Up And Running With Angular in 45 Minutes

Justin James

LET YOUR NERD BE HEARD

Speaking to More Than 1s and 0s

Angular

Framework for building client side applications using HTML, CSS, and TypeScript

Why AngularJS

Expressive HTML

Powerful Data Binding

Modular

Streamlined Http Calls

Why Angular 2+?

Modern

Speed

Simplified API

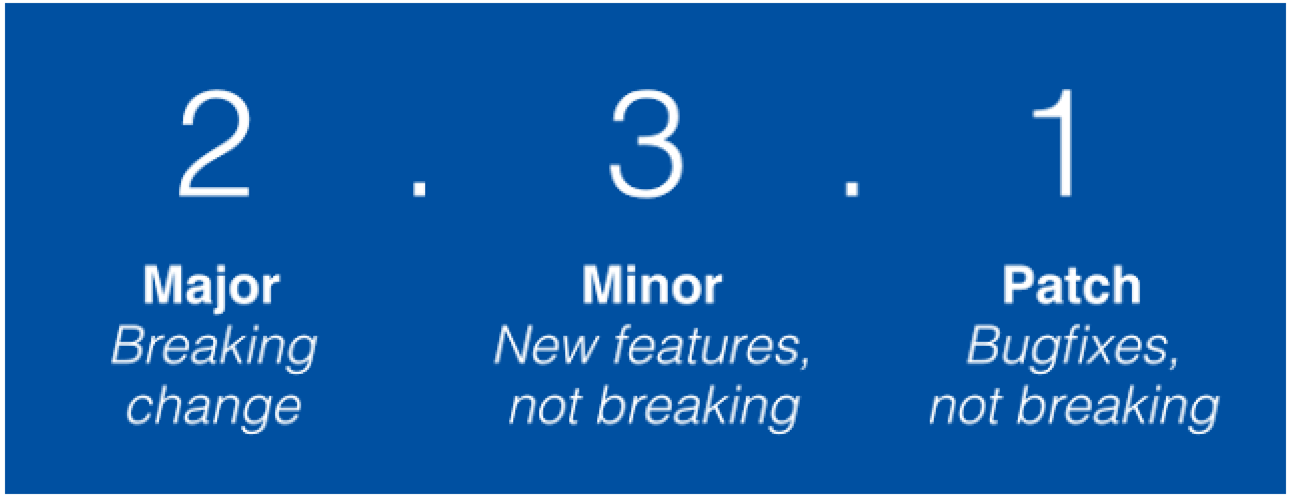
Enhanced Productivity

Justin James

Amplify Your Voice, Message, and Authority Through Video

Entrepreneur | Podcaster | YouTuber | Speaker

Angular Release Plan



2 . 3 . 1

Major
Breaking change

Minor
New features, not breaking

Patch
Bugfixes, not breaking

~6 Months Monthly Weekly

Setup

Development Environment

Global Setup

Node JS (10+ LTS)

http://nodejs.org

NPM Global Packages

```
npm install -g @angular/cli
```

Creating Applications

Create New Project

```
ng  new AppName --style scss --routing
```

Generate Stuff

```
ng generate [TYPE] [NAME]
```

Dev Server

```
ng serve
```

Linting

```
ng lint
```

Building

```
ng build --prod
```

Editor

Visual Studio Code

Extensions:

- Angular Language Service
- TSLint (Microsoft Version)

Typescript Essentials

more than enough to be really dangerous

TypeScript

Typed Superset of JavaScript

Features

Supports JavaScript

Static Typing

Encapsulation

Intellisense & Syntax Checking

Typing....

Dynamic

JavaScript, Python, Ruby, PHP

```
var number = 5;
number = "Hello!";
// works great!
```

Static

C, C++, C#, Java

```
int number = 5;
number = "Hello!";
// Compiler Error
```

```
var items = getItems();
var goSportsTeam = true;
var super = 'mario';
```

Most JavaScript is already valid TypeScript

~~some-javascript.js~~

some-typescript.ts

```
status: string;
havingFun: boolean;
daysTillVacation: number;
```

Types

boolean, number, string
array, enum
any, void

```
status = 'TS Rocks!';
havingFun = true;
daysTillVacation = 60;
```

Types

boolean, number, string
array, enum
any, void

```
myNumbers: number[] = [170, 2.6, 2245,400];
// Or...
myNumbers: Array<number> = [170, 2.6, 2245, 400];
```

Arrays

List of values

Compiler Error:

Argument of type 'string' is not assignable to parameter of type 'number'

```
myNumbers: number[] = [];
myNumbers.push('700');
```

Arrays

Enforce types for array content

```
initSomething() : void {
   // do something
}
```

Void

A function that returns nothing

```
initSomething() : void {
   // do something
}
```

Void

A function that returns nothing

```
initSomething() : void {
   // do something
}
let pointless = initSomething();
```

Void

A function that returns nothing

```
initSomething() : void {
   // do something
}
let pointless = initSomething(); COMPILER ERROR
```

Void

A function that returns nothing

```
let foo: any;
foo = 'Hello';
foo = true;
foo = 42;
```

THINK TWICE BEFORE USING!

Any

Restores basic JavaScript dynamic typing behavior

```
export class Greeter {
    greeting: string;
    constructor(message: string) {
        this.greeting = message;
    }
    greet() {
        return `Hello ${this.greeting}`;
    }
}
```

Classes

Encapsulate functionality and data for an object

```
let greeter = new Greeter('world').greet();
```

Classes

Encapsulate functionality and data for an object

```
interface Label {
  value: string;
}
```

Interfaces

Powerful way of defining contracts

```
interface Label {
  value: string;
}

class label implements Label {
  value: string = 'label value';
}
```

Interfaces

Powerful way of defining contracts

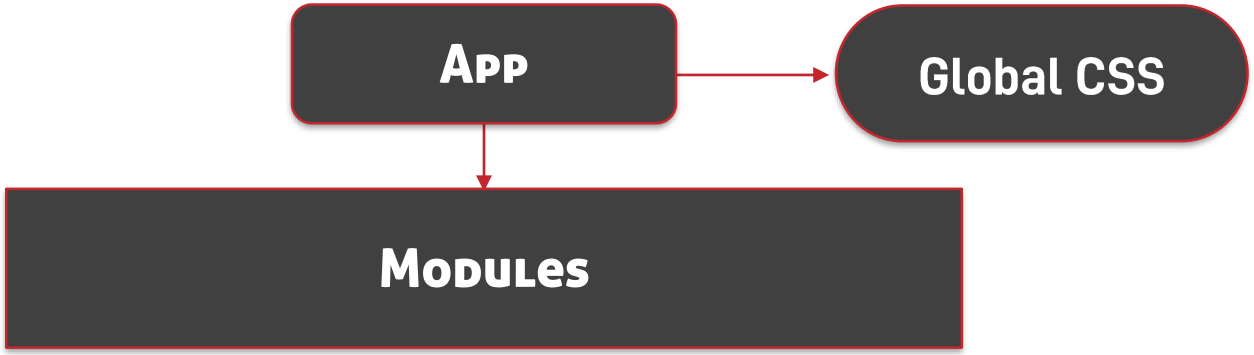
```
interface Label {
  value: string;
}

label: Label = {
  value: 'label value';
}
```

Interfaces

Powerful way of defining contracts

Angular Essentials

Just enough to be really dangerous

App
Global CSS
Modules

- ng generate module <Name>
- ng g m <Name>

Module - AppModule

app.module.ts

```
@NgModule({
  declarations: [
    AppComponent, LoginComponent, SignupComponent
  ],
  imports: [
    BrowserModule, FormsModule, ReactiveFormsModule,
    HttpClientModule, AppRoutingModule
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Module - AppModule

app.module.ts

```
@NgModule({
  declarations: [
    AppComponent, LoginComponent, SignupComponent
  ],
  imports: [
    BrowserModule, FormsModule, ReactiveFormsModule,
    HttpClientModule, AppRoutingModule
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Module - AppModule

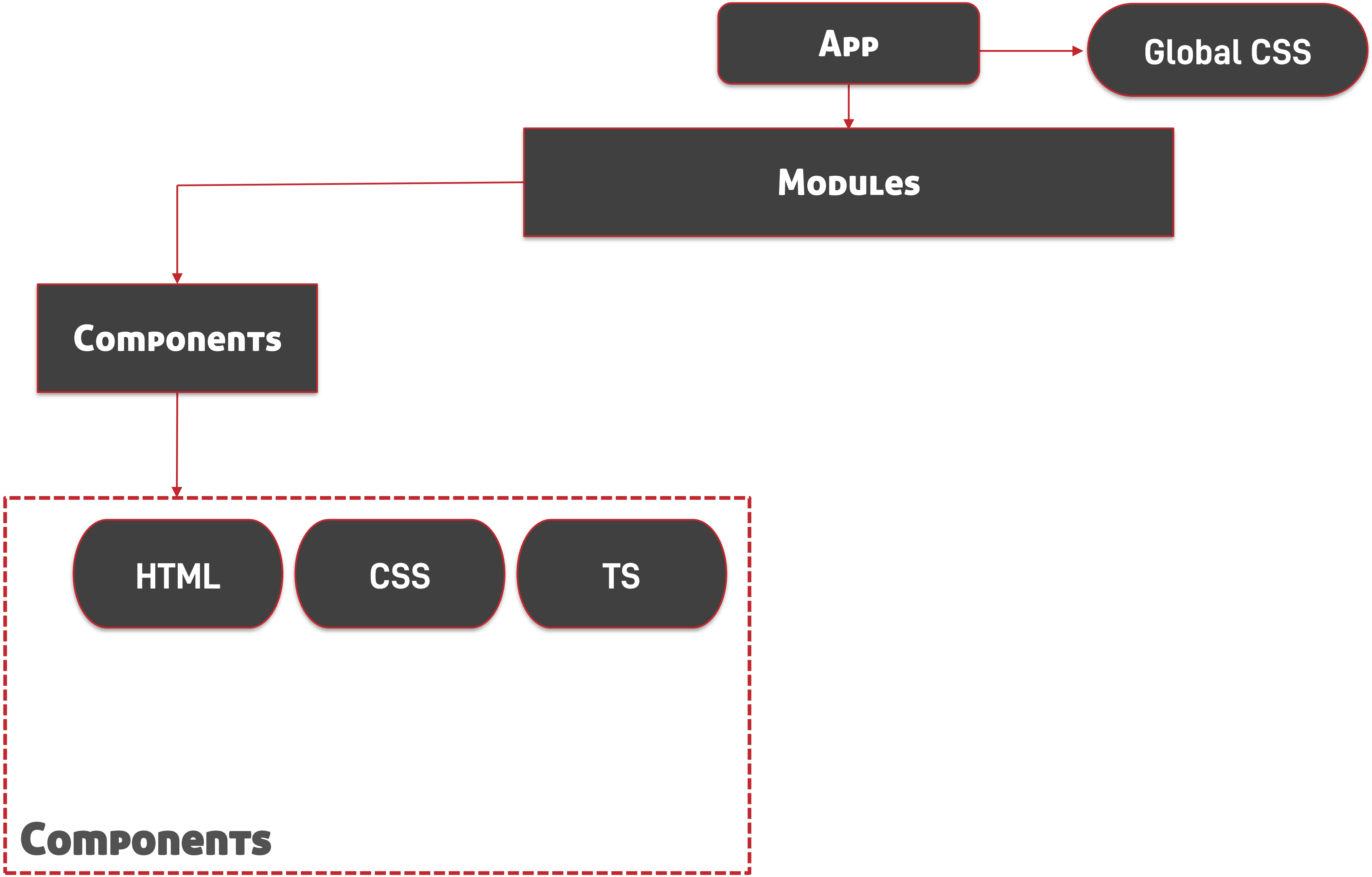
app.module.ts

```
@NgModule({
  declarations: [
    AppComponent, LoginComponent, SignupComponent
  ],
  imports: [
    BrowserModule, FormsModule, ReactiveFormsModule,
    HttpClientModule, AppRoutingModule
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Module - AppModule

app.module.ts

```
@NgModule({
  declarations: [
    AppComponent, LoginComponent, SignupComponent
  ],
  imports: [
    BrowserModule, FormsModule, ReactiveFormsModule,
    HttpClientModule, AppRoutingModule
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```


App
Global CSS
Modules
Components
HTML
CSS
TS
Components

```
➢ ng generate component <Name>
➢ ng g c <Name>
```

AppComponent

app.component.ts

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

@Component({
    selector: 'app-root',
    templateUrl: './app.component.html',
    styleUrls: ['./app.component.scss'],
})
export class AppComponent {
    title = 'app works!';
}
```

AppComponent

app.component.ts

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.scss'],
})
export class AppComponent {
   title = 'app works!';
}
```

AppComponent

app.component.ts

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.scss'],
})
export class AppComponent {
   title = 'app works!';
}
```

AppComponent

app.component.ts

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.scss'],
})
export class AppComponent {
   title = 'app works!';
}
```

AppComponent

app.component.ts

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.scss'],
})
export class AppComponent {
   title = 'app works!';
}
```

AppComponent

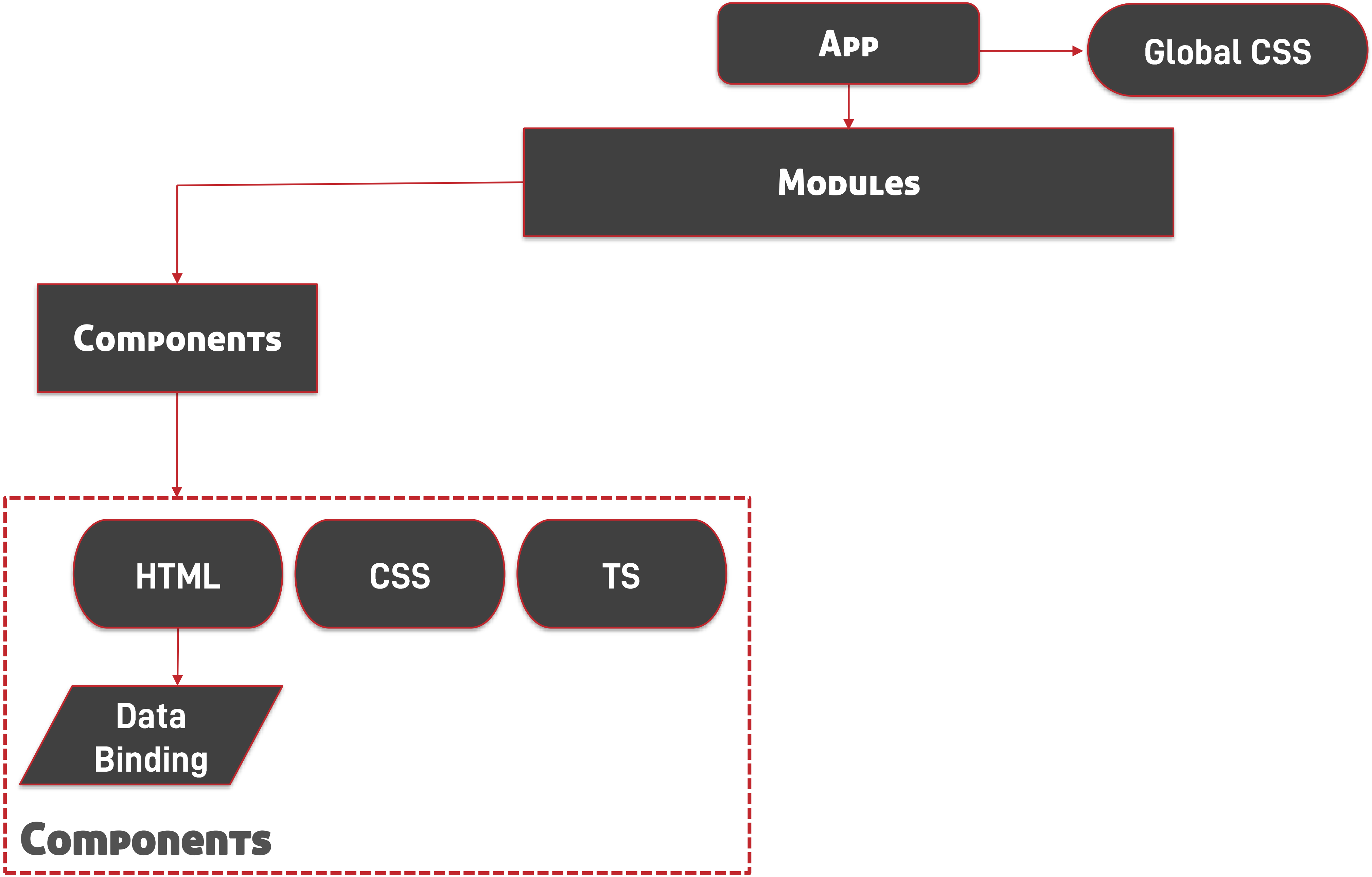
app.component.ts

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.scss'],
})
export class AppComponent {
   title = 'app works!';
}
```

AppComponent

app.component.ts

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.scss'],
})
export class AppComponent {
   title = 'app works!';
}
```



App

Global CSS

Modules

Components

HTML

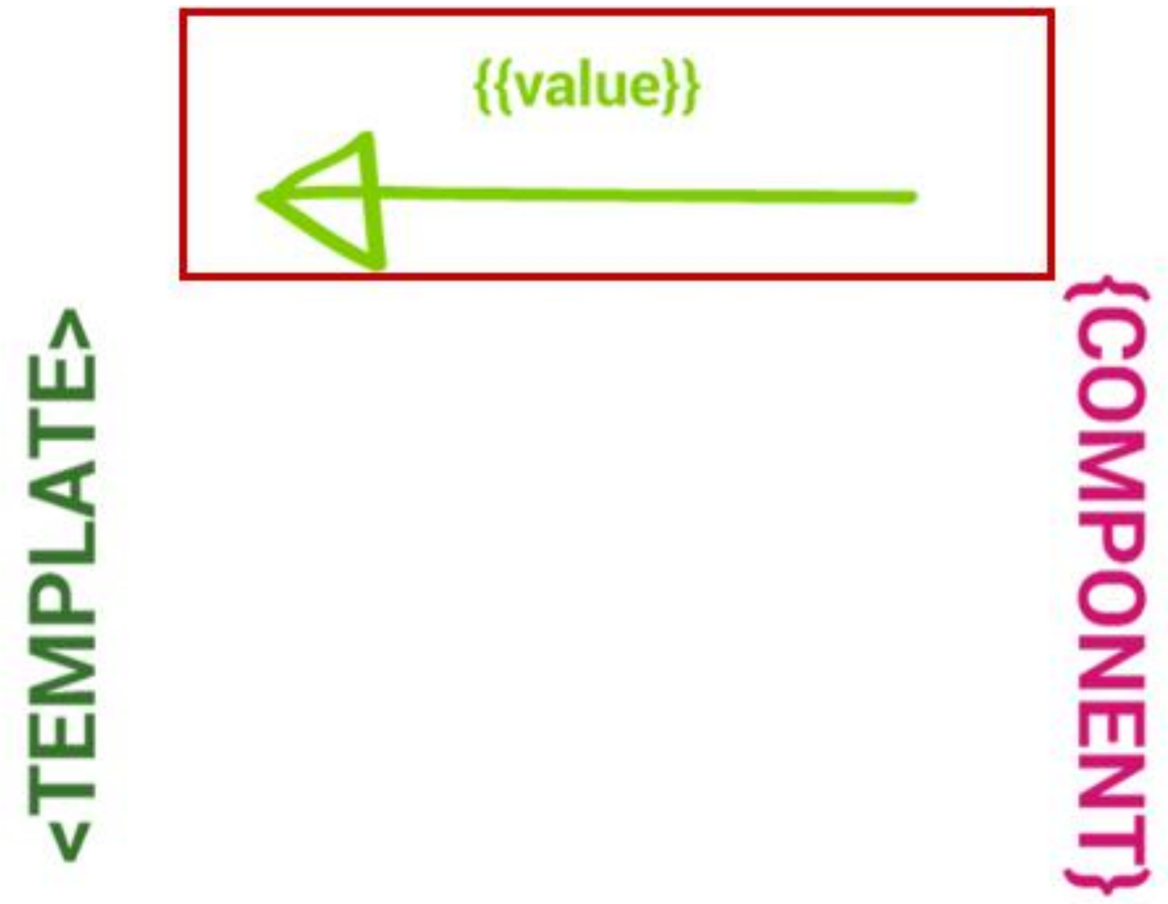
CSS

TS

Data Binding

Components

Data Binding



{{value}}

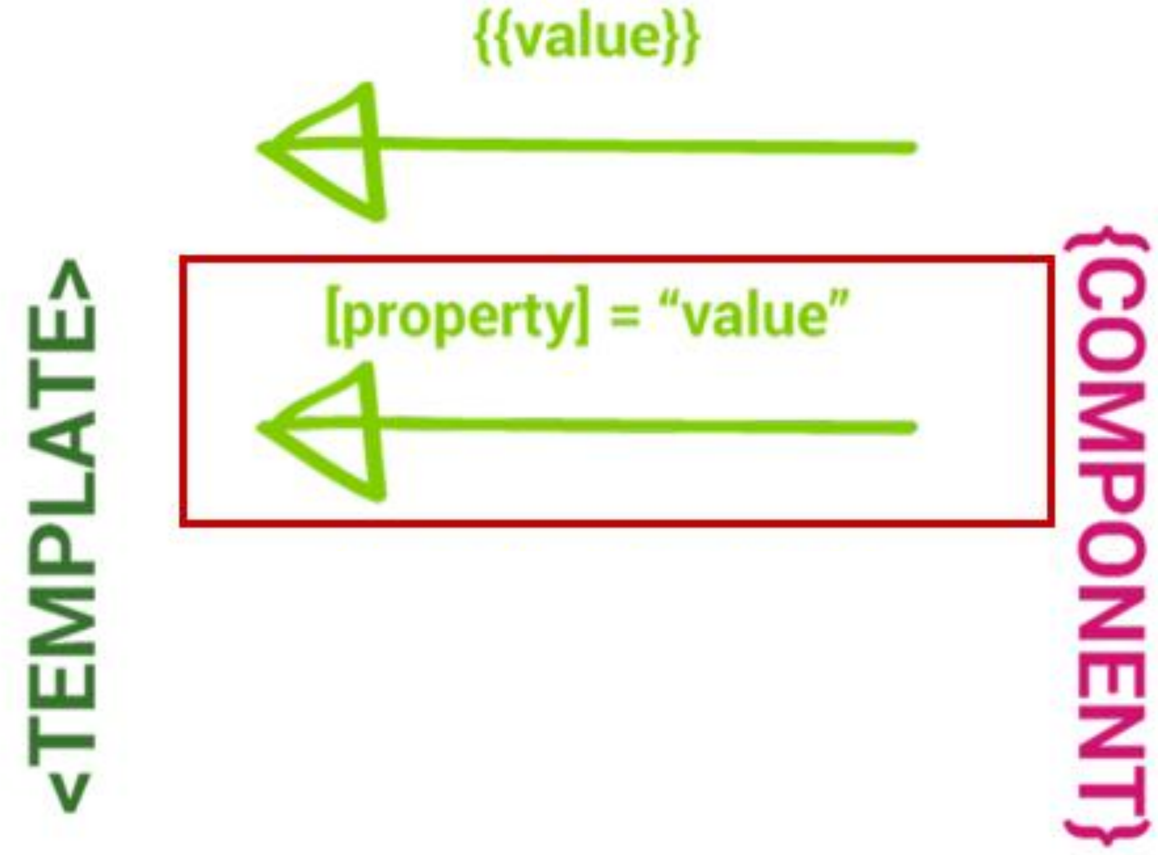
<TEMPLATE>

{COMPONENT}

Data Binding

```
<h3>{{ title }}</h3>
```

Data Binding



{{value}}

[property] = “value”

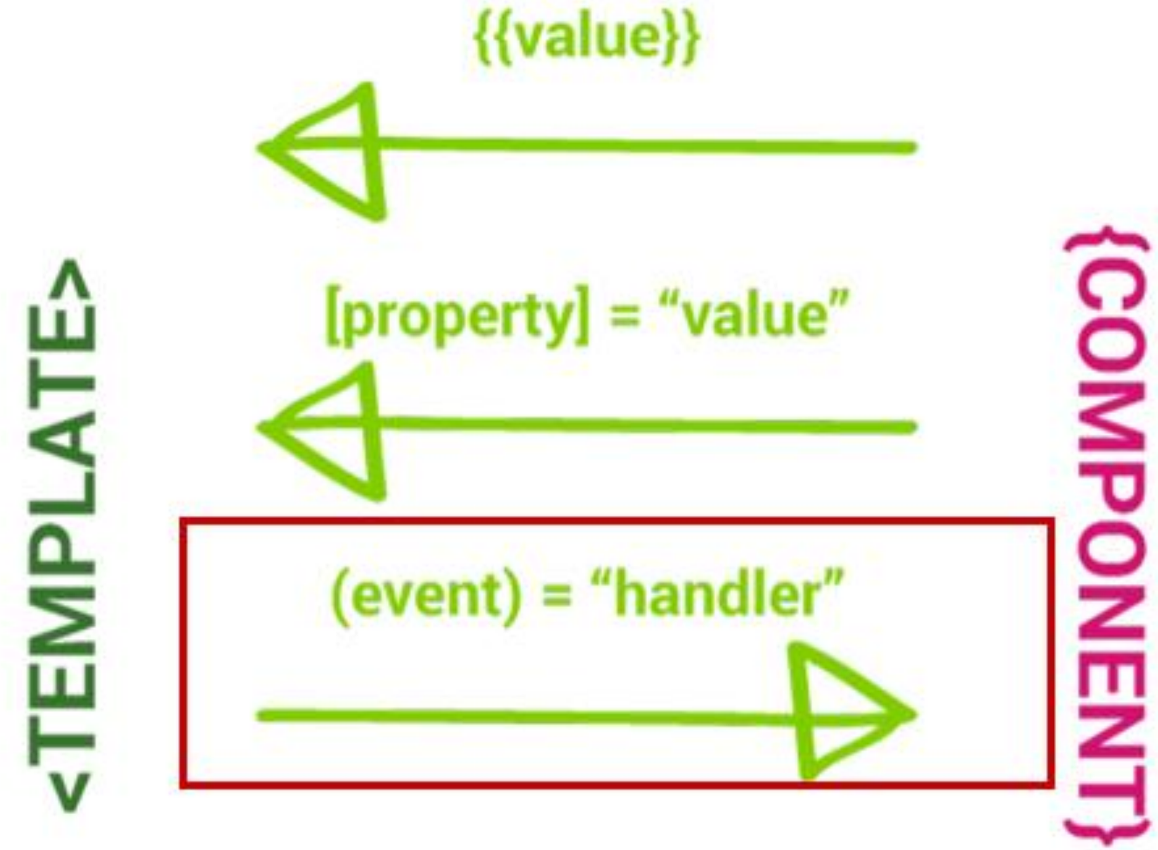
<TEMPLATE>

{COMPONENT}

Data Binding

```
<h3 [innerHTML]="title"></h3>

<img [src]="logo">
```

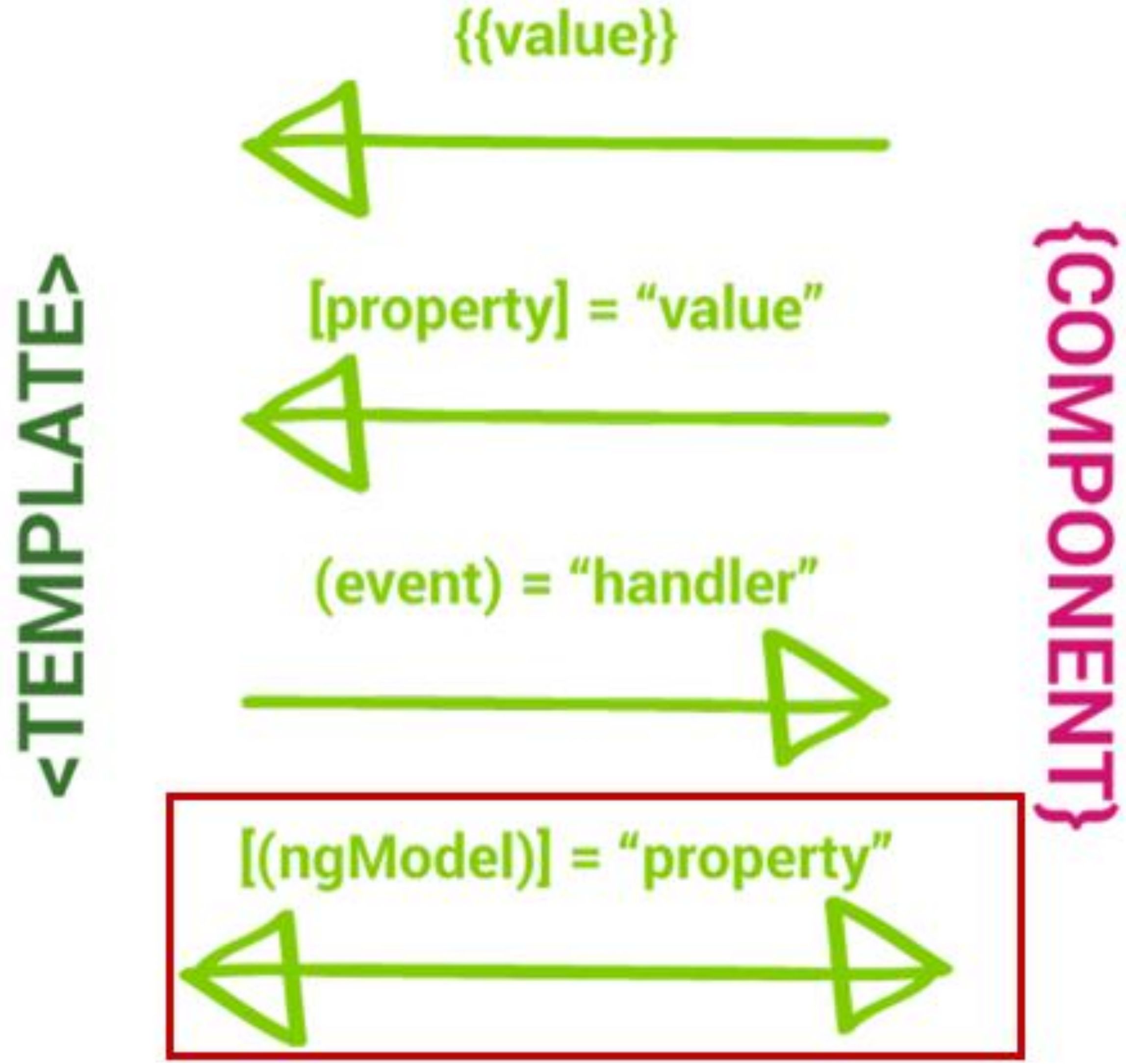
Data Binding



Data Binding

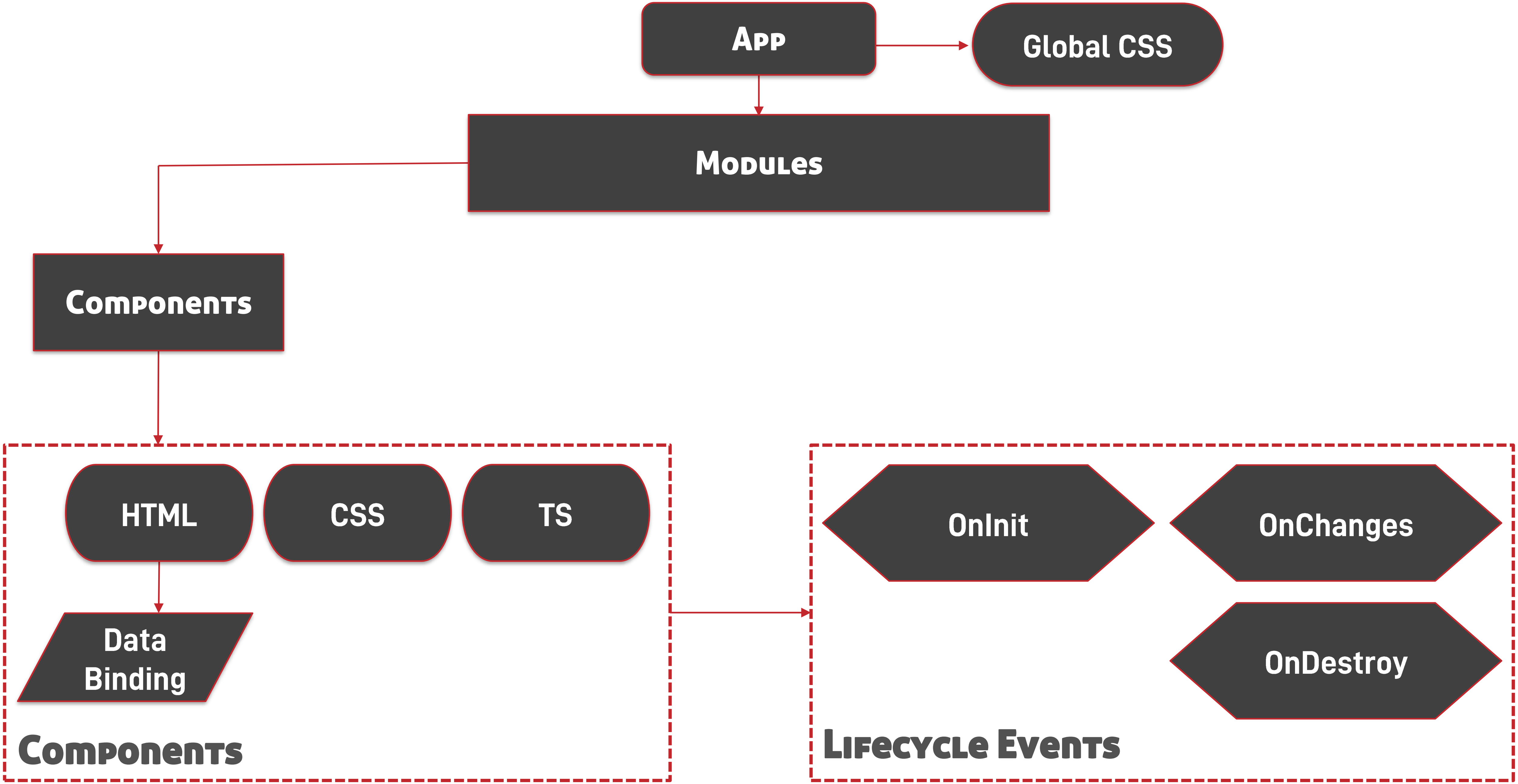
```
<div (click)="doSomething($event)"></div>
```

Data Binding



Data Binding

```
<input [(ngModel)]="title" />
```


App
Global CSS
Modules
Components
HTML
CSS
TS
Data Binding
Components
OnInit
OnChanges
OnDestroy
Lifecycle Events

HeaderComponent

todo.component.ts

```
@Component({
  selector: 'app-todo',
  templateUrl: './todo.component.html',
  styleUrls: ['./todo.component.scss'],
})
export class TodoComponent implements OnInit {
  ngOnInit() {
    this.getTodoListAll();
  }
}
```

HeaderComponent

todo.component.ts

```
@Component({
  selector: 'app-todo',
  templateUrl: './todo.component.html',
  styleUrls: ['./todo.component.scss'],
})
export class TodoComponent implements OnInit {
  ngOnInit() {
    this.getTodoListAll();
  }
}
```

HeaderComponent

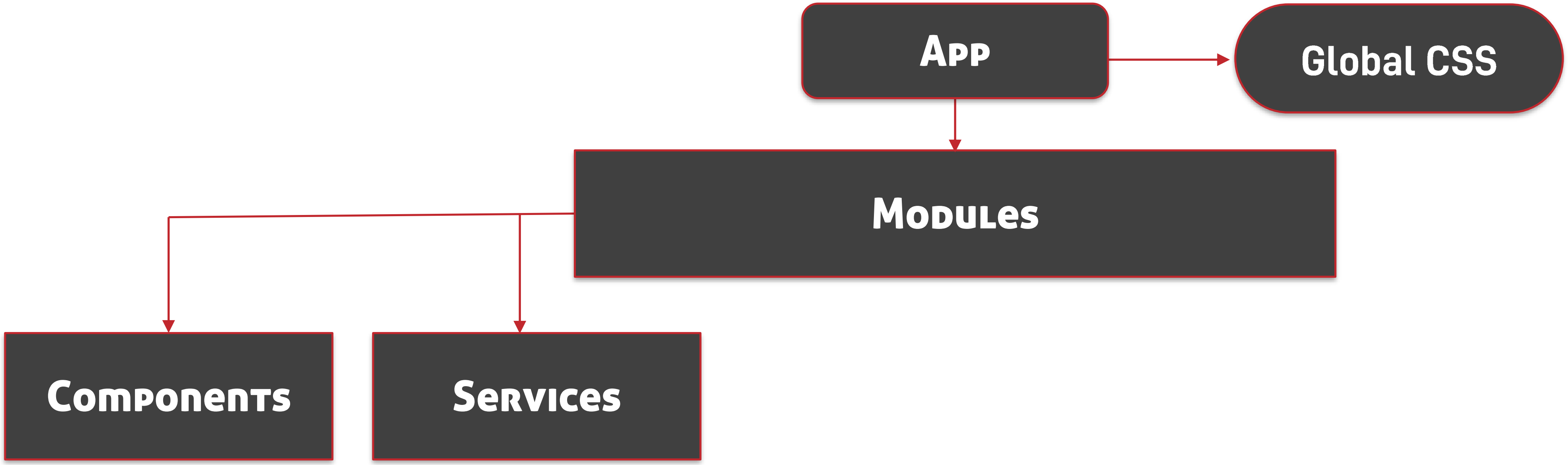
todo.component.ts

```
@Component({
  selector: 'app-todo',
  templateUrl: './todo.component.html',
  styleUrls: ['./todo.component.scss'],
})
export class TodoComponent implements OnInit {
  ngOnInit() {
    this.getTodoListAll();
  }
}
```

HeaderComponent

todo.component.ts

```
@Component({
  selector: 'app-todo',
  templateUrl: './todo.component.html',
  styleUrls: ['./todo.component.scss'],
})
export class TodoComponent implements OnInit {
  ngOnInit() {
    this.getTodoListAll();
  }
}
```


App
Global CSS
Modules
Components
Services

```
ng generate service <Name>
ng g s <Name>
```

Service Code – TodoService

shared\services\todo.service.ts

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

@Injectable({
  providedIn: 'root'
})
export class TodoService {
 constructor(private http: HttpClient) {}
}
```

Service Code – TodoService

shared\services\todo.service.ts

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

@Injectable({
  providedIn: 'root'
})
export class TodoService {
 constructor(private http: HttpClient) {}
}
```

Service Code – TodoService

shared\services\todo.service.ts

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

@Injectable({
  providedIn: 'root'
})
export class TodoService {
 constructor(private http: HttpClient) {}
}
```

Service Code – TodoService

shared\services\todo.service.ts

```typescript
export class TodoService {
  constructor(private http: HttpClient) {}

  getAll(): Observable<Todo[]> {
    return this.http.get<Todo[]>(this.url);
  }
}
```

Service Code – TodoService

shared\services\todo.service.ts

```
import { Observable } from 'rxjs';
import { Todo } from '../classes/todo';

export class TodoService {
   constructor(private http: HttpClient) {}

   getAll(): Observable<Todo[]> {
      return this.http.get<Todo[]>(this.url);
   }
}
```

Service Code – TodoService

shared\services\todo.service.ts

```
import { Observable } from 'rxjs';
import { Todo } from '../classes/todo';

export class TodoService {
   constructor(private http: HttpClient) {}

   getAll(): Observable<Todo[]> {
      return this.http.get<Todo[]>(this.url);
   }
}
```

Service Code – TodoService

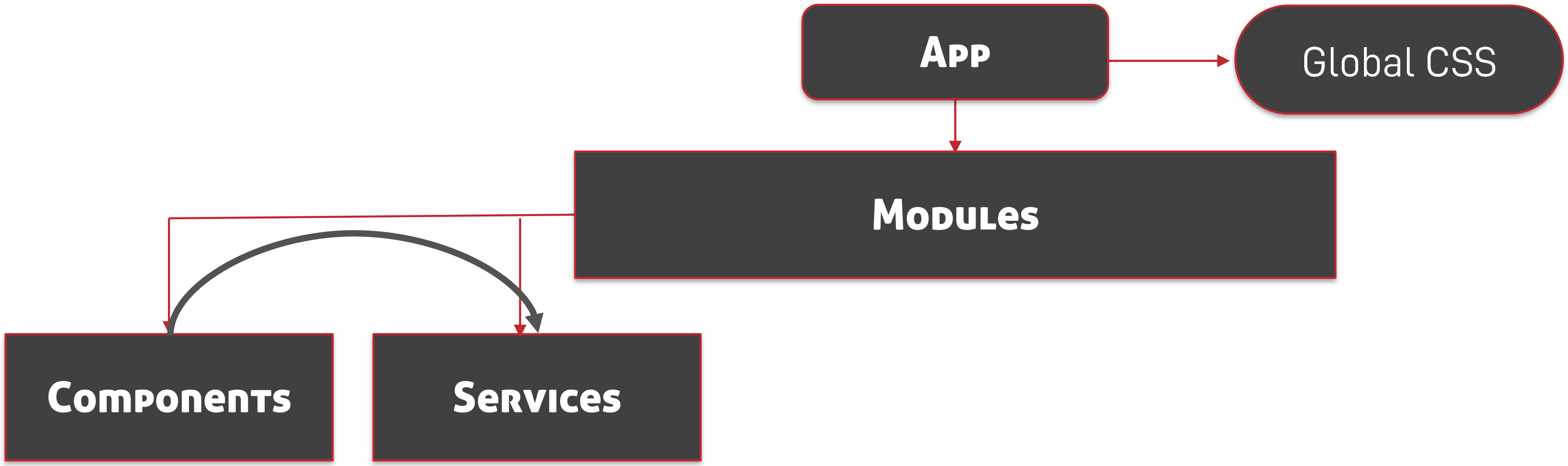
shared\services\todo.service.ts

```
import { Observable } from 'rxjs';
import { Todo } from '../classes/todo';

export class TodoService {
   constructor(private http: HttpClient) {}

   getAll(): Observable<Todo[]> {
      return this.http.get<Todo[]>(this.url);
   }
}
```

Service Code – TodoService

shared\services\todo.service.ts

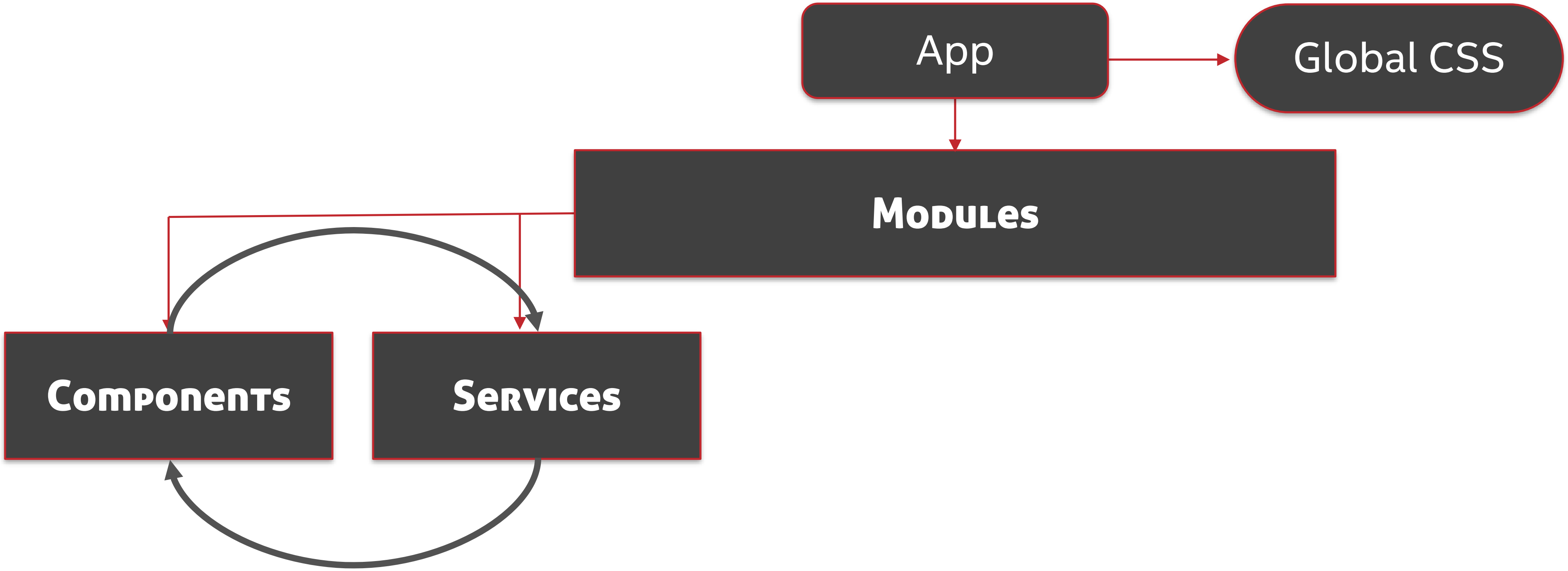
```
import { Observable } from 'rxjs';
import { Todo } from '../classes/todo';

export class TodoService {
   constructor(private http: HttpClient) {}

   getAll(): Observable<Todo[]> {
      return this.http.get<Todo[]>(this.url);
   }
}
```


App
Global CSS
Modules
Components
Services

Getting Data – TodoComponent

todo.component.ts

```
todoList: Todo[] = [];

this.todoService.getAll().subscribe(
   (data: Todo[]) => {
      this.todoList = data;
   },
   (error: HttpErrorResponse) => {
      this.errorMessage = `${error.message}`;
   }
);
```



App

Global CSS

Modules

Components

Services

Getting Data – TodoComponent

todo.component.ts

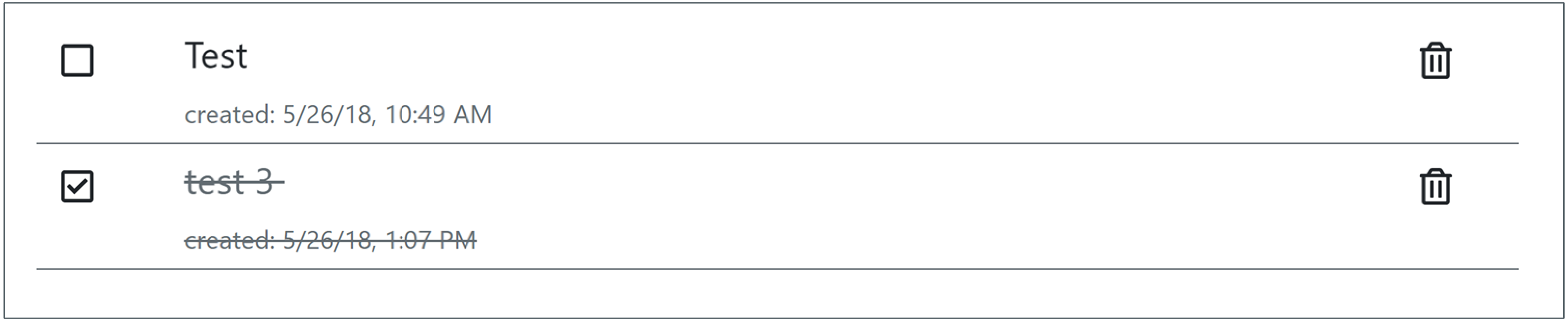
```
todoList: Todo[] = [];

this.todoService.getAll().subscribe(
   (data: Todo[]) => {
      this.todoList = data;
   },
   (error: HttpErrorResponse) => {
      this.errorMessage = `${error.message}`;
   }
);
```

Getting Data – TodoComponent

todo.component.ts

```
todoList: Todo[] = [];

this.todoService.getAll().subscribe(
   (data: Todo[]) => {
      this.todoList = data;
   },
   (error: HttpErrorResponse) => {
      this.errorMessage = `${error.message}`;
   }
);
```

Display – TodoComponent



□ Test

created: 5/26/18, 10:49 AM

☑ ~~test 3~~

~~created: 5/26/18, 1:07 PM~~

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```html
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

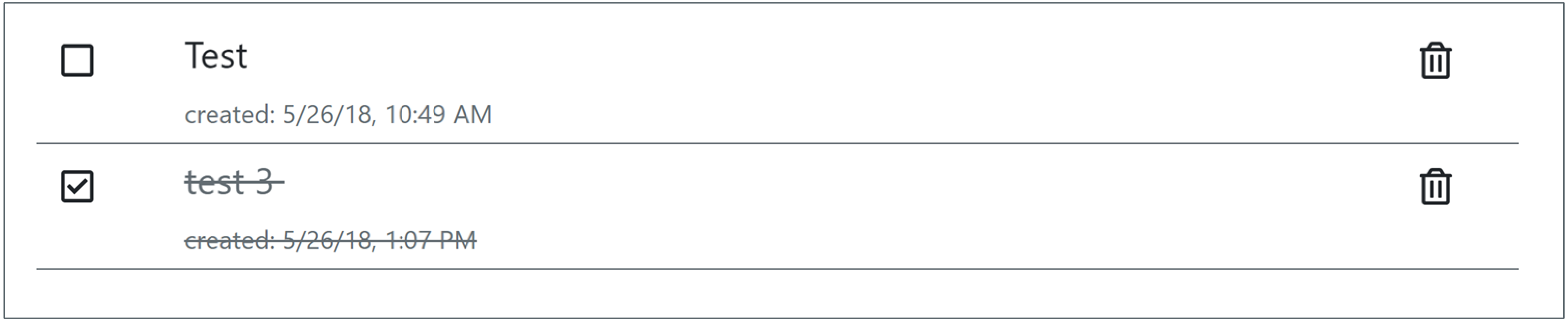
```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display Data – TodoComponent

todo.component.html

```
<div class="row todo" *ngFor="let todoItem of todoList">
  <div class="col-1" (click)="completeTodo(todoItem)">
    <fa-icon
        [icon]="['far', todoItem.completed ? 'check-square' : 'square']">
    </fa-icon>
  </div>
  <div class="col-10 done-{{todoItem.completed}}">
    {{todoItem.item}}
    <br />
    <small>created: {{ todoItem.createdAt | date:'short'}}</small>
  </div>
  <div class="col-1" (click)="deleteTodo(todoItem)">
    <fa-icon [icon]="['far', 'trash-alt']"></fa-icon>
  </div>
</div>
```

Display – TodoComponent

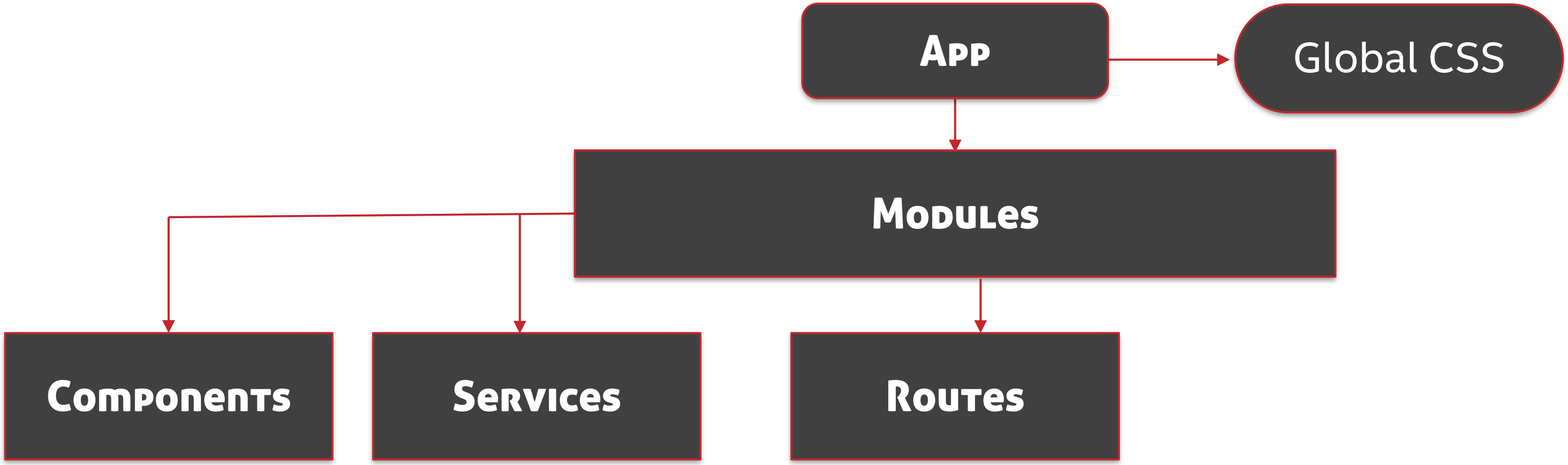


☐ Test 🗑

created: 5/26/18, 10:49 AM

☑ ~~test 3~~ 🗑

~~created: 5/26/18, 1:07 PM~~


App
Global CSS
Modules
Components
Services
Routes

AppRoutingModule

app-routing.module.ts

```
const routes: Routes = [
    {
        path: '',
        component: TodoComponent,
    },
    {
        path: 'login',
        children: [],
        component: LoginComponent
    },
];
```

AppRoutingModule

app-routing.module.ts

```
const routes: Routes = [
   {
      path: '',
      component: TodoComponent,
   },
   {
      path: 'login',
      children: [],
      component: LoginComponent
   },
];
```

AppRoutingModule

app-routing.module.ts

```
const routes: Routes = [
    {
        path: '',
        component: TodoComponent,
    },
    {
        path: 'login',
        children: [],
        component: LoginComponent
    },
];
```

AppRoutingModule

app-routing.module.ts

```
const routes: Routes = [
    {
        path: '',
        component: TodoComponent,
    },
    {
        path: 'login',
        children: [],
        component: LoginComponent
    },
];
```

AppRoutingModule

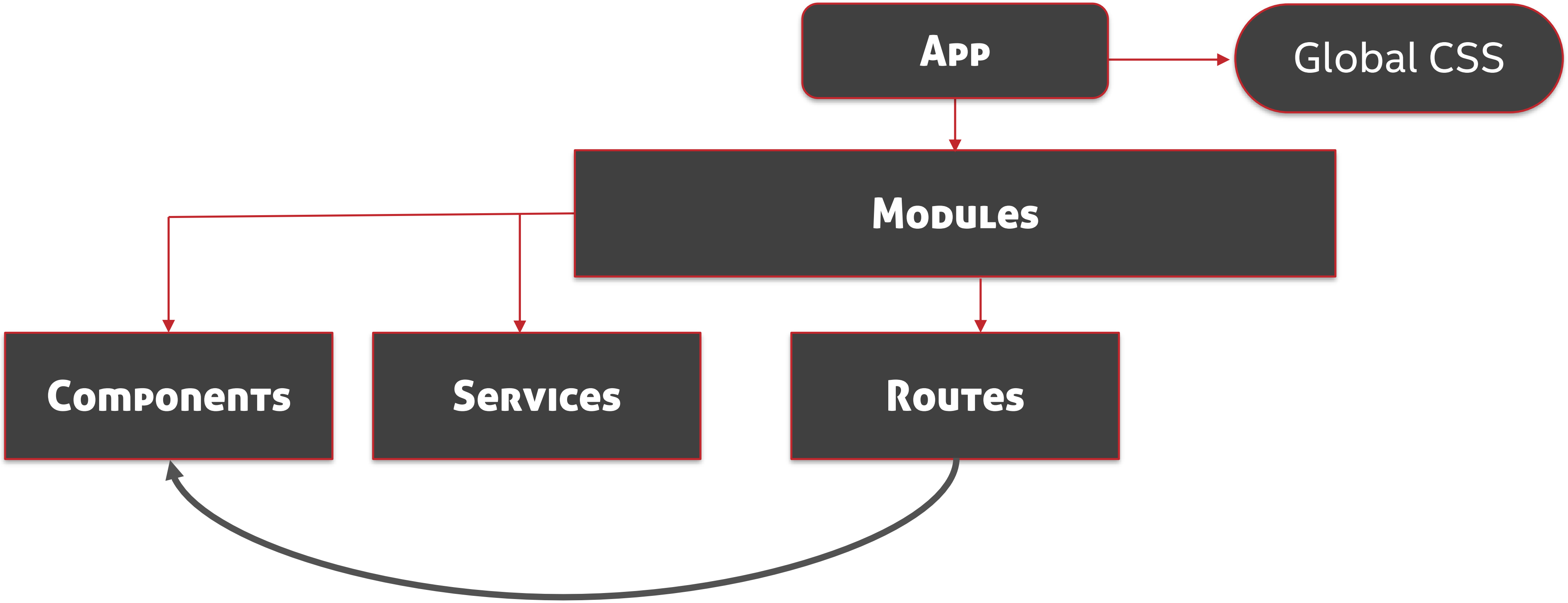
app-routing.module.ts

```
const routes: Routes = [
    {
        path: '',
        component: TodoComponent,
    },
    {
        path: 'login',
        children: [],
        component: LoginComponent
    },
];
```

AppRoutingModule

app-routing.module.ts

```
const routes: Routes = [
   {
      path: '',
      component: TodoComponent,
   },
   {
      path: 'login',
      children: [],
      component: LoginComponent
   },
];
```



App

Global CSS

Modules

Components

Services

Routes

App Component Html

app.component.html

```
<div class="jumbotron">
  <div class="container">
    <h1>{{title}}</h1>
  </div>
</div>
<div class="container">
  <router-outlet></router-outlet>
</div>
```

App Component Html

app.component.html

```html
<div class="jumbotron">
  <div class="container">
    <h1>{{title}}</h1>
  </div>
</div>
<div class="container">
  <router-outlet></router-outlet>
</div>
```

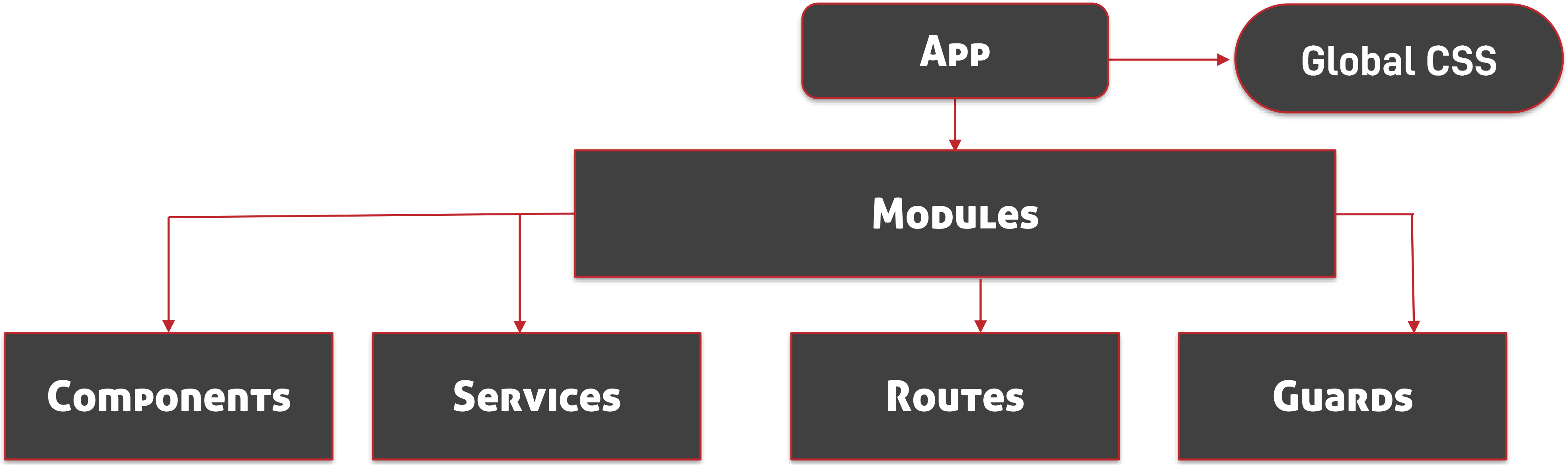
Default Routes

What to do when Angular doesn't where to go to....

AppRoutingModule

app-routing.module.ts

```
const routes: Routes = [
   ....,
   {
      path: '**',
      component: NotFoundComponent
   }
];
```


App
Global CSS
Modules
Components
Services
Routes
Guards

```
ng generate guard <Name>
ng g g <Name>
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
    CanActivate, ActivatedRouteSnapshot,
    RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
    canActivate(
        next: ActivatedRouteSnapshot,
        state: RouterStateSnapshot
    ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
    CanActivate, ActivatedRouteSnapshot,
    RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
    canActivate(
        next: ActivatedRouteSnapshot,
        state: RouterStateSnapshot
    ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
   CanActivate, ActivatedRouteSnapshot,
   RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
   canActivate(
      next: ActivatedRouteSnapshot,
      state: RouterStateSnapshot
   ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
    CanActivate, ActivatedRouteSnapshot,
    RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
    canActivate(
        next: ActivatedRouteSnapshot,
        state: RouterStateSnapshot
    ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
    CanActivate, ActivatedRouteSnapshot,
    RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
    canActivate(
        next: ActivatedRouteSnapshot,
        state: RouterStateSnapshot
    ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
   CanActivate, ActivatedRouteSnapshot,
   RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
   canActivate(
      next: ActivatedRouteSnapshot,
      state: RouterStateSnapshot
   ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Guard

is-logged-in.guard.ts

```
import { Injectable } from '@angular/core';
import {
   CanActivate, ActivatedRouteSnapshot,
   RouterStateSnapshot, Router
} from '@angular/router';

@Injectable()
export class IsLoggedInGuard implements CanActivate {
   canActivate(
      next: ActivatedRouteSnapshot,
      state: RouterStateSnapshot
   ): Observable<boolean> | Promise<boolean> | boolean {}
}
```

Restricting Route

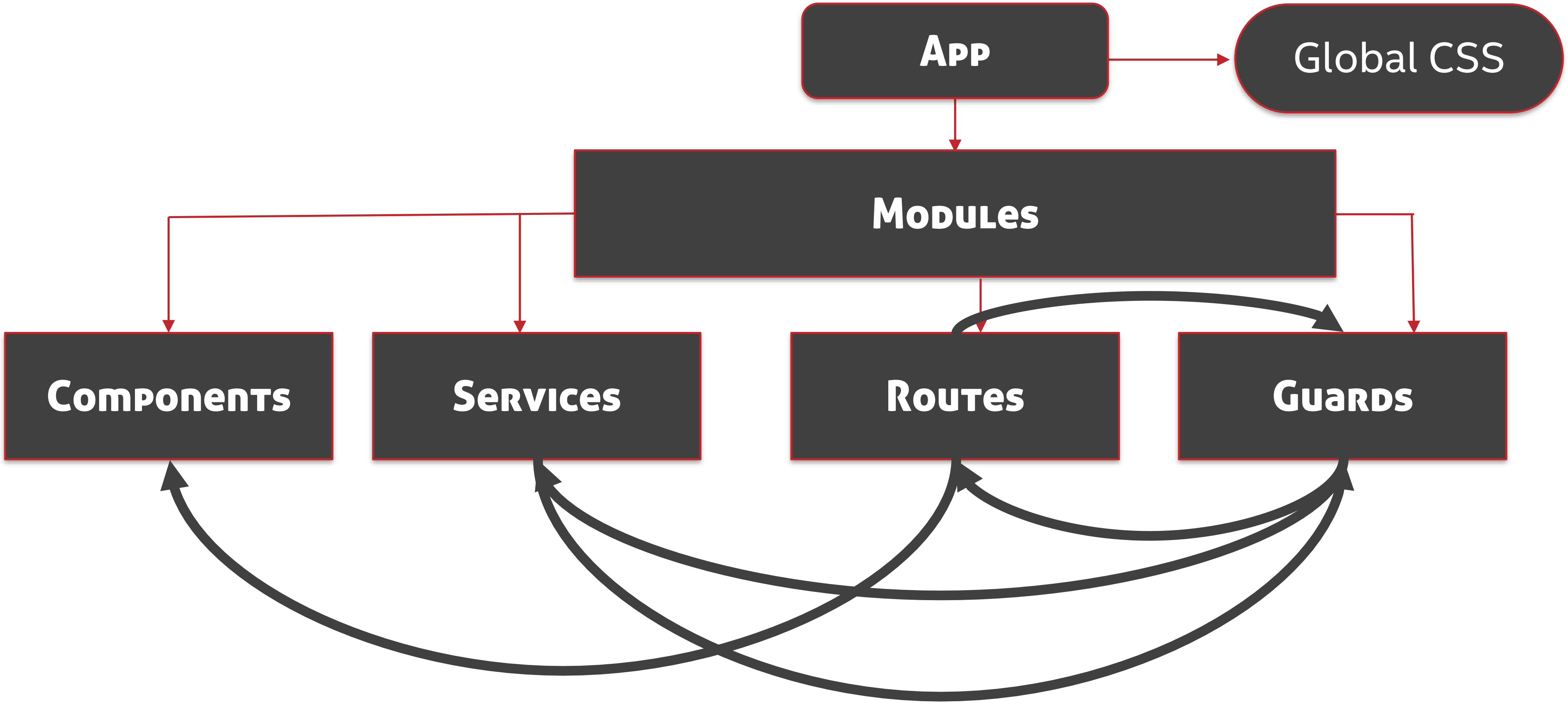
app-routing.module.ts

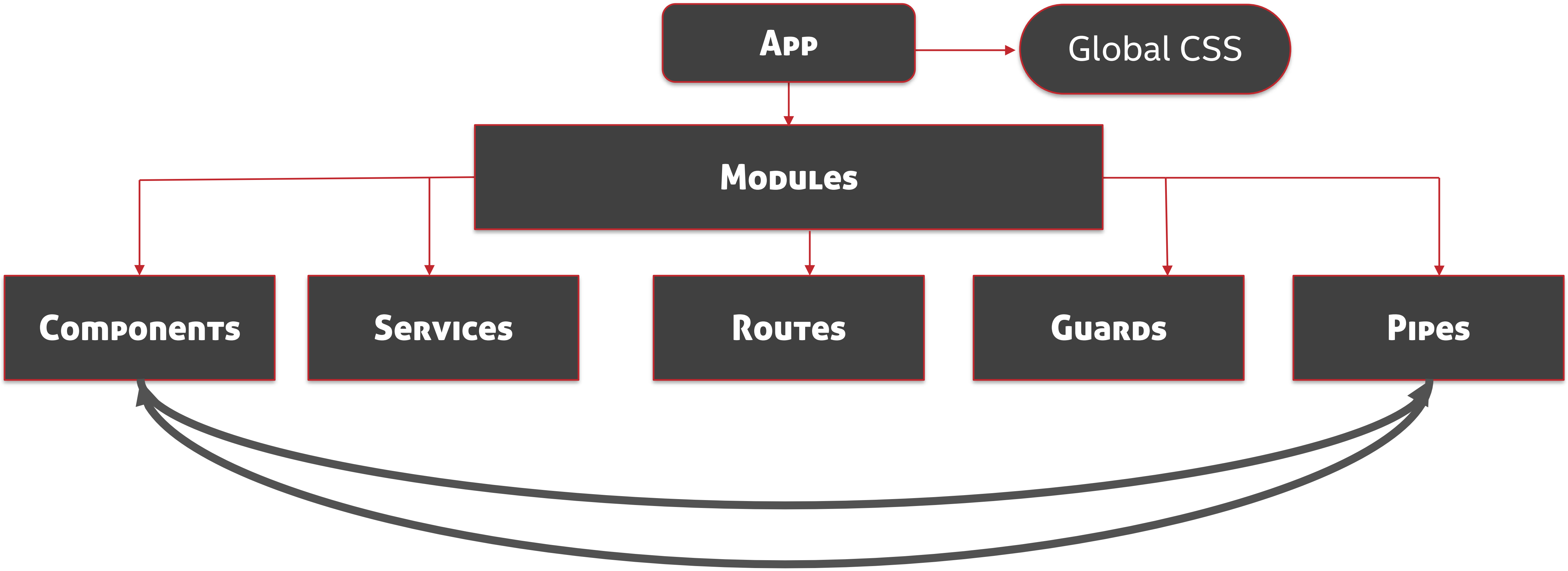
```
const routes: Routes = [
   {
      path: '',
      component: TodoComponent,
      canActivate: [IsLoggedInGuard],
   },
];
```

Restricting Route

app-routing.module.ts

```
const routes: Routes = [
  {
    path: '',
    component: TodoComponent,
    canActivate: [IsLoggedInGuard],
  },
];
```


App
Global CSS
Modules
Components
Services
Routes
Guards


App
Global CSS
Modules
Components
Services
Routes
Guards
Pipes

```
ng generate pipe <Name>
ng g p <Name>
```

Item Text Pipe

shared/pipe/item-text.pipe

```
@Pipe({
  name: 'itemText',
})
export class ItemTextPipe implements PipeTransform {
  constructor(private datePipe: DatePipe,
       private sanitizer: DomSanitizer) {}

  transform(todo: Todo, args?: any): SafeHtml {
    return this.sanitizer.sanitize(
          SecurityContext.HTML,
          `formatted item here`);
  }
}
```

Item Text Pipe

shared/pipe/item-text.pipe

```
@Pipe({
  name: 'itemText',
})
export class ItemTextPipe implements PipeTransform {
  constructor(private datePipe: DatePipe,
       private sanitizer: DomSanitizer) {}

  transform(todo: Todo, args?: any): SafeHtml {
   return this.sanitizer.sanitize(
          SecurityContext.HTML,
          `formatted item here`);
  }
}
```

Item Text Pipe

shared/pipe/item-text.pipe

```
@Pipe({
  name: 'itemText',
})
export class ItemTextPipe implements PipeTransform {
  constructor(private datePipe: DatePipe,
       private sanitizer: DomSanitizer) {}

  transform(todo: Todo, args?: any): SafeHtml {
    return this.sanitizer.sanitize(
          SecurityContext.HTML,
          `formatted item here`);
  }
}
```

Item Text Pipe

shared/pipe/item-text.pipe

```
@Pipe({
  name: 'itemText',
})
export class ItemTextPipe implements PipeTransform {
  constructor(private datePipe: DatePipe,
       private sanitizer: DomSanitizer) {}

  transform(todo: Todo, args?: any): SafeHtml {
    return this.sanitizer.sanitize(
          SecurityContext.HTML,
          `formatted item here`);
  }
}
```

Using Pipe

todo.component.html

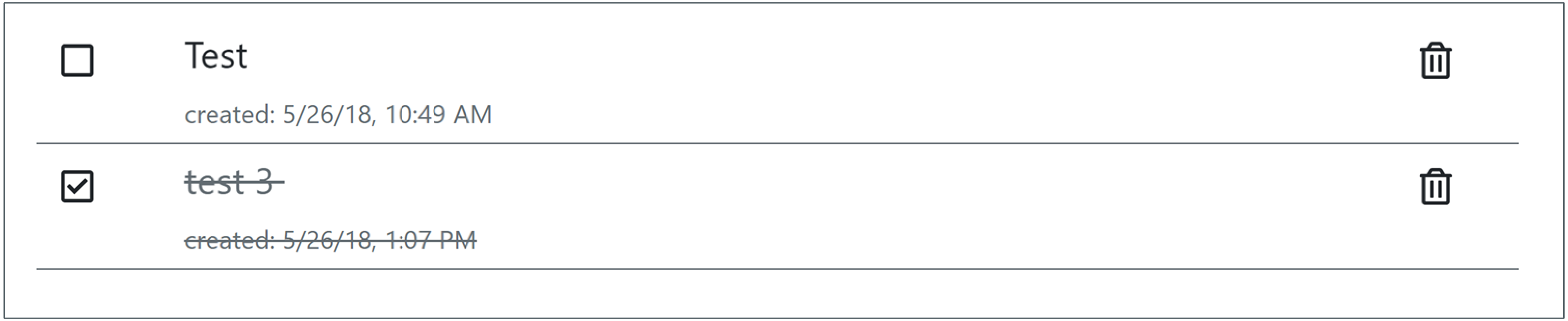
```
<div
      class="col-10 done-{{ todoItem.completed }}"
      [innerHtml]="todoItem | itemText">
</div>
```

Using Pipe

todo.component.html

```
<div
      class="col-10 done-{{ todoItem.completed }}"
      [innerHtml]="todoItem | itemText">
</div>
```

Display – TodoComponent

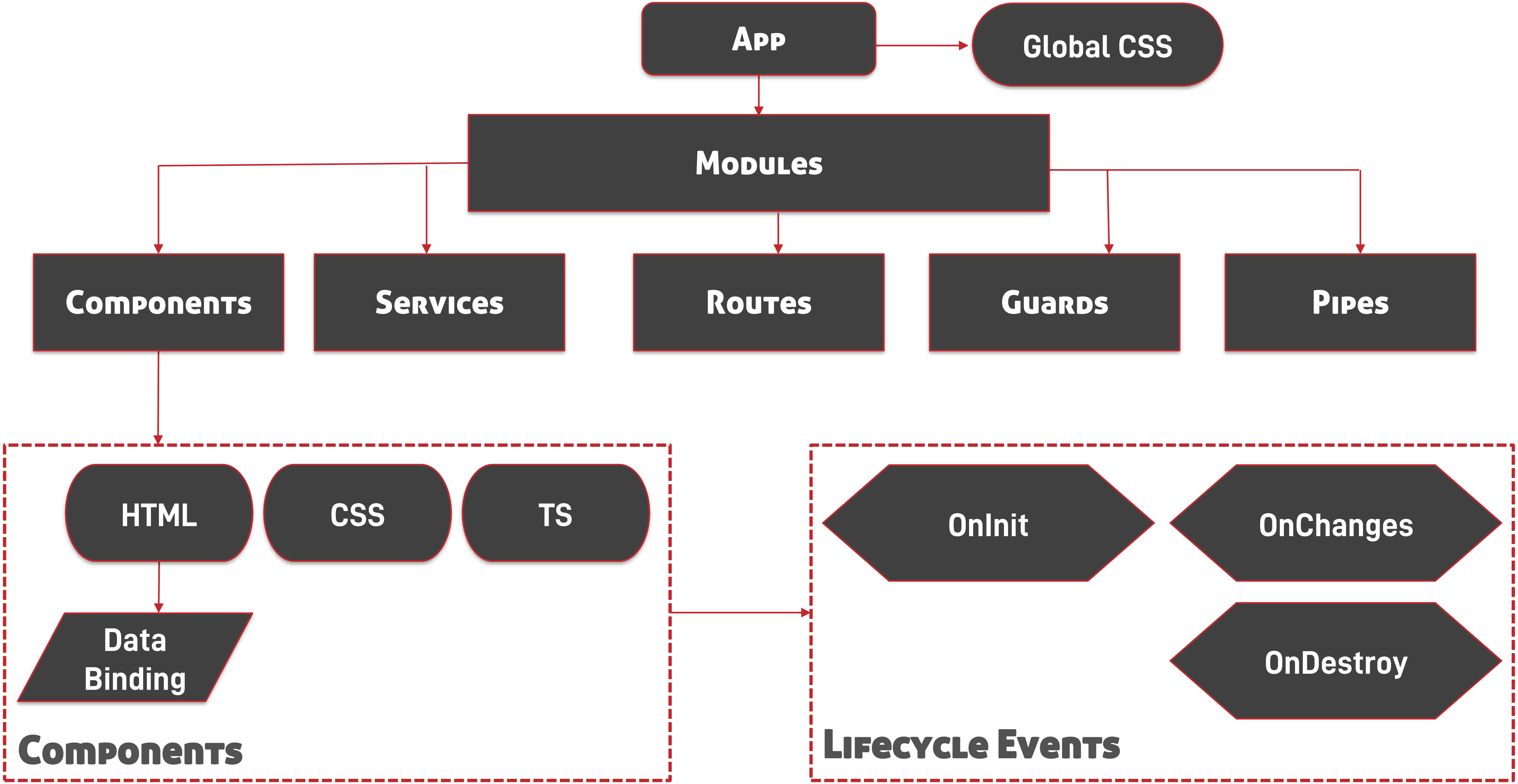


☐ Test

created: 5/26/18, 10:49 AM

☑ ~~test 3~~

~~created: 5/26/18, 1:07 PM~~


App
Global CSS
Modules
Components
Services
Routes
Guards
Pipes
HTML
CSS
TS
Data Binding
Components
OnInit
OnChanges
OnDestroy
Lifecycle Events

Connecting After Presentation

letyournerdbeheard.com/youtube

nerdbeheard

letyournerdbeheard.com/apply