# Angular HttpClient Demo

## 1. Setup Project

### 1.1 Install JSON Server Dependencies

* Change directory to json-server:
* cd json-server
* Install dependencies by running the following command:
* npm install

### 1.2 Start JSON Server

* Start JSON server:
* npx json-server MOCK\_DATA.json -p 3001

### 1.3 Install Angular Dependencies

* Change directory to calab:
* cd calab
* Install dependencies by running the following command:
* npm install

### 1.4 Start The Application

* Start Angular Development Server if not yet started:
* npx -p @angular/cli ng serve --host 0.0.0.0
* Otherwise refresh the browser tab to see updated view.

## 2. Setup HttpClient

### 2.1 Update App Configuration

* Open src/app/app.config.ts file and do the following:
  + Provide provideHttpClient helper function:
  + export const appConfig: ApplicationConfig = {  
     providers: [ provideRouter(routes), provideHttpClient() ]  
    };

## 3. Create and configure a new Service

### 3.1 Create a new Service

* Create a new component using CLI and name it MovieService:
* npx -p @angular/cli ng generate service services/movie

### 3.2 Inject The HttpClient Service

* Open src/app/services/movie.service.ts file and do the following:
  + Inject HttpClient as a dependency into MovieService constructor.
  + export class MovieService {  
     constructor(private httpClient: HttpClient) { }  
    }

## 4. Create a Movie Model

### 4.1 Create a new class representing a movie model

* Create a new directory in src/app/ called models.
* Change directory to src/app/models
* Create new TypeScript file called movie.ts
* Add the following code inside movie.ts:
* export class Movie {  
   constructor(  
   public title: string,  
   public genre: string,  
   public release\_date: string,  
   public director: string,  
   public rating: number,  
   public duration\_minutes: number,  
   ) {}  
  }

## 5. Making Http Requests

### 5.1 Create a GET http request

* Open src/app/services/movie.service.ts file and do the following:
  + Declare a new function called getAllMovies that calls http get() method.
  + getAllMovies(){  
     this.httpClient.get<Movie>('http://localhost:3001/movies').subscribe(data => {  
     console.log(data);  
     });  
    }

### 5.2 Call method containing GET http request

* Open src/app/app.component.ts file and do the following:
  + Inject MovieService as dependency into AppComponent.
  + export class AppComponent {  
     constructor(private movieService: MovieService){}  
     ...  
    }
  + Inside constructor, make a call to getAllMovies() method.
  + export class AppComponent {  
     constructor(private movieService: MovieService){  
     movieService.getAllMovies();  
     }  
     ...  
    }

### 5.3 Create a POST http request

* Open src/app/services/movie.service.ts file and do the following:
  + Declare a new function called createMovie that calls http post() method.
  + this.httpClient.post<Movie>('http://localhost:3001/movies', movie).subscribe(res => {  
     console.log('Created movie:', res);  
    });

### 5.4 Call method containing POST http request

* Open src/app/app.component.ts file and do the following:
  + Inside constructor, create an instance of a movie.
  + const movie = new Movie(   
     "Forrest Gump",  
     "Drama",  
     "1994",  
     "Robert Zemeckis",  
     8.8,  
     142  
    );
  + Inside constructor, make a call to createMovie() method.
  + movieService.createMovie(movie);

### 5.5 Start The Application

* Start Angular Development Server if not yet started:
* npx -p @angular/cli ng serve --host 0.0.0.0
* Otherwise refresh the browser tab to see updated view.

## 6. Interceptors

### 6.1 Define an Interceptor

* Open src/app/app.config.ts file and do the following:
  + Define a loggingInterceptor helper function:
  + export function loggingInterceptor(req: HttpRequest<unknown>, next: HttpHandlerFn): Observable<HttpEvent<unknown>> {  
     console.log(`Request URL is: ${req.url}`);  
     return next(req);  
    }
  + Declare an interceptor inside provideHttpClient helper function:
  + export const appConfig: ApplicationConfig = {  
     providers: [provideRouter(routes), provideHttpClient(  
     withInterceptors([loggingInterceptor]),  
     ),]  
    };

### 6.2 Start The Application

* Start Angular Development Server if not yet started:
* npx -p @angular/cli ng serve --host 0.0.0.0
* Otherwise refresh the browser tab to see updated view.