

[Spring Rest API 실습 2]

메일제목 : 스프링실습17-X팀

압축 소스명 : 소스52.zip

이번 실습은 클라이언트와 서버를 모두 구현하는 실습이다. 또한 팀 실습이기도 하다.
서로 협력하고 논의해 가면서 함께 구현한다. 클라이언트는 HTML 에 JavaScript 로 구현한다.

- (1) 다음에 제시된 필드를 갖는 RepresentationModel<ComicActorModel>을 상속하는
ComicActorModel 클래스를 생성한다.

String name;

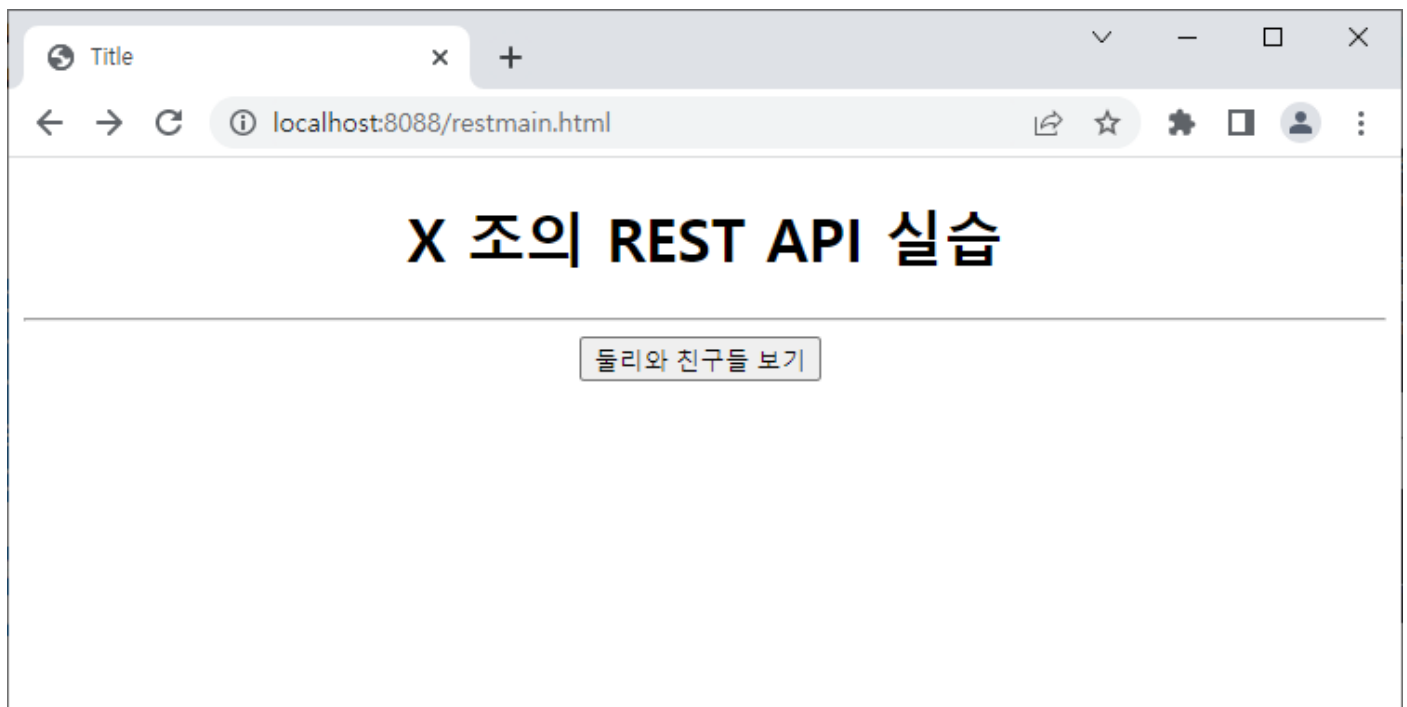
- (2) 다음에 제시된 필드를 갖는 ComicActor 클래스를 생성한다.

String name;

String addr;

String image;

- (3) 다음 화면의 restmain.html 이라는 파일을 생성한다. (static)



(4) Hateoas 를 만족하게 응답하는 컨트롤러 클래스를 구현한다.

@RestController

```
public class HateoasRestController {
```

```
    @GetMapping("/all")
```

```
    public CollectionModel<ComicActorModel> allActor() {
```

```
        "둘리", "또치", "도우너"를 가지고 각각 ComicActorModel 객체를 생성한 다음  
        생성된 ComicActorModel 객체에 자세한 정보를 제공하는 URL(각각 아래 메서드에 대한  
        URL) 정보를 추가한 다음 CollectionModel 객체에 담아 응답한다.
```

```
    }
```

```
    @GetMapping("/dooly")
```

```
    public ResponseEntity<ComicActor> getDooly() {
```

```
        둘리 정보("둘리", "쌍문동", "dooly.jpg")를 가지고 ComicActor 객체를 생성한 다음  
        ResponseEntity 객체에 담아서 응답한다.
```

```
    }
```

```
    @GetMapping("/ddochi")
```

```
    public ResponseEntity<ComicActor> getDdochi() {
```

```
        또치 정보("또치", "아프리카", "ddochi.jpg")를 가지고 ComicActor 객체를 생성한 다음  
        ResponseEntity 객체에 담아서 응답한다.
```

```
    }
```

```
    @GetMapping("/dauner")
```

```
    public ResponseEntity<ComicActor> getDauner() {
```

```
        도우너 정보("도우너", "간파비아", "dauner.jpg")를 가지고 ComicActor 객체를 생성한  
        다음 ResponseEntity 객체에 담아서 응답한다.
```

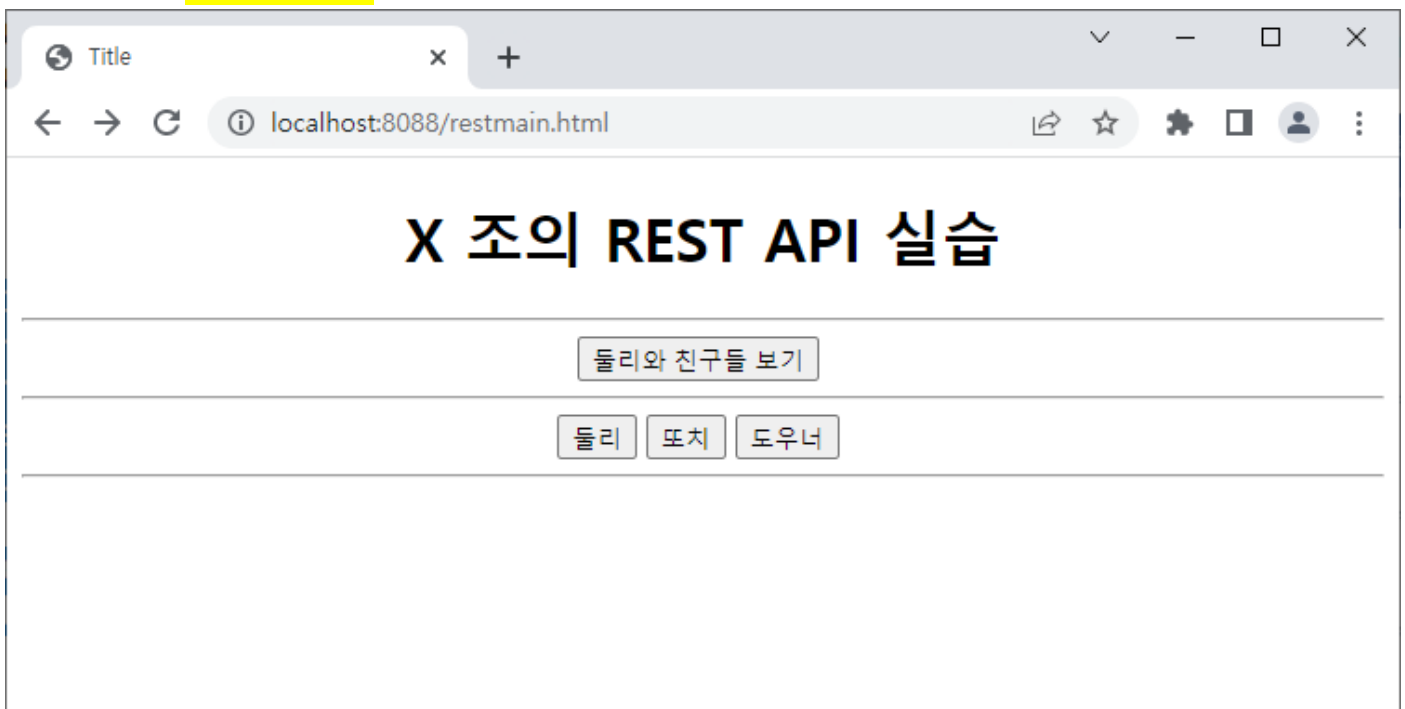
```
    }
```

```
}
```

(5) 3번 화면에서 버튼을 클릭하면 <http://localhost:8088/all> 을 AJAX 로 요청하고 응답되는 내용에서

```
{
  _embedded : {
    comiclist : [
      {
        name : "둘리",
        _links : { detail : { href : "http://localhost:8088/dooly" } }
      },
      {
        name : "또치",
        _links : { detail : { href : "http://localhost:8088/ddochi" } }
      },
      {
        name : "도우너",
        _links : { detail : { href : "http://localhost:8088/dauner" } }
      }
    ]
  }
}
```

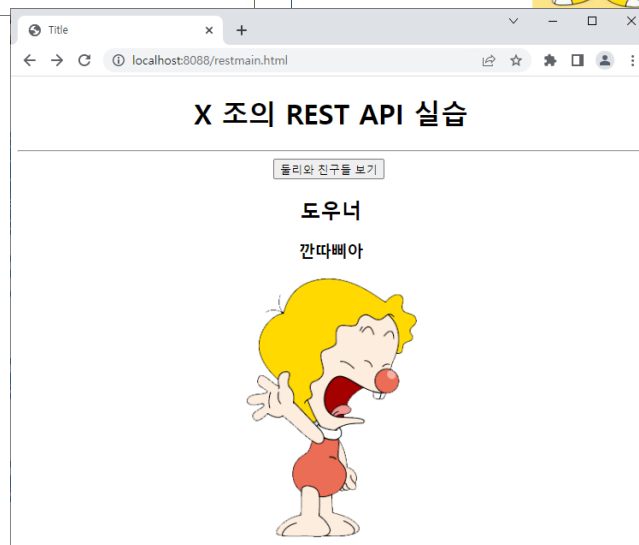
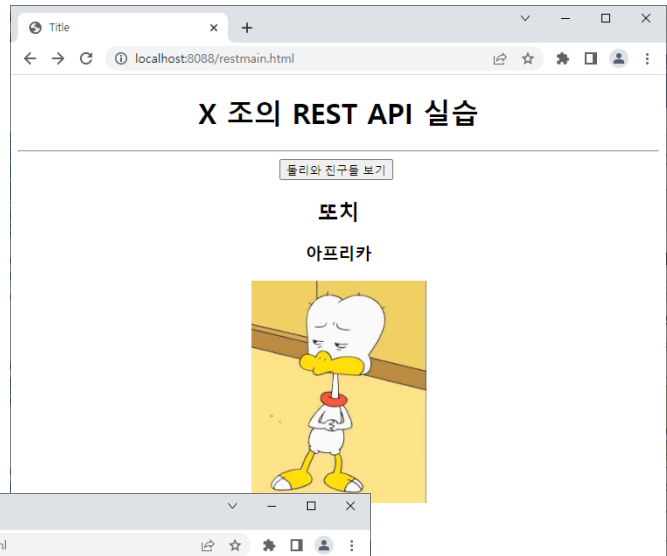
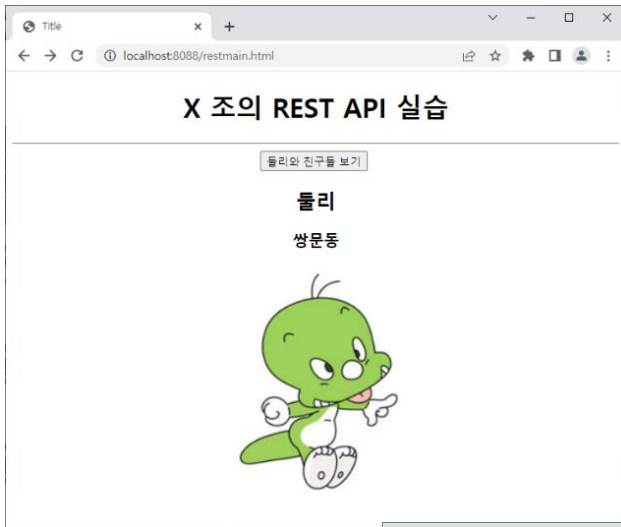
각 배우들의 name과 href 정보를 추출하여 배우 이름으로 버튼을 만들어 다음과 같이 출력한다.



버튼이 클릭되면 추출된 href 정보를 가지고 AJAX 로 요청하고 응답되는 다음 내용에서

```
{
  name : "둘리",
  addr : "쌍문동",
  image : "dooly.jpg"
}
```

각 정보를 추출해서 다음 화면을 출력한다.



팀원들 간에 논의해 보자!!!

배우들의 인원수나 정보의 URL 이 변경된다고 할 때 클라이언트 소스의
변경이 필요하겠는가?