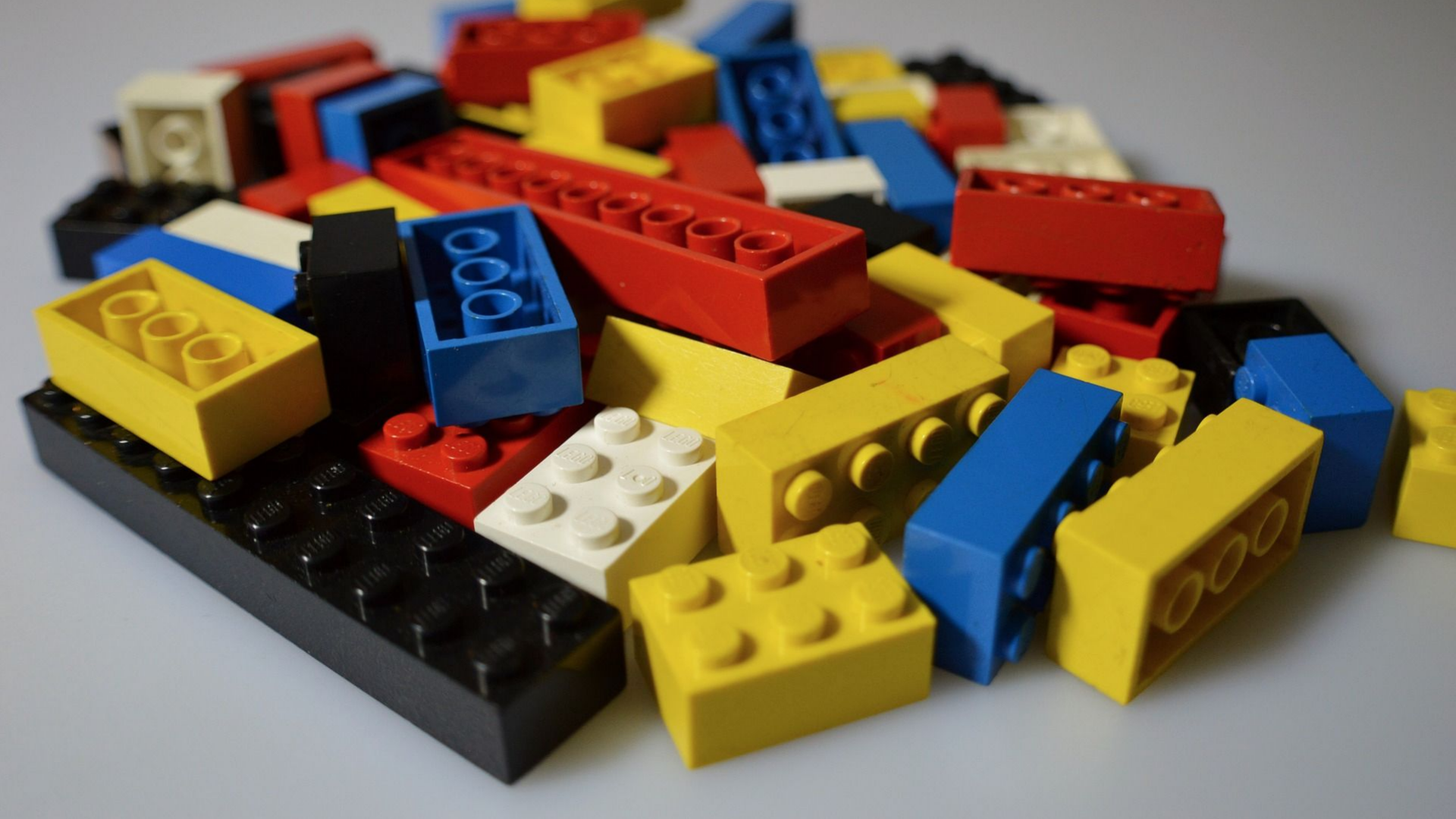
Get the Content

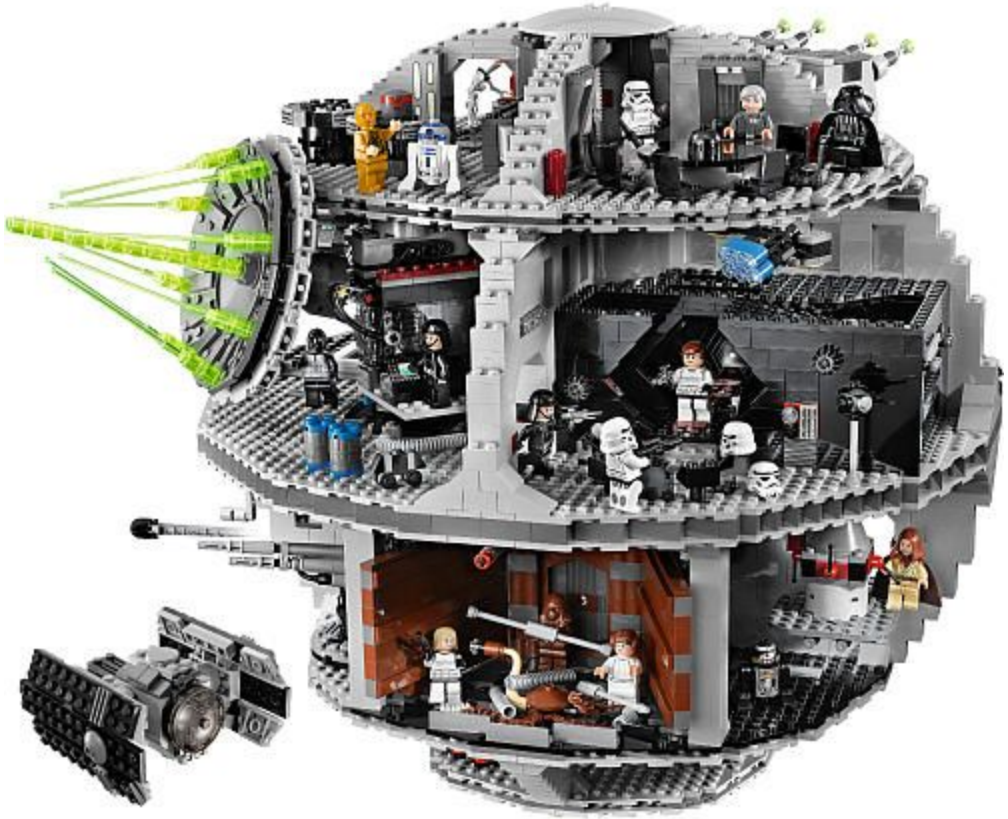
http://codewithdan.me/AngularIn60

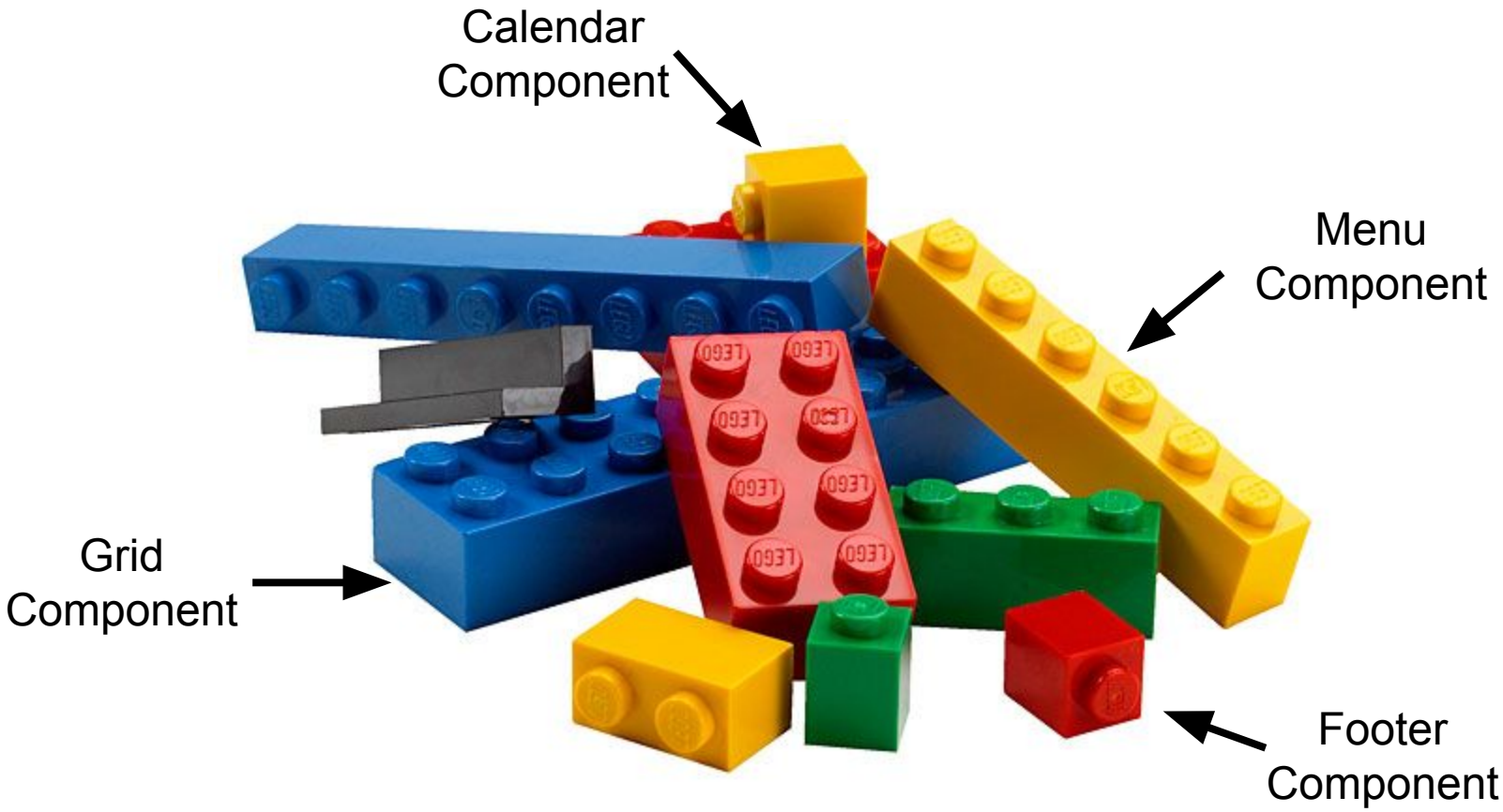
Agenda

- The Big Picture
- Getting Started
- Components
- Declarative Template Syntax
- Services
- Modules
- Routing

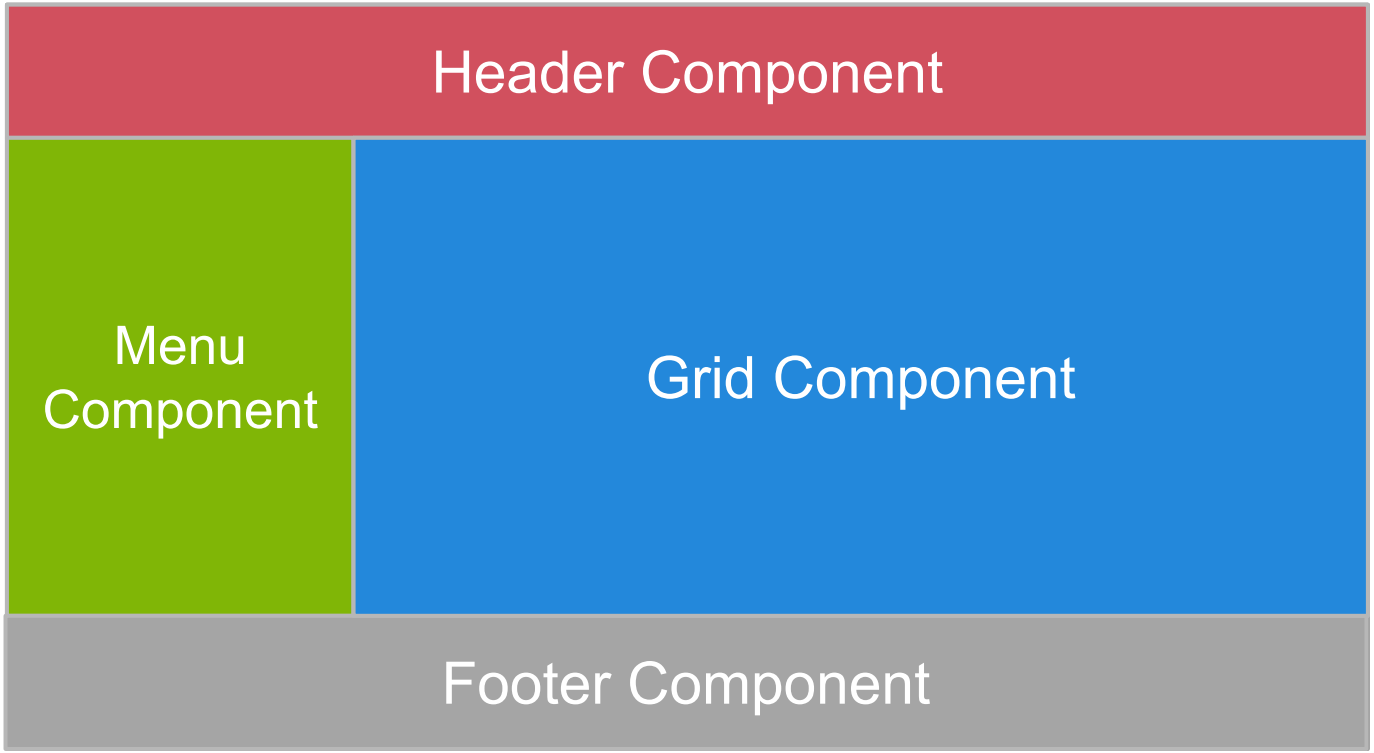
The Big Picture



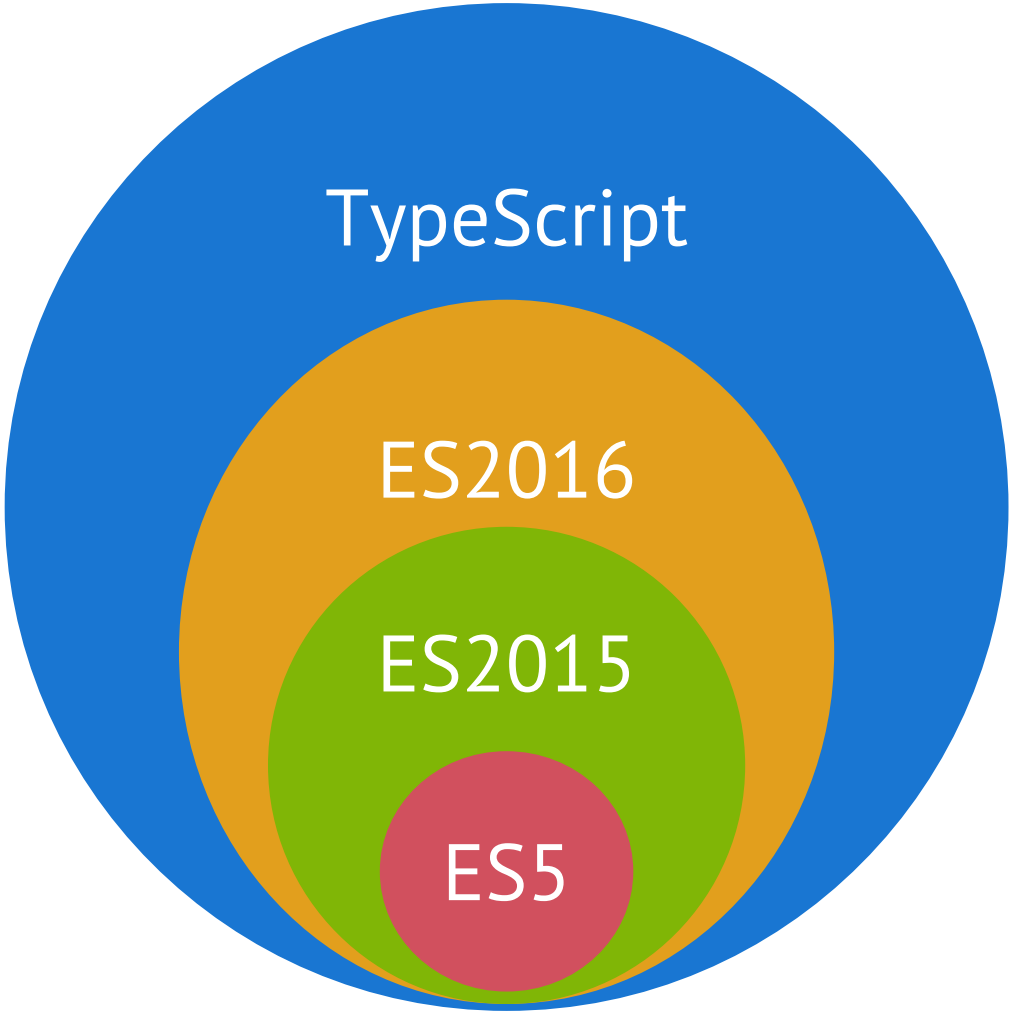



Calendar
Component
Menu
Component
Grid
Component
Footer
Component

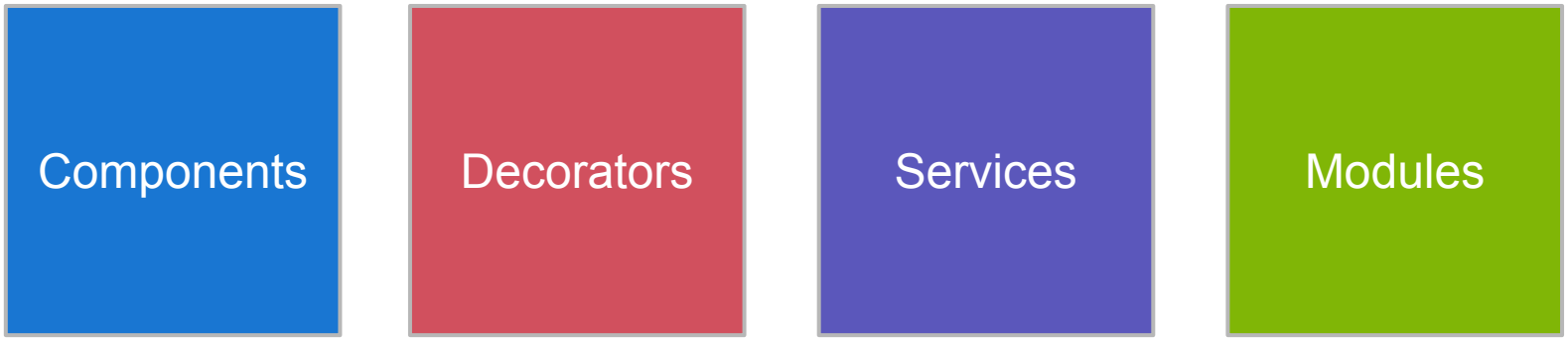
Building SPAs with Components

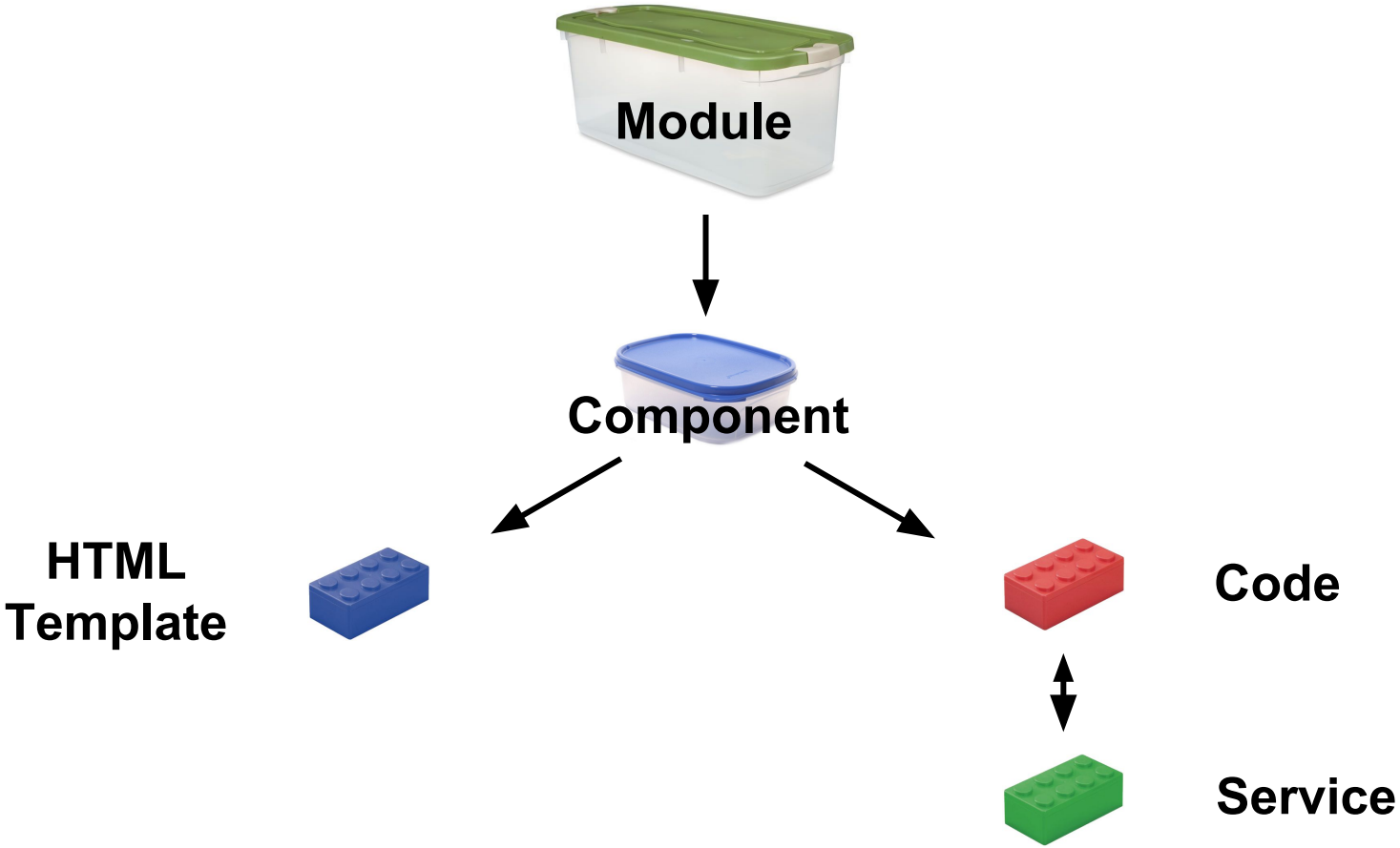


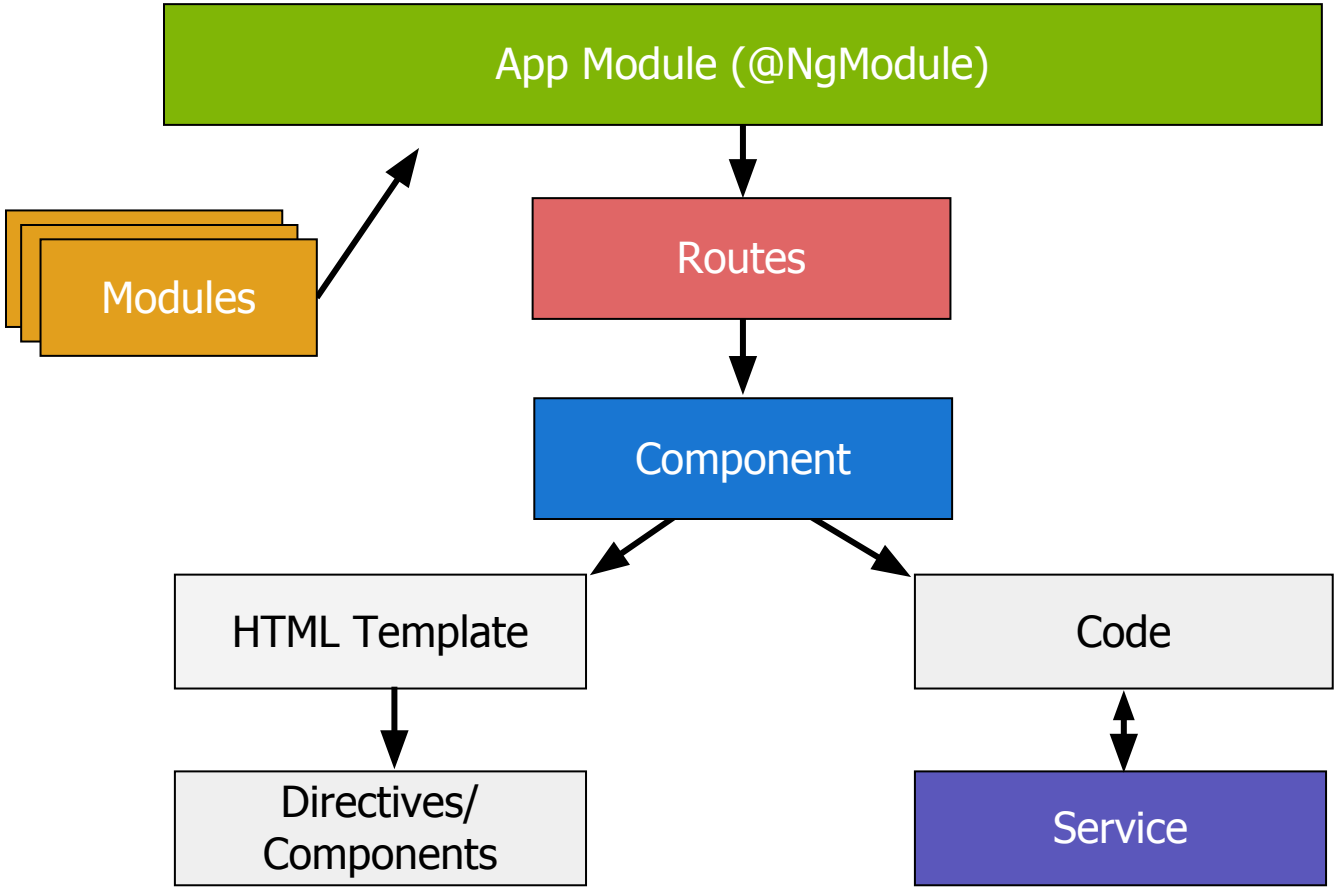
Getting Started


TypeScript
ES2016
ES2015
ES5

Angular Building Blocks



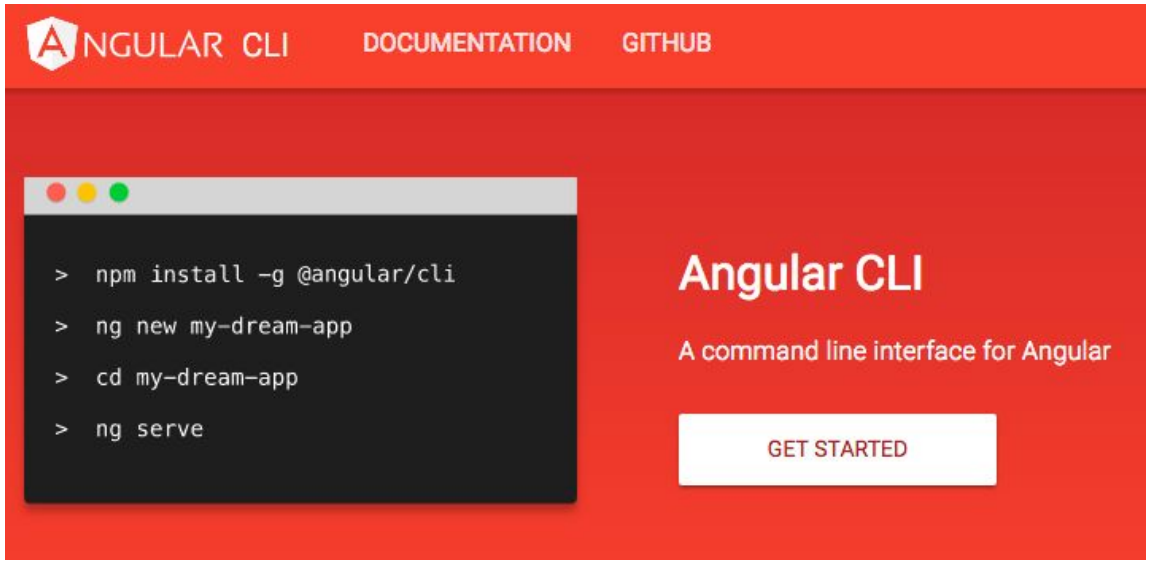

Module
Component
HTML
Template
Code
Service


App Module (@NgModule)
Modules
Routes
Component
HTML Template
Code
Directives/
Components
Service

Angular CLI

The Angular CLI

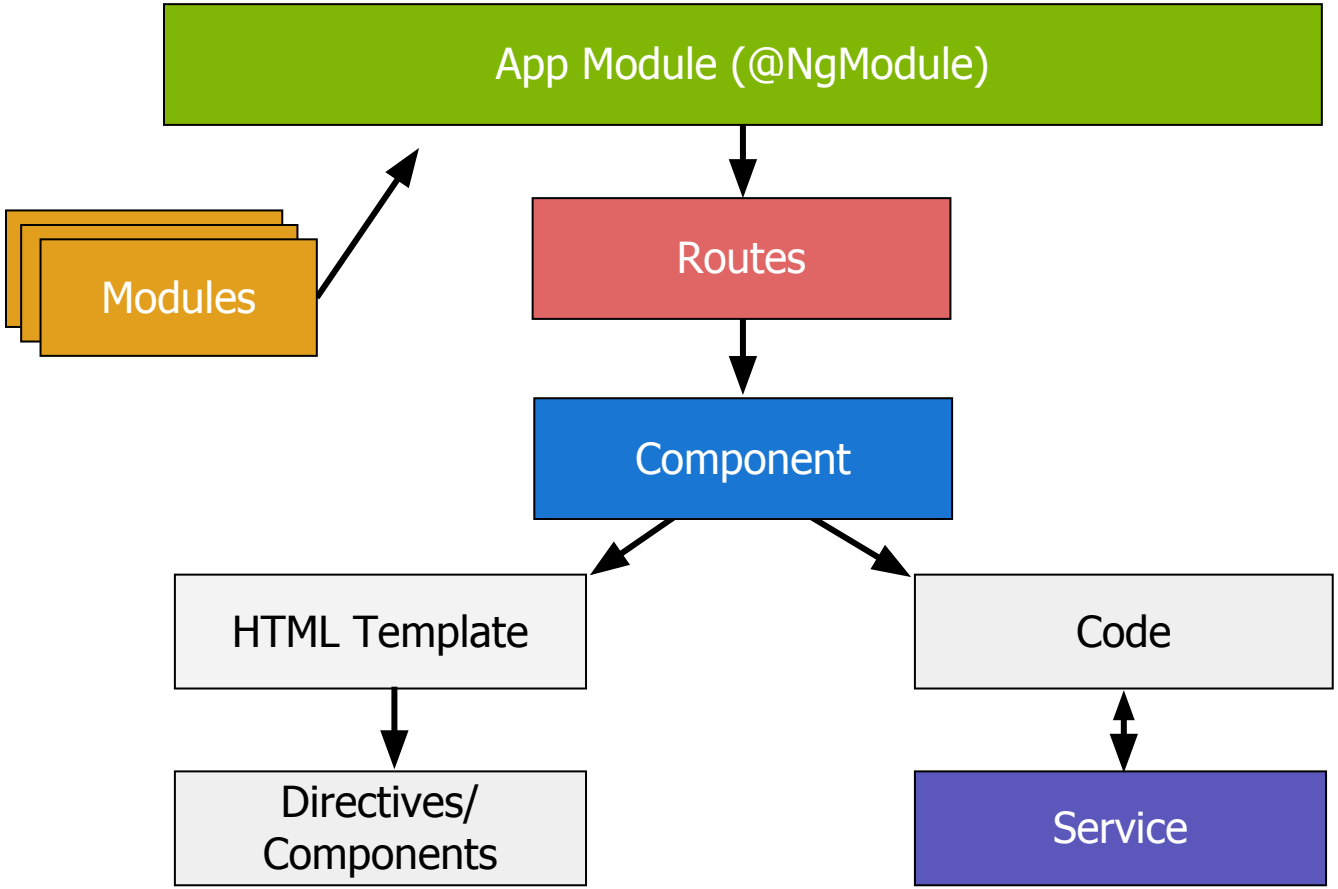
Angular applications can be generated using the Angular Command-Line Interface (CLI): https://cli.angular.io



Key Angular CLI Commands

```
ng --version
ng --help

ng new my-app-name

ng generate
    [component | directive | pipe | service | class | interface | enum | guard]

ng build

ng serve

ng lint

ng test
```

Components


App Module (@NgModule)
Modules
Routes
Component
HTML Template
Code
Directives/
Components
Service

What is a Component?

- Building blocks of Angular

- Made up of:

- Has a "selector": <my-customers></my-customers>

What's in Component Code?

imports

```
import { Component } from '@angular/core';
import { DataService } from '../services/data.service';
```

decorators

```
@Component({
  selector: 'my-customers',
  templateUrl: './customers.component.html'
})
```

class

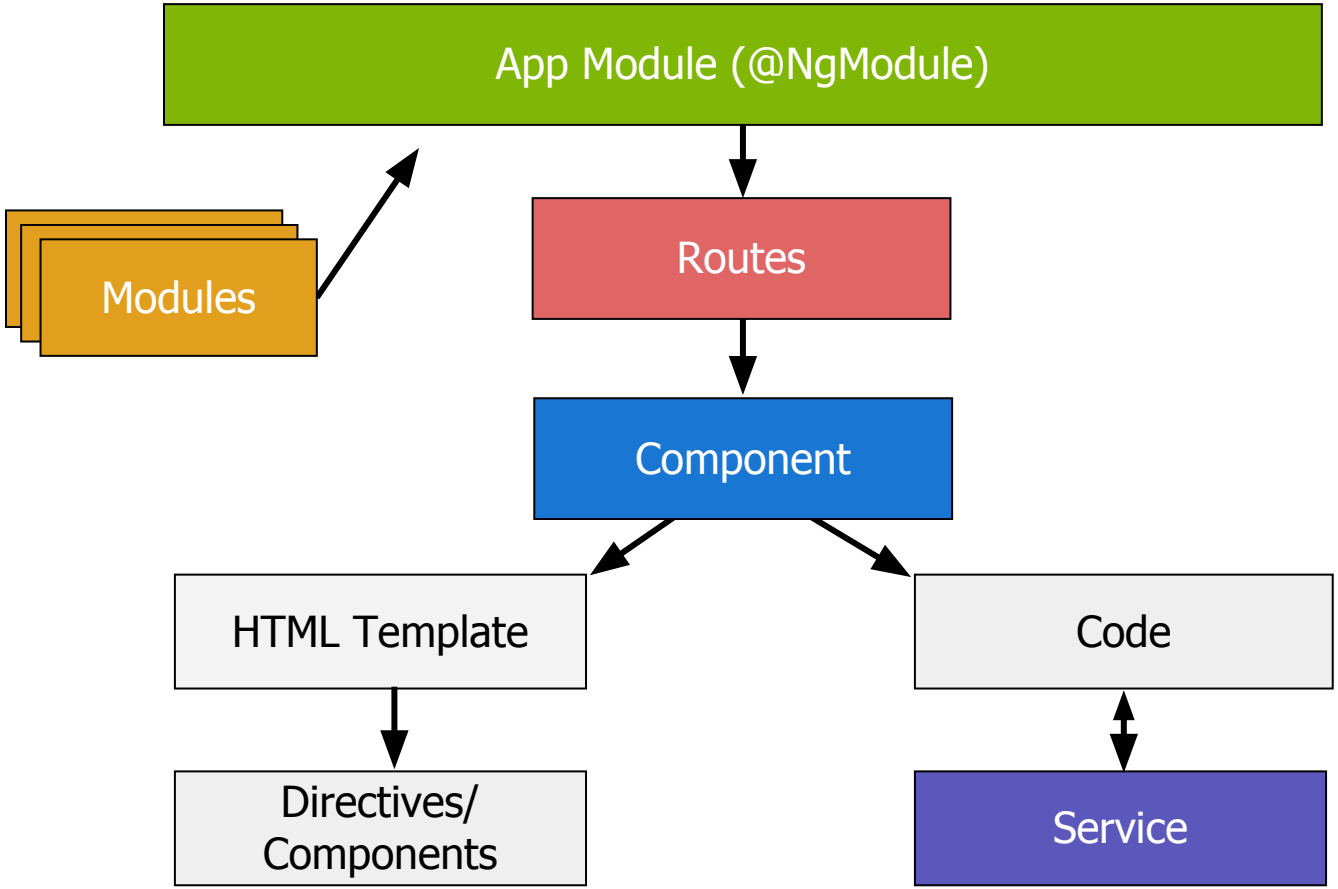
```
export class CustomersComponent {

}
```

http://codewithdan.me/angular-starter-template

Demo

Getting Started with Components

Component Template Syntax


App Module (@NgModule)
Modules
Routes
Component
HTML Template
Code
Directives/
Components
Service

Component Templates

Components rely on the **@Component** decorator to define the location of the template

customers.component.ts

<customers></customers>

```
@Component({
  selector: 'my-customers',
  templateUrl: './customers.component.html'
})
export class CustomersComponent {
  constructor(private dataService: DataService) { }
}
```

Data Binding Syntax

Bind to DOM properties using **[property]** or **bind-property**
Bind to DOM events using **(event)** or **on-property**
Create "two-way" bindings using **[(property)]** or **bindon-property**

Bind to DOM property

```
<button [disabled]="!isEnabled" (click)="save()">Save</button>

<div [hidden]="!isVisible"
     [class.active]="isActive">...</div>

<input type="text" [(ngModel)]="name" />
```

Built-In Directives

Angular has built-in directives such as **ngFor** and **ngIf**

```
<tr *ngFor="let customer of customers">
    <td>{{ customer.firstName }}</td>
    <td>{{ customer.lastName }}</td>
</tr>

<div *ngIf="customer">{{ customer.details }}</div>
```

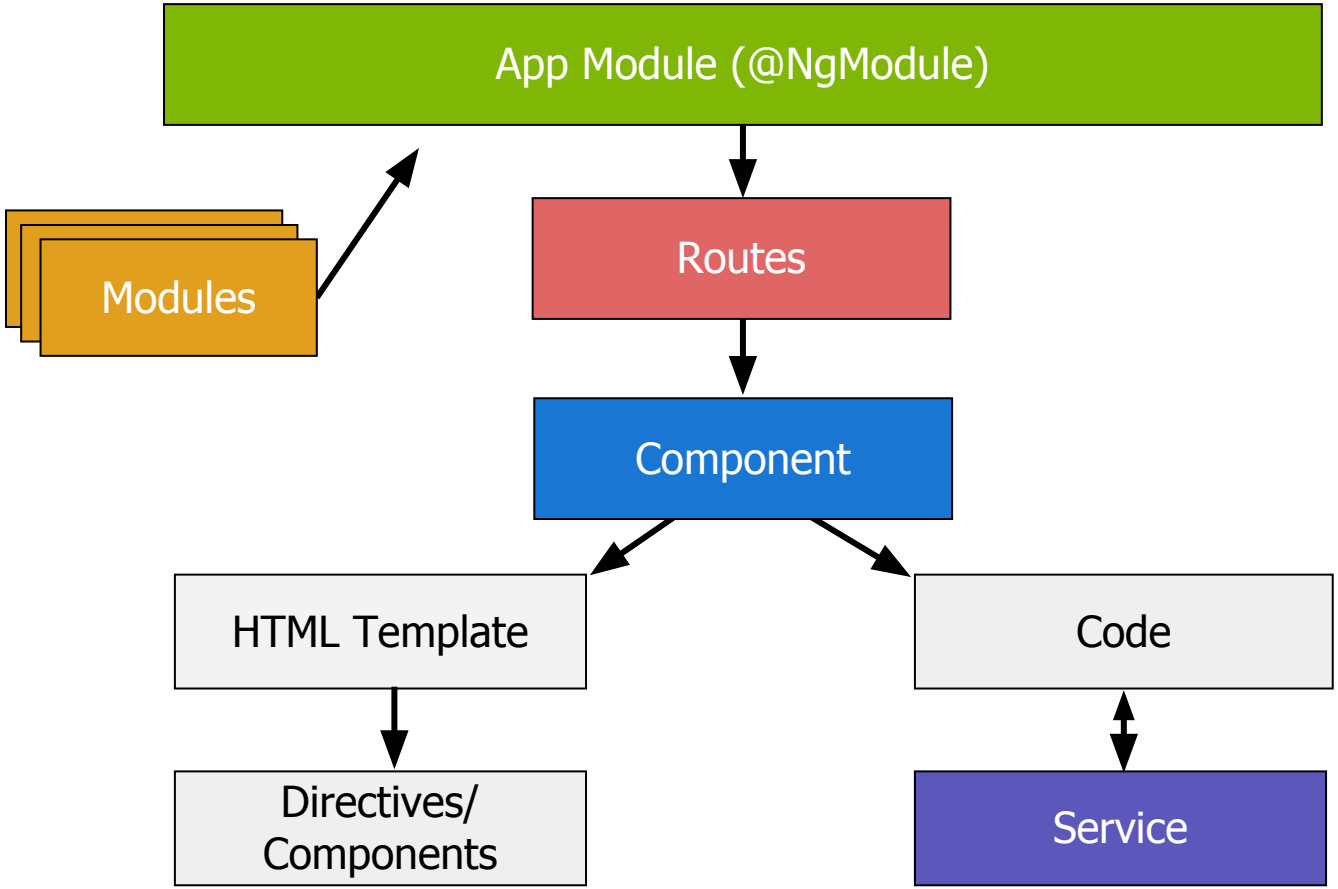
Built-In Pipes

- Angular apps can use **Pipes** to format data
- uppercase, lowercase, slice, date, currency, json

```
{{ customer.orderTotal | currency:'USD':true }}
```

$9.99

Services


App Module (@NgModule)
Modules
Routes
Component
HTML Template
Code
Directives/
Components
Service

Creating a Service Class

Services are classes that perform reusable functionality (business rules, calculations, Ajax calls, etc.)

data.service.ts

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

@Injectable()
export class DataService {
   constructor(private http: Http) { }
}
```

Injected at runtime

Using Http to Call RESTful Services

- RESTful services can be called using Http (@angular/http module)
- Can use Observables or Promises for async operations

data.service.ts

```
@Injectable()
export class DataService {
   constructor(private http: Http) { }

   getCustomers() : Observable<Customer[]> {
       return this.http.get('api/customers')
                       .map((response: Response) => response.json())
                       .catch(this.handleError);
   }
}
```

Map response to callback

Injecting a Service into a Component

Services can be injected into components at runtime

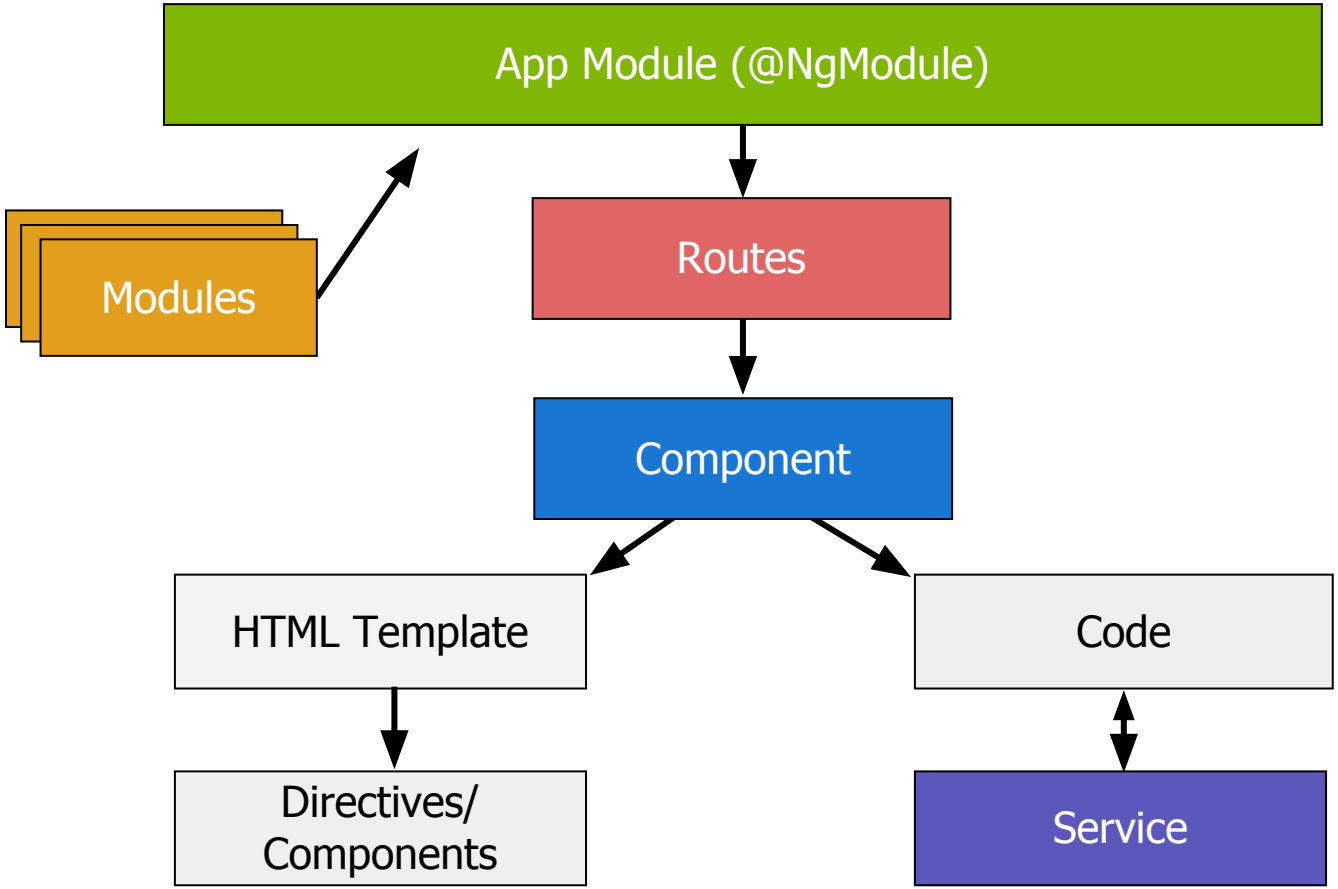
customers.component.ts

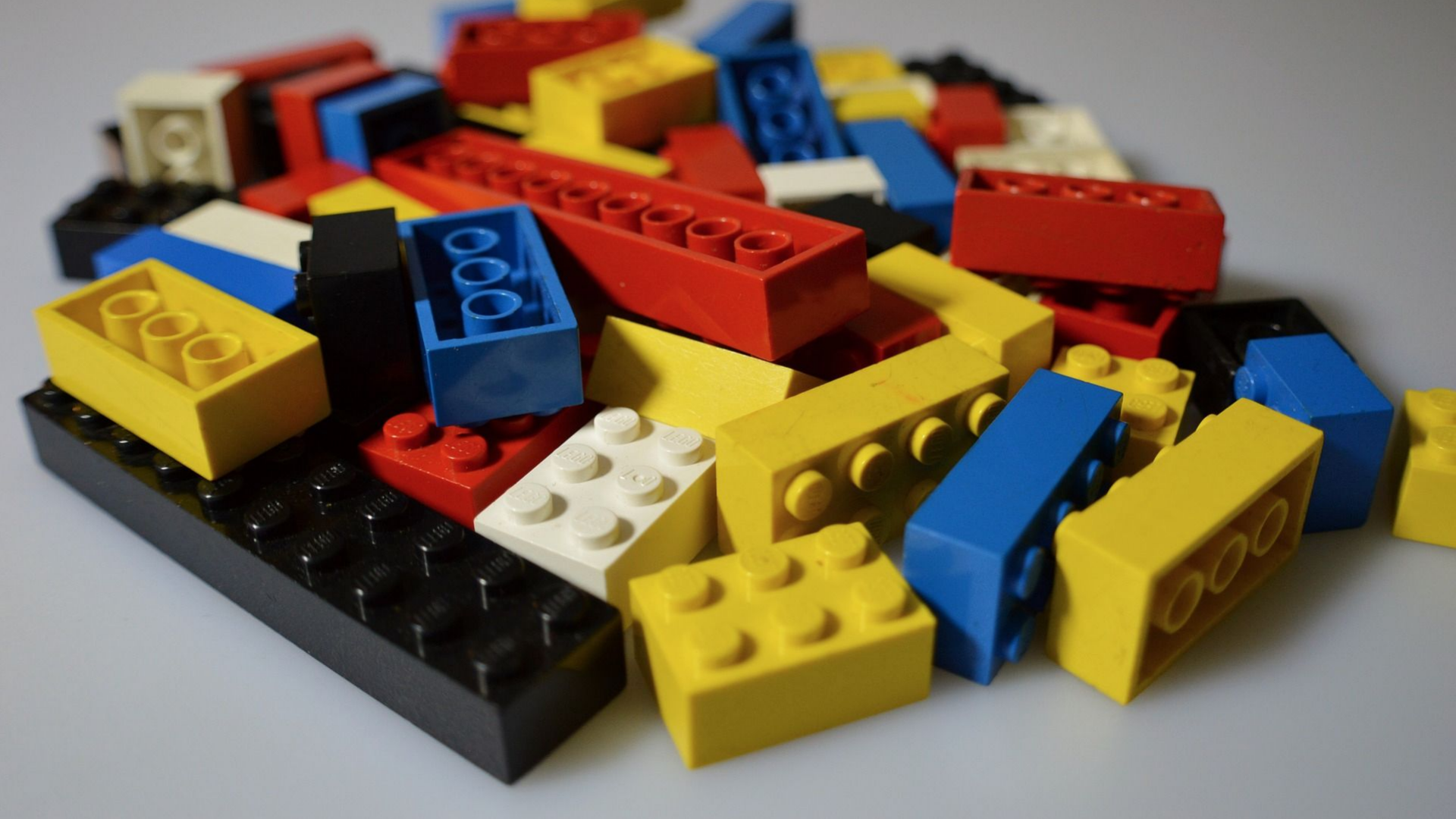
```
@Component({
   selector: 'my-customers',
   templateUrl: './customers.component.html'
})
export class CustomersComponent {
   customers: Customer[];
   constructor(private dataService: DataService) { }
   ngOnInit() {
     this.dataService.getCustomers()
           .subscribe((customers: Customer[]) => {
               this.customers = customers;
           });
   }
}
```

Inject data service

Subscribe to observable

Modules


App Module (@NgModule)
Modules
Routes
Component
HTML Template
Code
Directives/
Components
Service



What are Modules?

- Containers for code

- Angular apps use different types of modules:
  - ES2015 modules - Individual files loaded with a module loader
  - Angular modules - Created using @NgModule

ES2015 Modules: Exporting

- ES2015 modules allow functionality to be imported/exported
- A file can act as an ES2015 module that **exports** features

data.service.ts

```
export class DataService {
    ...
}
```

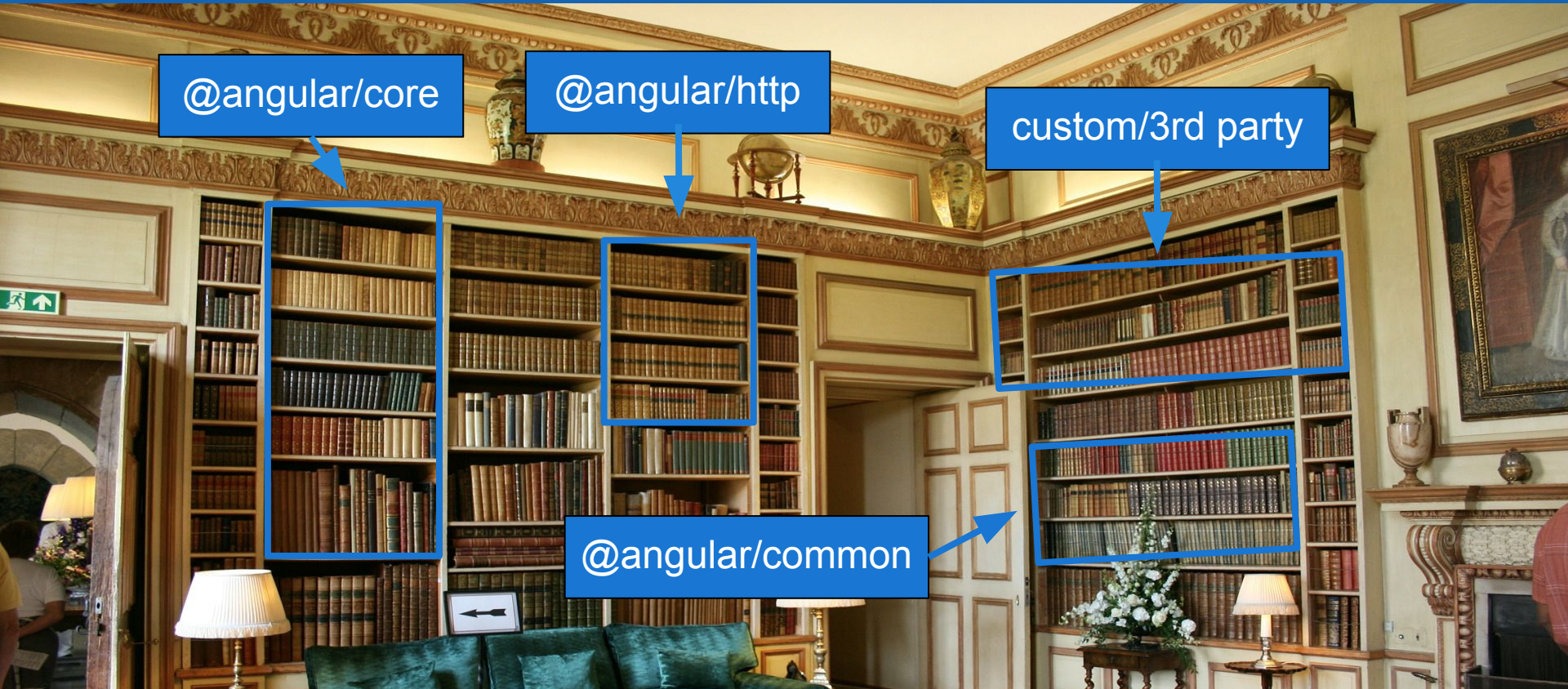
ES2015 Modules: Importing

Modules can be imported using the **import** keyword

customers.component.ts

```
import { Component  } from '@angular/core';
import { DataService } from '../shared/services/data.service';
...
```

Angular Modules



Creating an Angular Module

- Angular organizes components (and more) into modules
- Relies on the **@NgModule** decorator

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
Import { DataService } from './shared/data.service';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  providers:    [ DataService ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Modules and Bootstrapping

index.html

```
<html>
<body>

  <app-component></app-component>

  <script>
    System.import('app');
  </script>

</body>
</html>
```

1

4

Bootstrap

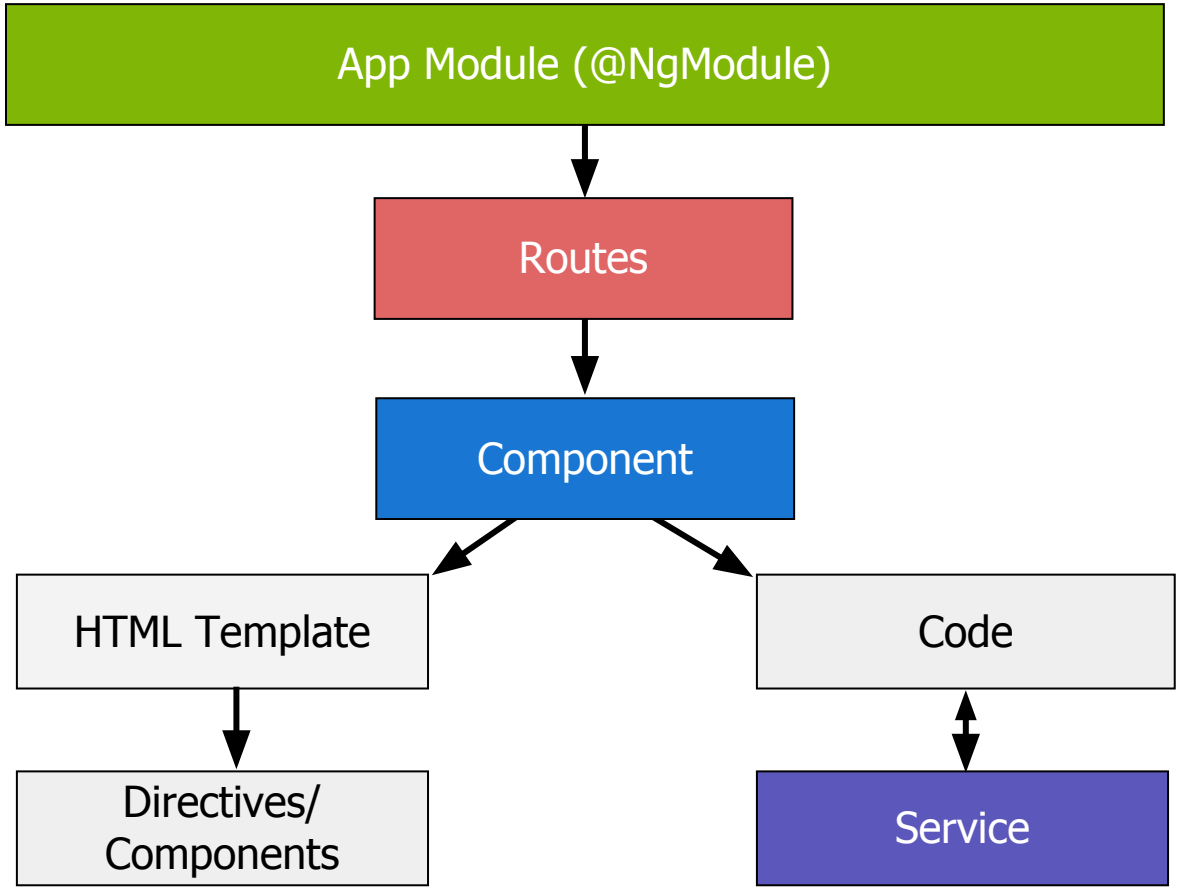
3

```
@NgModule({
  declarations: [AppComponent]
  bootstrap: [AppComponent ]
})
export class AppModule { }
```

2

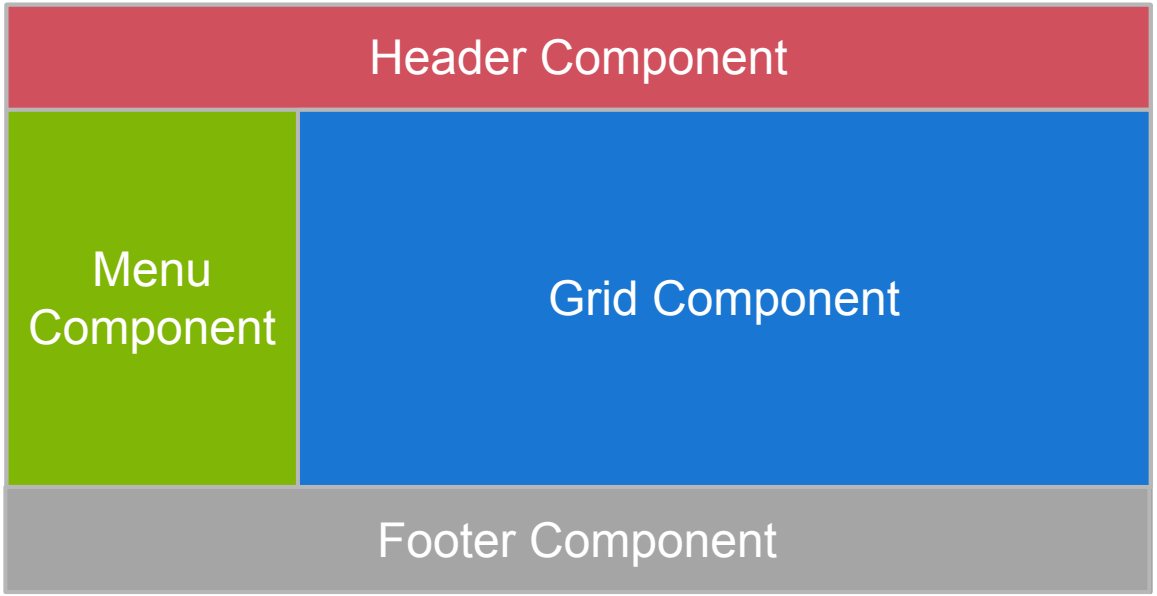
```
@Component({
  selector: 'app-component'
})
class AppComponent { }
```

Routing


App Module (@NgModule)
Routes
Component
HTML Template
Code
Directives/
Components
Service

Routes and Components

Components can be loaded into a <router-outlet></router-outlet>



Using <router-outlet>

<router-outlet> acts as an "X marks the spot" for components

app.component.ts

```
@Component({
  selector: 'app-component',
  template: '<router-outlet></router-outlet>'
})
export class AppComponent {

}
```

Routing will load components here

Router Configuration

Define routes using RouterModule in @angular/router

```
import { RouterModule, Routes } from '@angular/router';
import { CustomersComponent } from './customers/customers.component';
import { OrdersComponent } from './orders/orders.component';

const app_routes: Routes = [
  { path: 'customers', component: CustomersComponent},
  { path: 'orders/:customerId', component: OrdersComponent }
];

export const app_routing = RouterModule.forRoot(app_routes);
```

app_routing will be imported into root module

Adding Routes into a Module

Routes are passed into a module

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { app_routing } from './app.routes';
import { AppComponent } from './app.component';
...

@NgModule({
  imports: [ BrowserModule, app_routing ],
  declarations: [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Add routes into module

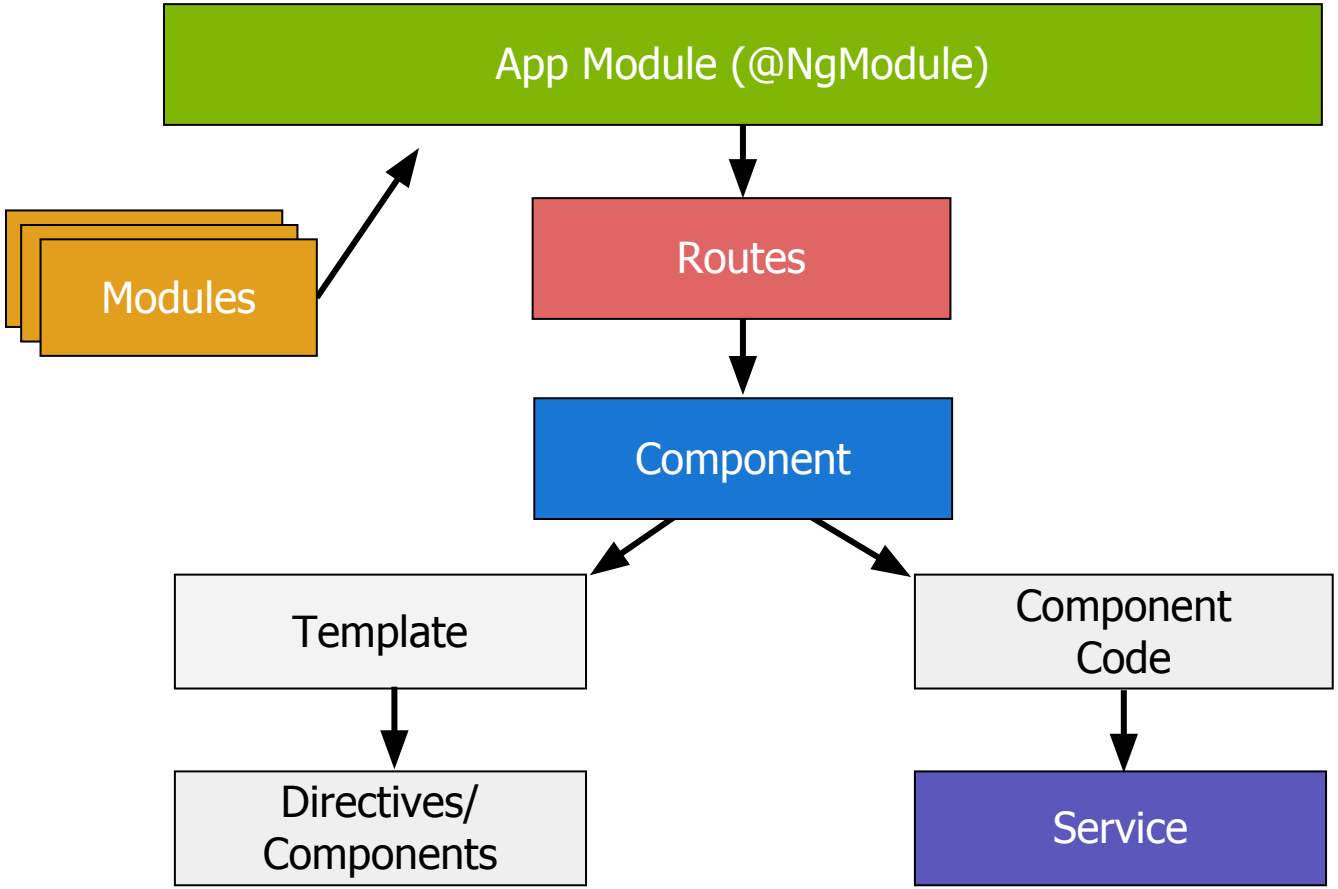
The RouterLink Directive

Navigate between components using the RouterLink directive

```
<div class="card-header">

  <a routerLink="/customers">View Customers</a>

  <a [routerLink]="['orders', customer.id]">
     {{ customer.firstName }}
     {{ customer.lastName }}
  </a>

</div>
```

Link to "customers" route

Link to "orders" route and pass route parameter


App Module (@NgModule)
Modules
Routes
Component
Template
Component Code
Directives/ Components
Service

Key Links

- Angular Site
  - https://angular.io

- Angular API Docs
  - https://angular.io/docs/ts/latest/api

- Angular Github content:
  - https://github.com/danwahlin

http://codewithdan.me/angular-10-projects

http://codewithdan.me/angular-snippets

Code

Sample Code and Snippets

Thanks!

http://codewithdan.me/AngularIn60

Developer Training

Angular, Node, TypeScript, JavaScript , C#, ASP.NET Core, Docker & more at your company or online!

http://codewithdan.com

CODE WITH DAN