



**One framework.
Mobile & desktop.**

10강. HTTP 모듈



DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

10 강.

HTTP 모듈

* HTTP 프로토콜

- 브라우저와 서버 통신을 위한 기본 프로토콜.
- XMLHttpRequest 객체를 사용하여 비동기 통신 가능. (Ajax)

* SOP (Same Origin Policy) 정책

- 현재 브라우저에서 보여지는 웹 페이지와 동일한 도메인에게만 Ajax 요청을 할 수 있다는 것.

* Cross Domain

- 다수의 도메인에 접근하는 것을 의미. (Open API 사용 및 분리된 서버 접근)
- 구현 방법 2가지
 - : CORS(Cross-Origin Resource Sharing)
 - : JSONP(JSON with Padding)



□ 2) Angular의 기본 HTTP 모듈

* Angular의 HTTP 처리

- Angular HTTP 라이브러리는 XHR과 JSONP를 사용하는 방법을 단순화 시킴.
- @angular/common/http 모듈을 사용한 서비스 형태로 지원.

```
import { HttpClientJsonpModule, HttpClientModule } from '@angular/common/http';
```

- Angular는 Promise 또는 Observable 객체를 사용하여 비동기 HTTP 통신을 처리함.
- Angular는 REST 서비스 지원함.

* RxJS 라이브러리

- 비동기 Observable 패턴을 구현하는 Angular가 보증하는 외부 라이브러리.
- Reactive eXtensions 로서 MS에서 만들고 2012년부터 OpenSource로 제공함.

```
import { Observable } from 'rxjs/Rx';  
import 'rxjs/add/operator/catch';  
import 'rxjs/add/operator/map';
```

□ 3) JSON 서버 구축

Node.js에서 제공하는 json 서버를 구축하여 실습에 활용한다. (크로스 도메인 가능)

1) VSC에서 프로젝트 생성

```
C:\angular>ng new json
```

```
C:\angular_chul>ng new json
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE json/angular.json (3550 bytes)
```

2) npm 이용하여 json 서버 설치(cd json)

```
C:\angular\json>npm install json-server --save
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fs
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsu
```

C:) > Angular_Study_chul_2

이름

- chul-app
- inky-app
- json

```
C:\angular_chul>cd json
```

```
C:\angular_chul\json>npm install json-server --save
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted
```

□ 3) JSON 서버 구축

3) Json파일 생성후 json 프로젝트에 저장
(db.json파일생성 후 작성)

```
{
  "todos": [
    {
      "id": 1,
      "title": "Read SitePoint article",
      "complete": false
    },
    {
      "id": 2,
      "title": "Clean inbox",
      "complete": false
    },
    {
      "id": 3,
      "title": "Make restaurant reservation",
      "complete": false
    }
  ]
}
```

```
C:\w\angular_chul\json>code .
```



```
▼ json
  > e2e
  > node_modules
  > src
  ⚙ .editorconfig
  🔍 .gitignore
  {} angular.json
  ≡ browserslist
  {} db.json
  ⬅ karma.conf.js
```

❑ 3) JSON 서버 구축

4) package.json 파일에 json 서버 실행 명령어 지정

```
"scripts": {  
  ...  
  "json-server": "json-server --watch db.json"  
}
```

package.json ×

on > {} package.json > {} dependencies > @angular/forms

```
1  {  
2    "name": "json",  
3    "version": "0.0.0",  
4    "scripts": {  
5      "ng": "ng",  
6      "start": "ng serve",  
7      "build": "ng build",  
8      "test": "ng test",  
9      "lint": "ng lint",  
10     "e2e": "ng e2e",  
11     "json-server": "json-server --watch db.json"  
12   },
```

□ 3) JSON 서버 구축

5) 서버 실행

```
npm run json-server
```

```
C:\#Angular_Study_chul_2\json>npm run json-server
> json@0.0.0 json-server C:\#Angular_Study_chul_2\json
> json-server --watch db.json
```

```
#{^_^}/ hi!
```

```
Loading db.json
Done
```

Resources

```
http://localhost:3000/todos
http://localhost:3000/books
```

Home



Congrats!

You're successfully running JSON Server
 ✧*.๑(๐`*)๑*.

Resources

[/todos](#) 3x

[/books](#) 6x

□ 4) JSON 서버 활용한 HTTP 모듈 사용 실습



1) Json 서버에 요청하여 응답 데이터 확인(“books추가”)

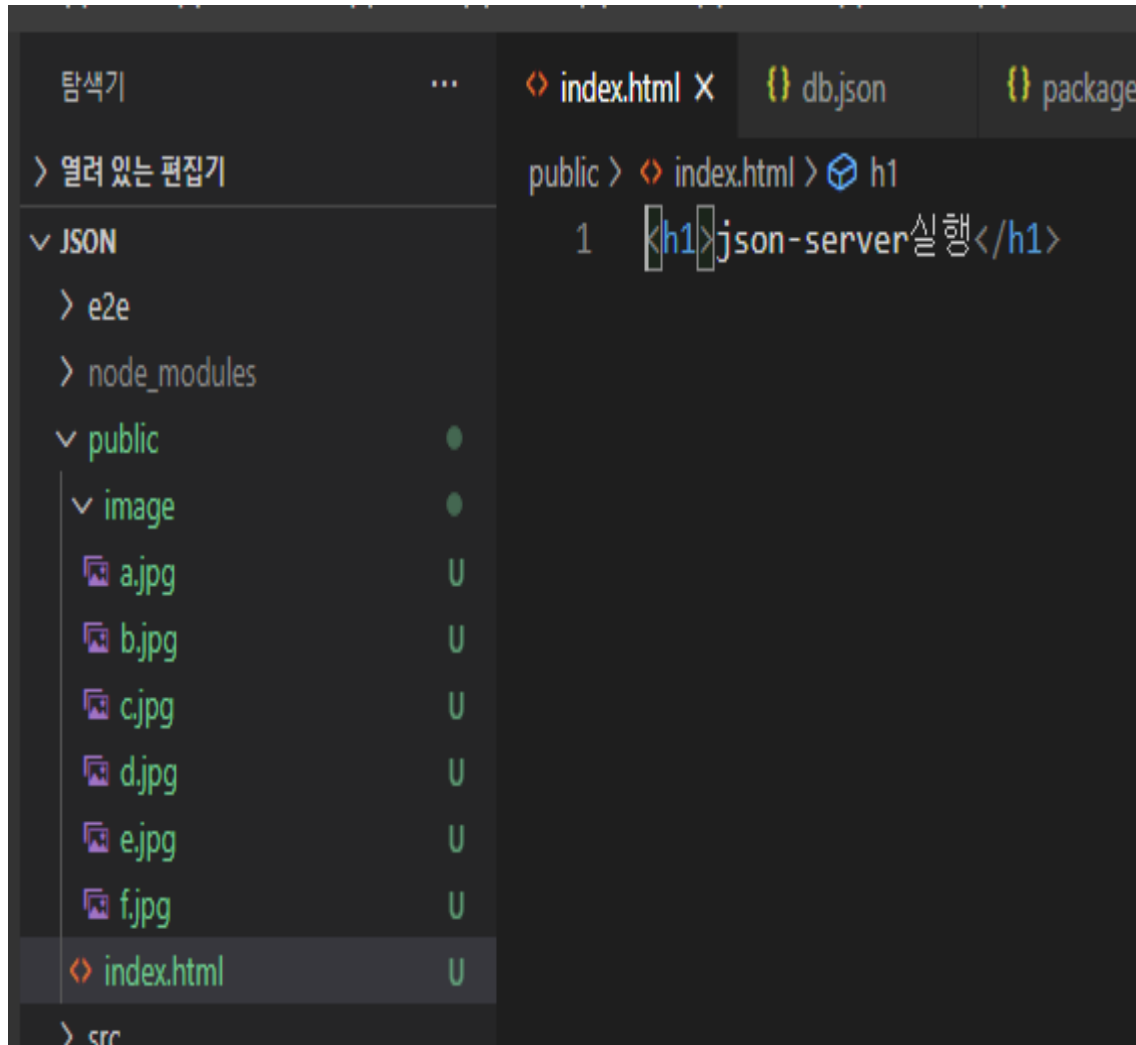
```
localhost:3000/books

[
  {
    "name": "위험한 식탁",
    "img": "a.jpg"
  },
  {
    "name": "공부의 비결",
    "img": "b.jpg"
  },
  {
    "name": "오메르타",
    "img": "c.jpg"
  },
  {
    "name": "행복의 비결",
    "img": "d.jpg"
  },
  {
    "name": "해커스 토익",
    "img": "e.jpg"
  },
  {
    "name": "도로 안내서",
    "img": "f.jpg"
  }
]
```

```
db.json
db.json > {} db.json > ...
1 {
2   "todos": [
3     {
4       "id": 1,
5       "title": "Read SitePoint article",
6       "complete": false
7     },
8     {
9       "id": 2,
10      "title": "Clean inbox",
11      "complete": false
12    },
13    {
14      "id": 3,
15      "title": "Make restaurant reservation",
16      "complete": false
17    }
18  ],
19  "books": [{ "name": "위험한 식탁", "img": "a.jpg"},
20            { "name": "공부의 비결", "img": "b.jpg"},
21            { "name": "오메르타", "img": "c.jpg"},
22            { "name": "행복의 비결", "img": "d.jpg"},
23            { "name": "해커스 토익", "img": "e.jpg"},
24            { "name": "도로 안내서", "img": "f.jpg"}
25  ]
26 }
```



❑ 서버의 수정(json/public/image폴더 생성, index.html생성)





□ *Json서버 접속 component의 작성, HttpClient, FormsModule app.module에 추가*

1. app 모듈의 수정

```
src > app > TS app.module.ts > AppModule
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { HttpClientModule } from '@angular/common/http';
7  import { FormsModule } from '@angular/forms';
8  @NgModule({
9    declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule,
14     AppRoutingModule,
15     HttpClientModule,
16     FormsModule
17   ]
18 })
19 export class AppModule {}
```

□ 4) JSON 서버 접속 컴포넌트 구현

2) service 생성 및 코드 구현

```
C:\angular\my-app2>ng g service book  
CREATE src/app/book.service.spec.ts (347 bytes)  
CREATE src/app/book.service.ts (133 bytes)
```

```
book.service.ts X  
/-app2 > src > app > TS book.service.ts > ...  
1  import { Injectable } from '@angular/core';  
2  import { HttpClient } from '@angular/common/http';  
3  
4  import { map } from 'rxjs/operators';  
5  
6  @Injectable({  
7    providedIn: 'root'  
8  })  
9  export class BookService {  
10  
11    constructor(public http: HttpClient) { }  
12  
13    send(){  
14      var url = "http://localhost:3000/books";  
15      return this.http.get(url)  
16        .pipe(map(res=>res));  
17    }  
18  }
```

□ 4) JSON 서버 활용한 HTTP 모듈 사용 실습

4) 컴포넌트 구현

```
$ app.component.ts ×
my-app2 > src > app > TS app.component.ts > ...
 1  import { Component } from '@angular/core';
 2  import { BookService } from '../book.service';
 3
 4  @Component({
 5    selector: 'app-root',
 6    templateUrl: './app.component.html',
 7    styleUrls: ['./app.component.css'],
 8    providers: [BookService]
 9  })
10  export class AppComponent {
11    title = 'my-app';
12    items;
13    constructor(private service: BookService) {}
14
15    send(){
16      this.service.send()
17        .subscribe( res=>{this.items = res;},
18                    error=>console.log(error)
19                  );
20    }
21  }
```

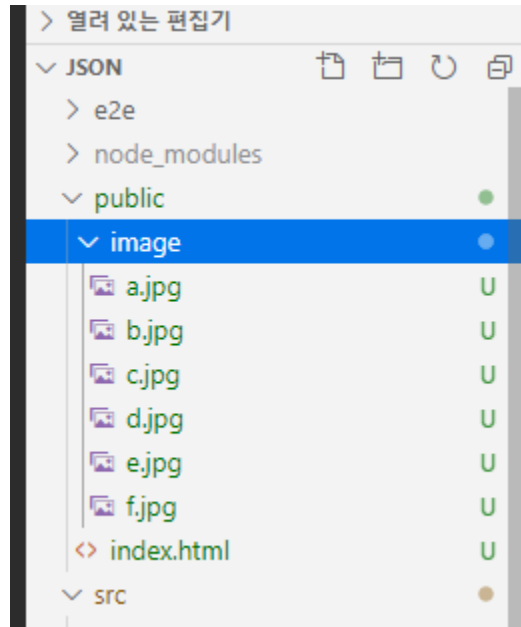
□ 4) JSON 서버 활용한 HTTP 모듈 사용 실습



5) template 구현

```
app.component.html ×
y-app2 > src > app > <> app.component.html > ...
1  <h1>JSON 서버를 활용한 HTTP 서비스</h1>
2  <button (click)="send()">요청</button>
3  <table border="1">
4      <tr>
5          <th>도서명</th>
6          <th>이미지</th>
7      </tr>
8      <tr *ngFor="let item of items">
9          <th>{{item.name}}</th>
10         <th></th>
12     </tr>
13 </table>
```

□ 이미지배치 및 서버 가동, 컴포넌트 동작



```
C:\angular_chul>cd json
C:\angular_chul\json>npm run json-server
> json@0.0.0 json-server C:\angular_chul\json
> json-server --watch db.json

#{^_^}/ hi!

Loading db.json
Done
```

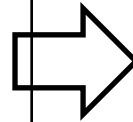
(c) 2020 Microsoft Corporation. All rights reserved.

```
C:\angular_chul\chul-app>ng serve --open
0% compiling
Compiling @angular/common/http : es2015 as esm2015
```

□ 4) JSON 서버 활용한 HTTP 모듈 사용 실습



6) 실행결과



JSON 서버를 활용한 HTTP 서비스	
요청	
도서명	이미지
위험한 식탁	
공부의 비결	
오메르타	
행복의 비결	



□ 5) JSON 서버 활용과 book 클래스 사용

1) Book.ts의 작성

```
src > app > TS book.ts > Book > img
1 export class Book{
2     name:string;
3     img:string;
4 }
```

1) app.module의 import

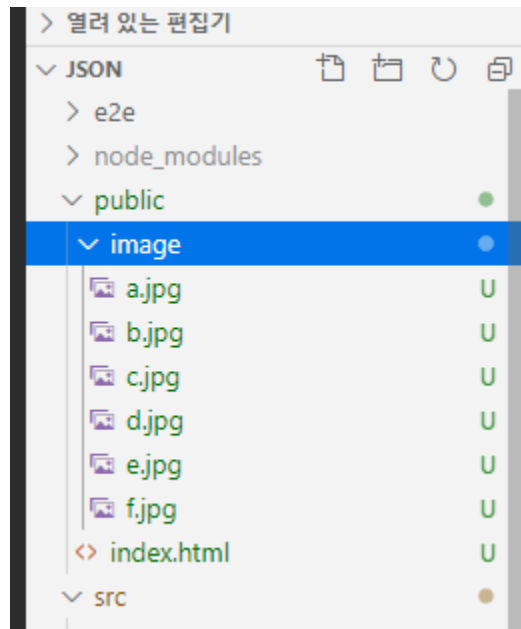
```
src > app > TS app.module.ts > AppModule
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { FormsModule } from '@angular/forms';
4 import { HttpClientModule } from '@angular/common/http';
5 import { AppRoutingModule } from './app-routing.module';
6 import { AppComponent } from './app.component';
7
8 @NgModule({
9     declarations: [
10         AppComponent
11     ],
12     imports: [
13         BrowserModule,
14         AppRoutingModule,
15         HttpClientModule, FormsModule
16     ],
17     providers: [],
18     bootstrap: [AppComponent]
19 })
20 export class AppModule {}
```

□ 5) JSON 서버 활용과 book 클래스 사용

2) BookService의 수정

```
TS book.service.ts X TS book.ts TS app.component.ts
src > app > TS book.service.ts > ...
 1  import { Injectable } from '@angular/core';
 2  import { HttpClient } from '@angular/common/http';
 3  import { map } from 'rxjs/operators'; //import
 4  import { Book } from './book';
 5  import { Observable } from 'rxjs';
 6  @Injectable({
 7    providedIn: 'root'
 8  })
 9  export class BookService {
10
11    constructor(public http: HttpClient) { }
12    send(): Observable<Book[]>{
13      var url="http://localhost:3000/books";
14      return this.http.get(url)
15        .pipe(map(res=>{console.log(res); return res as Book[]})));
16    }
17  }
18
```

□ 서버 가동, 컴포넌트 동작



```
C:\angular_chul>cd json
C:\angular_chul\json>npm run json-server
> json@0.0.0 json-server C:\angular_chul\json
> json-server --watch db.json

#{^_^}/ hi!

Loading db.json
Done
```

(c) 2020 Microsoft Corporation. All rights reserved.

```
C:\angular_chul\chul-app>ng serve --open
0% compiling
Compiling @angular/common/http : es2015 as esm2015
```

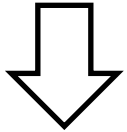
□ 4) JSON 서버 활용한 HTTP 모듈 사용 실습

7) 검색 기능 추가

JSON 서버를 활용한 HTTP 서비스

검색도서명: 요청

도서명 이미지



JSON 서버를 활용한 HTTP 서비스

검색도서명: 위험한 식탁 요청

도서명	이미지
위험한 식탁	

<> app.component.html X

src > app > <> app.component.html > button

```
1 <h1>jSON서버를 요청하는 HTTP서비스</h1>
2 검색도서명: <input type="text" #kkk>
3 <button (click)="send(kkk.value)">send</button>
4 <table border="1">
```

TS book.service.ts X

<> app.component.html

TS app.module.ts

TS book.ts

TS app.

src > app > TS book.service.ts > BookService

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpParams } from '@angular/common/http';
3 import { map } from 'rxjs/operators'; //import
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class BookService {
9   constructor(public http: HttpClient) { }
10  send(kkk){
11    var url="http://localhost:3000/books";
12    const params= new HttpParams().set("name", kkk);
13    if(kkk.length==0){
14      return this.http.get(url).pipe(map(res=>res));
15    }else{
16      return this.http.get(url, {params})
17        .pipe(map(res=>res));
18    }
19  }
20 }
```

```

}
}
export class AppComponent {
  title = 'chul-app';
  constructor(public service:BookService){}
  //then(성공,실패)
  items;
  send(kkk){
    this.service.send(kkk).subscribe(res=>{this.items=res},
    | | | | | | | | | | | | | | | | error=>{console.log(error)});
  }
}
}

```



수고하셨습니다.
