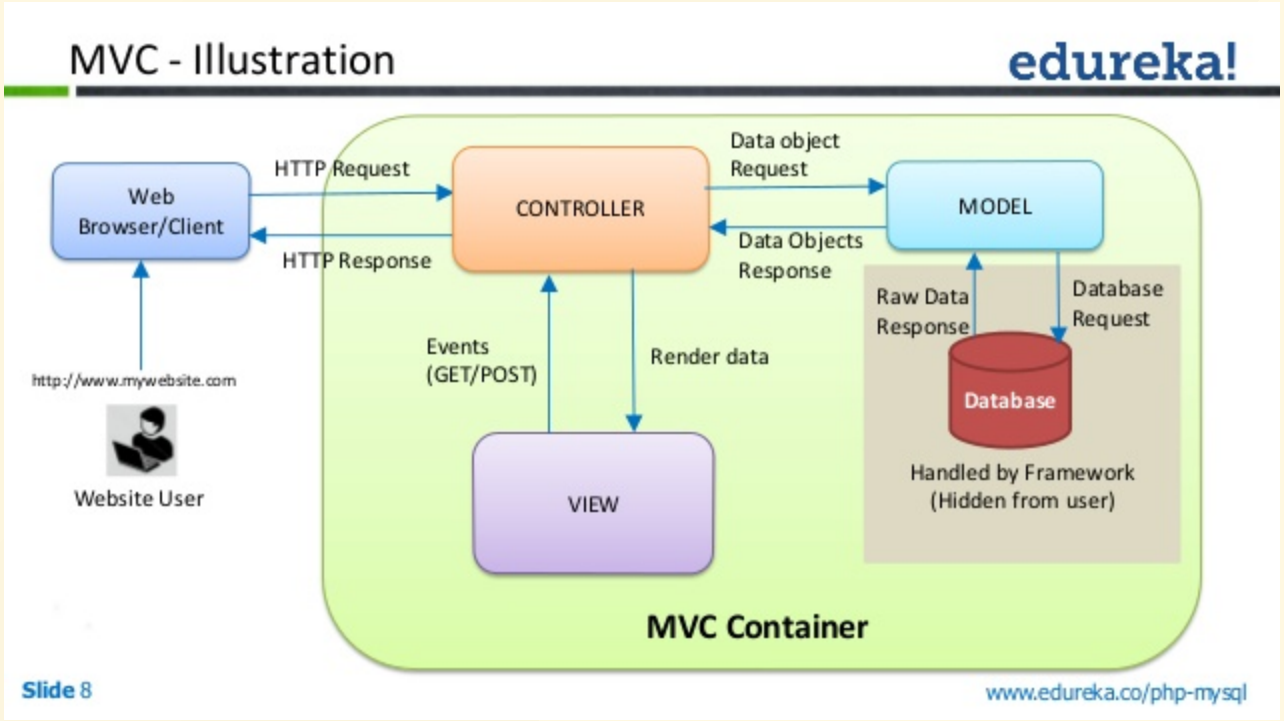
Angular

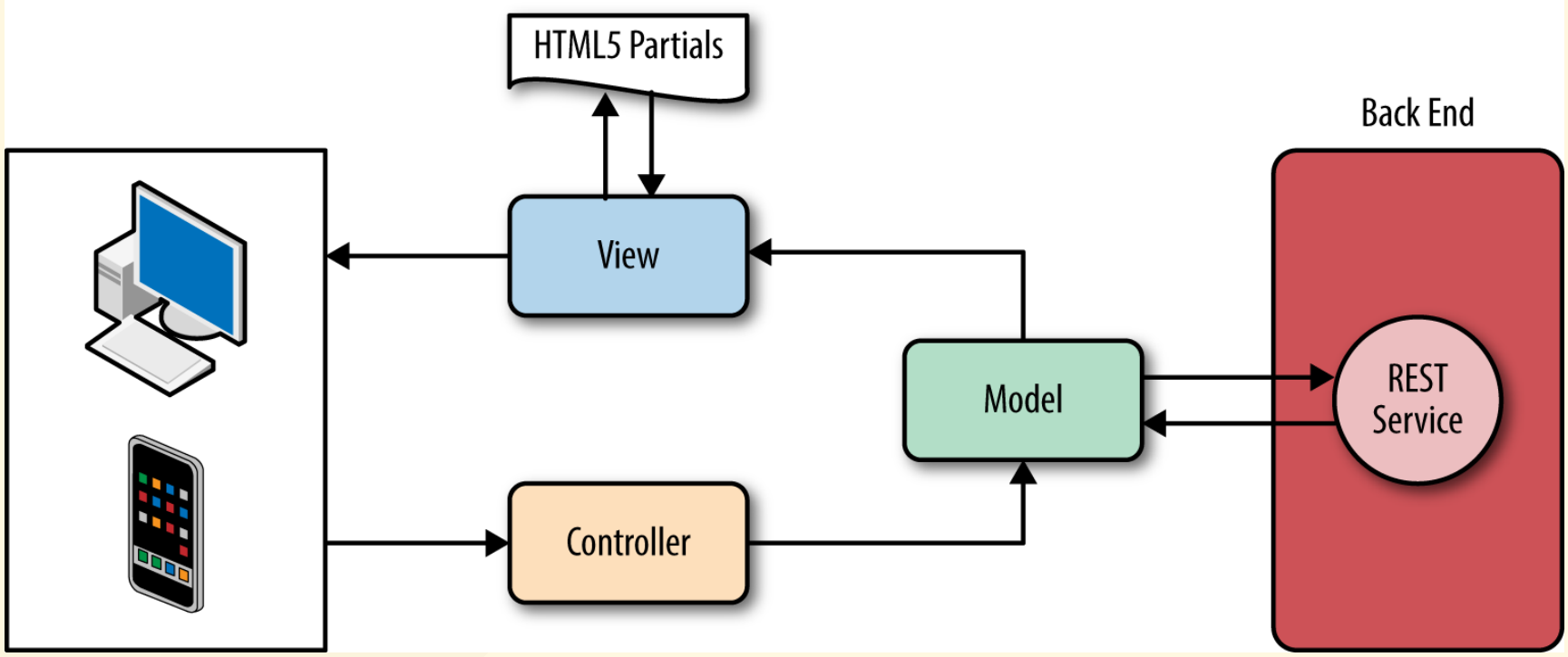
winter 2017 - Lunch & Learn

Stack - Before



- Model MVC backend

Stack - After



- Model MVC front-end

Trends

Requirements

- node 6.9.x +
- npm 3.x.x +

Angular - Getting start

1. Install the cli

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

2. Create new project

```
ng new demo-app --style=sass
```

3. Serve

```
cd demo-app
ng serve --open
```

Linter

- lint you app code using tslint.

```
ng lint
```

```
664240@FR-2FXVP12 C:\Users\A664240\Documents\work\worspace\angular-demo
 ng lint
arning: The 'no-use-before-declare' rule requires type checking

RROR: C:/Users/A664240/Documents/work/worspace/angular-demo/src/app/app.module.ts[19, 23]: missing whitespace
RROR: C:/Users/A664240/Documents/work/worspace/angular-demo/src/app/app.module.ts[20, 16]: missing whitespace

int errors found in the listed files.

664240@FR-2FXVP12 C:\Users\A664240\Documents\work\worspace\angular-demo
 ng lint
arning: The 'no-use-before-declare' rule requires type checking


ll files pass linting.

664240@FR-2FXVP12 C:\Users\A664240\Documents\work\worspace\angular-demo
```

Unit test

- run the unit tests :

```
ng test
```

End to end test

- run the test :

```
ng e2e
```

Bootstrap 4 - (1)

Bootstrap 4.0.0.beta : http://getbootstrap.com/

install via npm :

```
npm install --save bootstrap@4.0.0-beta font-awesome
```

Import css in `src/styles.css`

```
@import "~bootstrap/dist/css/bootstrap.min.css";
@import "~font-awesome/css/font-awesome.css";
```

Bootstrap 4 - (2)

Edit the `src/app/app.component.html` and `src/styles.css`

```
<div class="container">
  <div>
    <h1>Angular CLI + Bootstrap</h1>
    <p class="lead">This tutorial shows how to use Bootstr
  </div>
</div>
```

```
body {
  padding-top: 5rem;
}
```

Bootstrap 4 - (3)

Create nav component

```
ng generate component nav
```

will create a folder `src/app/nav/`

```
nav.component.ts       - The component class + definition
nav.component.css      - The component stylesheet
nav.component.html     - The component template
nav.component.spec.ts - The component test spec class
```

Refer `<app-nav></app-nav>` in the `app.component.html` file

Bootstrap 4 - (4)

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark fi
    <a class="navbar-brand" href="#">Angular CLI + Bootstr
    <button class="navbar-toggler" type="button" data-togg
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarsExamp
      <<ul class="nav navbar-nav mr-auto">
          <li class="nav-item">
            <a class="nav-link"  routerLinkActive="active"
          </li>
          <li class="nav-item">
            <a class="nav-link"  routerLinkActive="active"
          </li>
          <li class="nav-item">
            <a class="nav-link"  routerLinkActive="active"
          </li>
        </ul>
```

Pages

- Add two componoent pages :

```
ng generate component ticket/ticket-manager
ng generate component ticket/ticket-calculator
ng generate component home/home
```

```
A664240@FR-2FXVP12 C:\Users\A664240\Documents\work\worspace\formation\angular\lunch-learn
$ ng generate component ticket/ticket-manager
installing component
  create src\app\ticket\ticket-manager\ticket-manager.component.css
  create src\app\ticket\ticket-manager\ticket-manager.component.html
  create src\app\ticket\ticket-manager\ticket-manager.component.spec.ts
  create src\app\ticket\ticket-manager\ticket-manager.component.ts
  update src\app\app.module.ts

A664240@FR-2FXVP12 C:\Users\A664240\Documents\work\worspace\formation\angular\lunch-learn
$ ng generate component ticket/ticket-calculator
installing component
  create src\app\ticket\ticket-calculator\ticket-calculator.component.css
  create src\app\ticket\ticket-calculator\ticket-calculator.component.html
  create src\app\ticket\ticket-calculator\ticket-calculator.component.spec.ts
  create src\app\ticket\ticket-calculator\ticket-calculator.component.ts
  update src\app\app.module.ts
```

Routing (1)

in the `app.module.ts` add the router dependency

```
import { RouterModule }   from '@angular/router';
```

Configure the router

```
RouterModule.forRoot([{
    path: 'ticket-manager',
    component: TicketManagerComponent
  },{
    path: 'ticket-calculator',
    component: TicketCalculatorComponent
  }
])
```

Routing (2)

Default routing :

```
RouterModule.forRoot([{
        path: '',
        redirectTo : 'home',
        pathMatch: 'full'
      }...
])
```

Modify the main template `app.component.html`

```
<app-nav></app-nav>
<div class="container">
  <div>
    <router-outlet></router-outlet>
```

Routing (3)

Set links in `nav.component.html` :

```
<li class="nav-item">
<a class="nav-link" routerLink="/home">Home </a>
</li>
...
```

Active link

```
<li class="nav-item">
<a class="nav-link" routerLinkActive="active" routerLink="
</li>
...
```

Ticket manager

- Edit the `ticket-manage.component.ts` file

```
<div class="container">
    <h1>Ticket manager</h1>
    <div class="container">
      <div class="row alert alert-secondary " role="alert"
          <div class="col-md-7">
            test
          </div>
          <div class="col-md-3">
            test2
          </div>
          <div class="col-md-2">
            <button class="btn btn-sm btn-success" type="b
            <button class="btn btn-sm btn-primary" type="b
            <button class="btn btn-sm btn-danger" type="bu
```

Model Ticket

Add new model ticket `model/ticket.model.ts`

```
export class TicketModel  {
  owner:string; value:number;
}
```

Add list of tickets in a storage service
`ng generate service service/storage` add it in `provided section`

```
ticket :  Array<TicketModel> = new Array<TicketModel>();

  constructor() {
    let ticket = new TicketModel ();
    ticket.owner = "Jean";
```

Injection Depencies IOC

Use the contstructor to inject the Storage Service

```
tickets : Array<TicketModel>;
  constructor(private storageService: StorageService) {
    this.tickets = storageService.tickets;
   }
```

Directives - *ngFor

- Use the tickets variables

```
<div
        *ngFor="let ticket of tickets"
    class="row alert alert-secondary "
    role="alert">
    <div class="col-md-7">
    {{ticket.owner}}
    </div>
    <div class="col-md-3">
    {{ticket.value}}
```

Component event

- in the `ticket-manager.component.html` file :

```
<button class="btn btn-primary" (click)="addTicket()">
        + Add new ticket
</button>
```

- And in the `ticket-manager.components.ts` file :

```
  addTicket(){
    const ticket = new TicketModel();
    ticket.owner='';
    this.tickets.push(ticket);
  }
```

Directives - *ngIf

```
 <button type="button" class="btn btn-sm btn-primary" (cli
```

```
<span *ngIf="!ticket.editable">
        {{ticket.owner}}
</span>
<span *ngIf="ticket.editable">
        <input >
</span>
```

Two way binding

Import the form :

```
@NgModule({
  imports: [
    [...]
    FormsModule
  ],
  [...]
})
```

In the `ticket-calculator` template

```
  <input [(ngModel)] = "ticket.owner" name="owner" >
```

Service ticket calcul

Add new service ticket calcul with this method:

```
calcul(total:number,tickets : Array<TicketModel>)
        : Array<TicketResultModel>{
    return null;
  }
```

TDD approach

Create test in the `ticket-calcul.service.spec.ts`

```
it('should calc 25', inject([TicketCalculService], (serv
  expect(service).toBeTruthy();
  let tickets : Array<TicketModel> = new Array<TicketMod
  {
    let t:TicketModel = new TicketModel();
    t.value = 5;
    tickets.push(t);
  }
  expect(service.calcul(25,tickets)[0].number).toBe(5);
}));
```

Run the test `ng test`

Event Binding

Add click event :

```
<button type="button" class="btn btn-primary" (click)="cal
```

And the method :

```
  calc() {
     this.ticketsResult = this.ticketCalculService.calcul(t
     this.totalResult = this.ticketCalculService.sumResult(
   }
```

Resources

https://angular-2-training-book.rangle.io/handout/advanced-components/elementref.html