Angular Lab 4

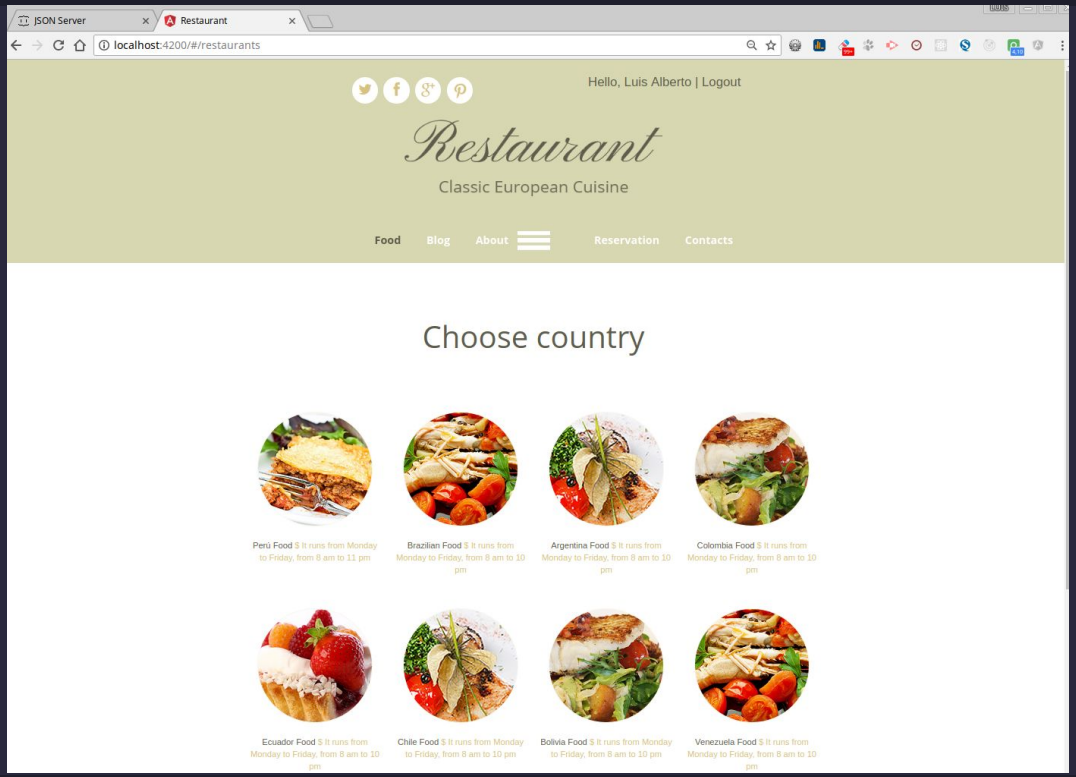
cst Academy

Componente

- Un set de elemente vizuale
- Definesc view-urile
- Folosite pentru reutilizabilitate & modularitate
- Au decoratorul @Component

```
@Component({
  selector:    'app-hero-list',
  templateUrl: './hero-list.component.html',
  providers:  [ HeroService ]
})
export class HeroListComponent implements OnInit {
  /* . . . */
}
```

Exercitiu


JSON Server
Restaurant
localhost:4200/#/restaurants
Hello, Luis Alberto | Logout
Restaurant
Classic European Cuisine
Food
Blog
About
Reservation
Contacts
Choose country
Perú Food $ It runs from Monday to Friday, from 8 am to 11 pm
Brazilian Food $ It runs from Monday to Friday, from 8 am to 10 pm
Argentina Food $ It runs from Monday to Friday, from 8 am to 10 pm
Colombia Food $ It runs from Monday to Friday, from 8 am to 10 pm
Ecuador Food $ It runs from Monday to Friday, from 8 am to 10 pm
Chile Food $ It runs from Monday to Friday, from 8 am to 10 pm
Bolivia Food $ It runs from Monday to Friday, from 8 am to 10 pm
Venezuela Food $ It runs from Monday to Friday, from 8 am to 10 pm

Module

- Mecanism de grupare a componentelor pentru crearea unei functionalitati
- Componentele sunt grupate dupa criteriile pe care le au in comun pentru functionalitatea pe care o descriu
- O aplicatie Angular poate fi vazuta ca un puzzle in care fiecare modul reprezinta o piesa
- Au decoratorul @NgModule

Module

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Routing

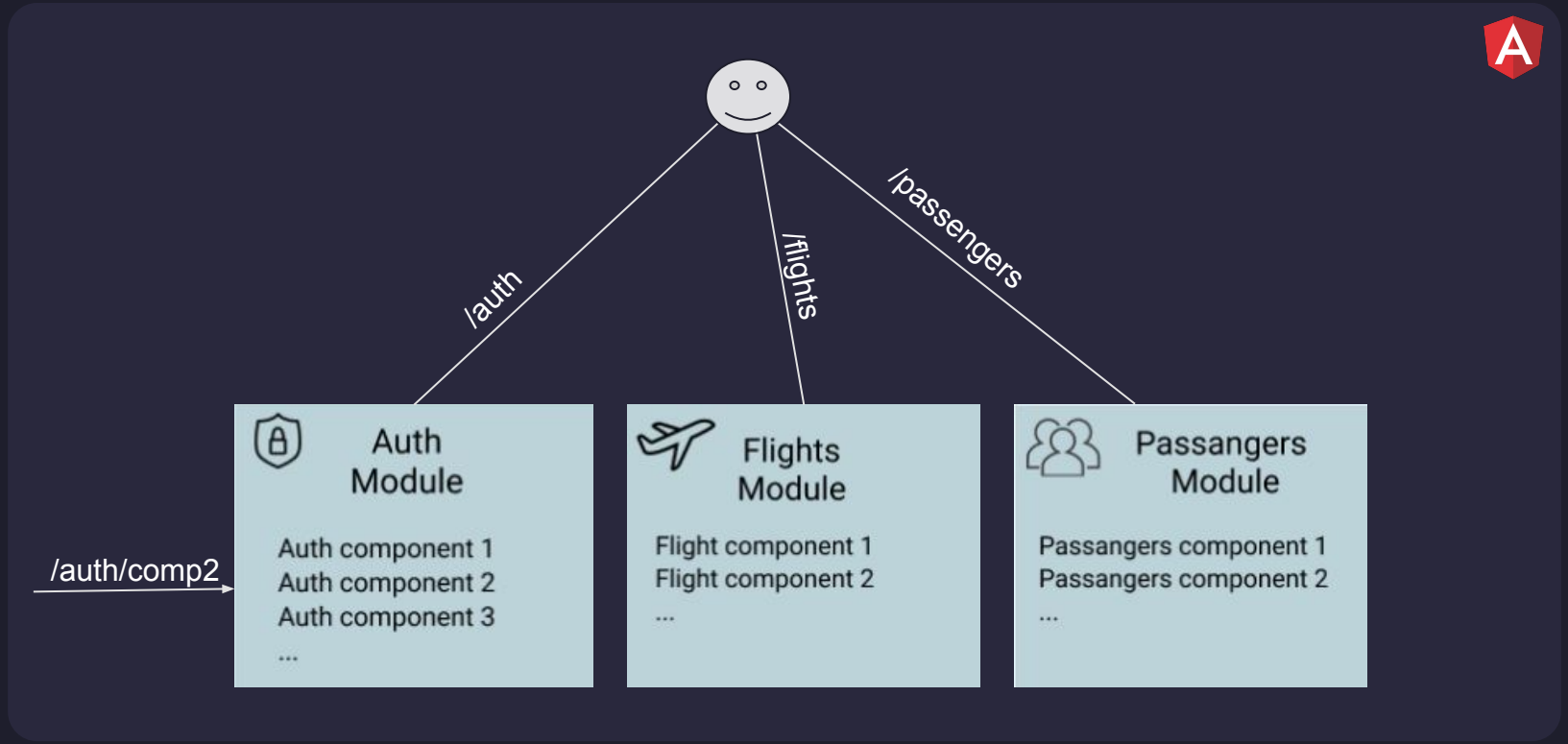
- Single Page Application - putem modifica ce vede userul in aplicatie ascunzand portiuni din aplicatie/componente, nu e nevoie sa cerem serverului o pagina noua
- Angular Router - vede url-ul browserului ca o instructiune de a schimba view-ul

Routing

```
const routes: Routes = [
  { path: 'first-component', component: FirstComponent },
  { path: 'second-component', component: SecondComponent },
];
```

```
// configures NgModule imports and exports
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Routing


/auth
/flights
/passengers
Auth Module
Auth component 1
Auth component 2
Auth component 3
...
Flights Module
Flight component 1
Flight component 2
...
Passangers Module
Passangers component 1
Passangers component 2
...
/auth/comp2

Lazy Loading

- Procedeu prin care incarcam modulele doar atunci cand e nevoie de ele

```
const routes: Routes = [
  {
    path: 'items',
    loadChildren: () => import('./items/items.module').then(m => m.ItemsModule)
  }
];
```

Data Binding

TYPE	SYNTAX	CATEGORY
Interpolation Property Attribute Class Style	`{{expression}}` `[target]="expression"`	One-way from data source to view target
Event	`(target)="statement"`	One-way from view target to data source
Two-way	`[(target)]="expression"`	Two-way

Data Binding

The binding punctuation of `[]`, `()`, `[()]`, and the prefix specify the direction of data flow.

- Use `[]` to bind from source to view
- Use `()` to bind from view to source
- Use `[()]` to bind in a two-way sequence of view to source to view

NgClass, NgStyle, NgModel

- Asculta si modifica comportamentul elementelor de HTML, atributelor, componentelor si proprietatilor

COMMON DIRECTIVES	DETAILS
`NgClass`	Adds and removes a set of CSS classes.
`NgStyle`	Adds and removes a set of HTML styles.
`NgModel`	Adds two-way data binding to an HTML form element.

ngIf & ngFor

```
<div *ngIf="hero" class="name">{{hero.name}}</div>
```

```
<div
  *ngFor="let hero of heroes; let i=index; let odd=odd; trackBy: trackById"
  [class.odd]="odd">
  ({{i}}) {{hero.name}}
</div>
```

cst© Academy