

Campus **L>SA** /2019/

OSP (Office for students with disabilities - Student Portal)

Technology stack

- Angular 7 front-end
- Dot Net Core 2.0 back-end

Points of Interest

- Communication between modules and components via services
- BehaviorSubjects for global variables and simple state management
- Lazy-Loaded modules
- Route Guards

Services

- Provide in the root for a single instance
- Good for sharing data across modules
- Good for communication with unrelated components/classes
- Good way to centralize http calls
- Helps separate business logic from presentation

Services

```
@Injectable({  
  providedIn: 'root'  
})  
export class StoredValuesService
```

```
constructor(private values: StoredValuesService,  
            private accommodationsService: AccommodationsService,  
            private errorHandler: ErrorHandlerService,  
            private modalService: BsModalService,  
            private termsService: AcademicTermsService,  
            private emailService: EmailService) { }
```

```
this.values.setHomeLink(['/home']);  
this.emptyShibHeaders = this.values.isShibEmpty();|
```

```
this.values.userName$.subscribe((name) => this.userName = name);
```

BehaviorSubjects

- Must have a default value
- Great for multi-casting
- Will always have access to the most recently emitted value

BehaviorSubjects

```
private homeUrlSource = new BehaviorSubject<string[]>(['']);
```

```
homeUrl$ = this.homeUrlSource.asObservable();
```

```
setHomeLink(home: string[]) { this.homeUrlSource.next(home); }  
getHomeLink(): string[] { return this.homeUrlSource.value; }
```

```
this.values.setHomeLink(['/home']);
```

```
constructor(private toggleService: StoredValuesService) { this.toggleService.homeUrl$.subscribe((link) => this.home = link); }
```

```
<a class="nav-link" [routerLink]="home">Home
```

Lazy Loading

- Speed up initial load time
- On demand loading
- Controlled by the angular router

Lazy Loading

```
export const appRoutes: Routes = [  
  { path: 'staff/home', loadChildren: './modules/Staff Modules/staff-home/staff-home.module#StaffHomeModule' },  
  { path: 'staff/:quarter', loadChildren: './modules/Staff Modules/staff-quarter/staff-quarter.module#StaffQuarterModule' },  
  { path: 'staff', redirectTo: '/staff/home', pathMatch: 'full' },  
  { path: 'home', loadChildren: './modules/Student Modules/home/home.module#HomeModule' },  
  { path: 'home/:quarter', loadChildren: './modules/Student Modules/quarter/quarter.module#QuarterModule' },  
  { path: '', redirectTo: '/home', pathMatch: 'full' },  
  { path: '**', redirectTo: '/home', pathMatch: 'full' }  
];
```

```
const routes: Routes = [{  
  path: '',  
  component: QuarterComponent,  
  children: [  
    {path: 'accommodations', component: AccommodationsPageComponent, canActivate: [AccommodationsGuardService]},  
    {path: 'consent', component: ConsentComponent, canActivate: [ConsentGuardService]}  
  ]  
}];
```

Route Guards

- Tell the router whether or not navigation is allowed on a specific route
- The available guards are:
 - CanActivate
 - CanActivateChild
 - CanDeactivate
 - CanLoad
 - Resolve

Route Guards

```
export class AccommodationsGuardService implements CanActivate {
  chosenQuarter: string;

  constructor(private router: Router, private route: ActivatedRoute, private accommodationsAccess: AccommodationsAccessService) {}

  canActivate(nextRoute: ActivatedRouteSnapshot, nextState: RouterStateSnapshot): boolean | Observable<boolean> | Promise<boolean> {
    this.chosenQuarter = nextRoute.parent.params.quarter;
    return this.accommodationsAccess.hasAccepted().pipe(map((result) => {
      if (result) { return true; } else {
        void this.router.navigate(['home', this.chosenQuarter, 'consent']);
        return false;
      }
    }));
  }
}
```

```
hasAccepted(): Observable<boolean> {
  const requestShape = this.values.getShape();
  return this.httpClient.get(this.agreementEndpoint + `/${requestShape.StudentId}/${requestShape.quarter}`)
    .pipe(map((result) => {
      console.log('Has student accepted consent: ', result);
      return true;
    })), catchError(this.handleError));
}
```

```
{path: 'accommodations', component: AccommodationsPageComponent, canActivate: [AccommodationsGuardService]},
```

Thank you!

<https://dev-academicaffairs.ucsd.edu/sso/osdsp/>