## Community 버전에서 프로젝트 만들려면

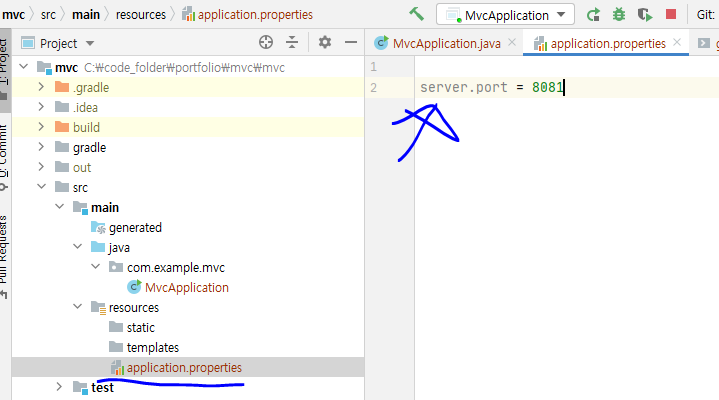
https://start.spring.io/  
에서 프로젝트 만들고, 인텔리제이에서 open 누른 후, 해당 프로젝트의 gradle 선택하면 된다.

## Spring 프로젝트 서버 포트 변경 하는 법

<https://proni.tistory.com/133>

아래 이미지처럼 해당 경로에 입력하면 된다.

server.port = 원하는 포트 번호



## SpringFox Boot Starter 을 maven repository에서 gradle로 붙여 넣을 때

**org.springframework.context.ApplicationContextException: Failed to start bean 'documentationPluginsBootstrapper'; nested exception is java.lang.NullPointerException:**

같은 에러 나는 경우.

텍스트이(가) 표시된 사진

자동 생성된 설명

해결법 적힌 블로그 : <https://ldne.tistory.com/332?category=451445>

문제 원인 : springfox 라이브러리와 스프링boot 라이브러리 버전 호환성 문제였는듯.

프로젝트의 스프링부트 라이브러리 버전을 2.6.1->2.4.0으로 다운하니 아래처럼 잘 동작 함.

텍스트이(가) 표시된 사진

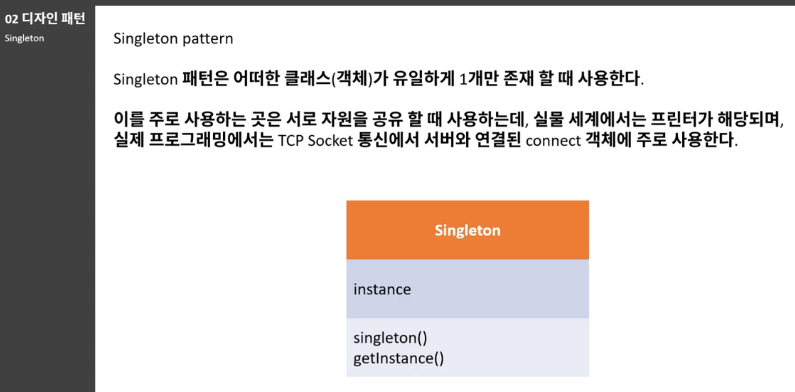
자동 생성된 설명

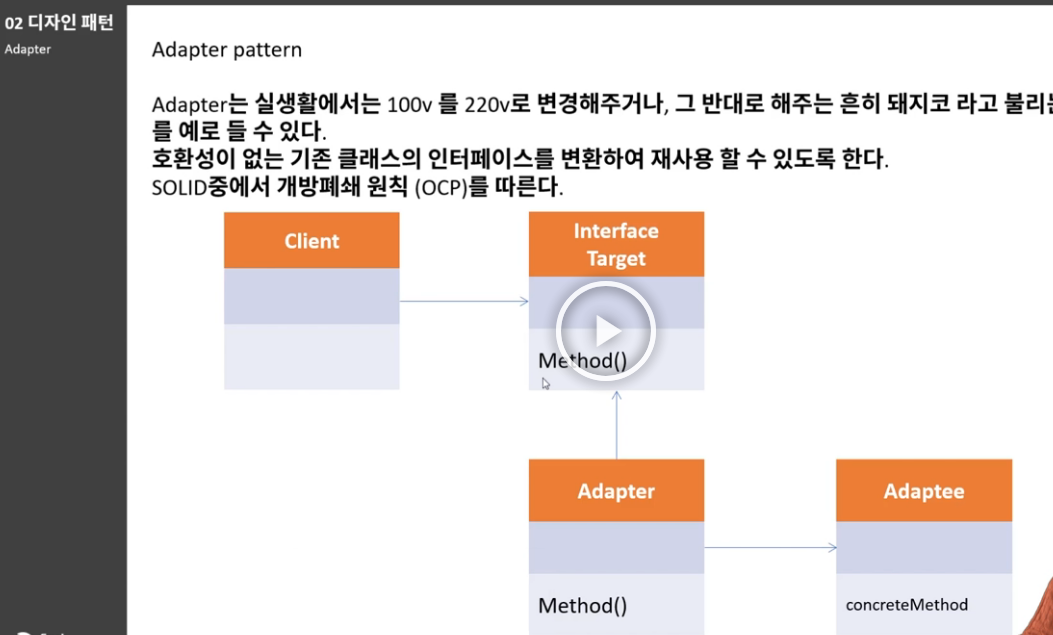
<http://127.0.0.1:8081/swagger-ui/>

텍스트이(가) 표시된 사진

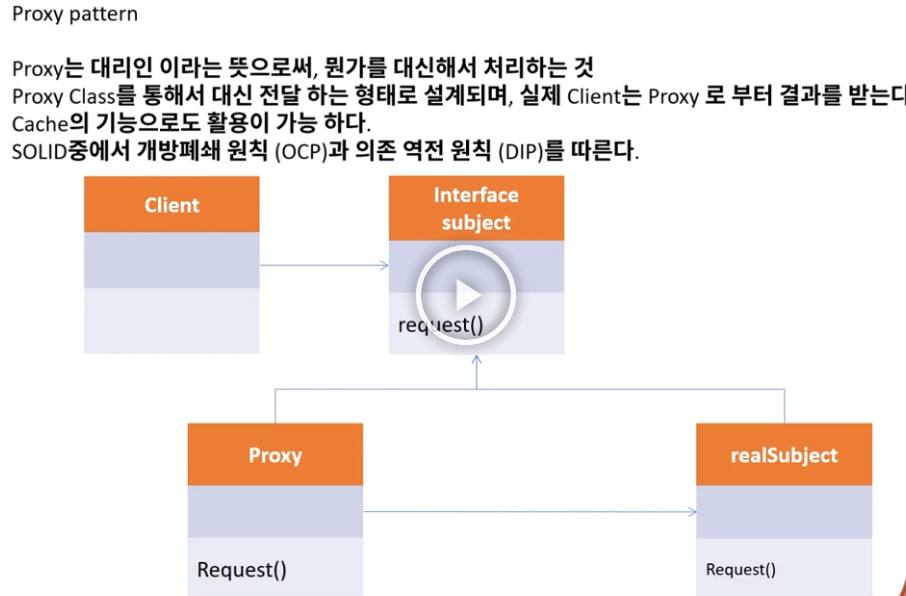
자동 생성된 설명

## 스프링의 Bean이라 불리는 클래스또는 객체들은 싱글톤으로 관리가 된다.



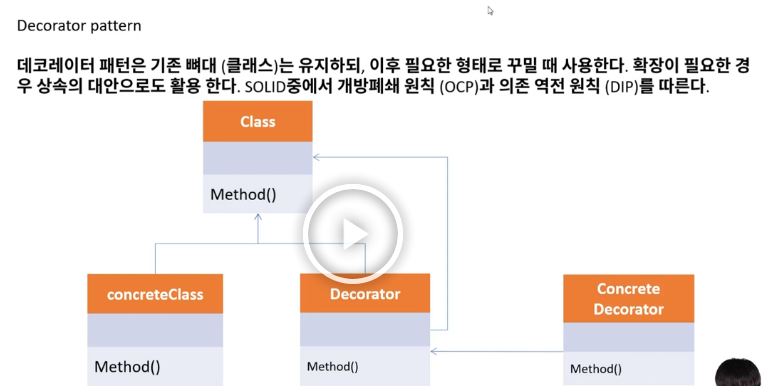


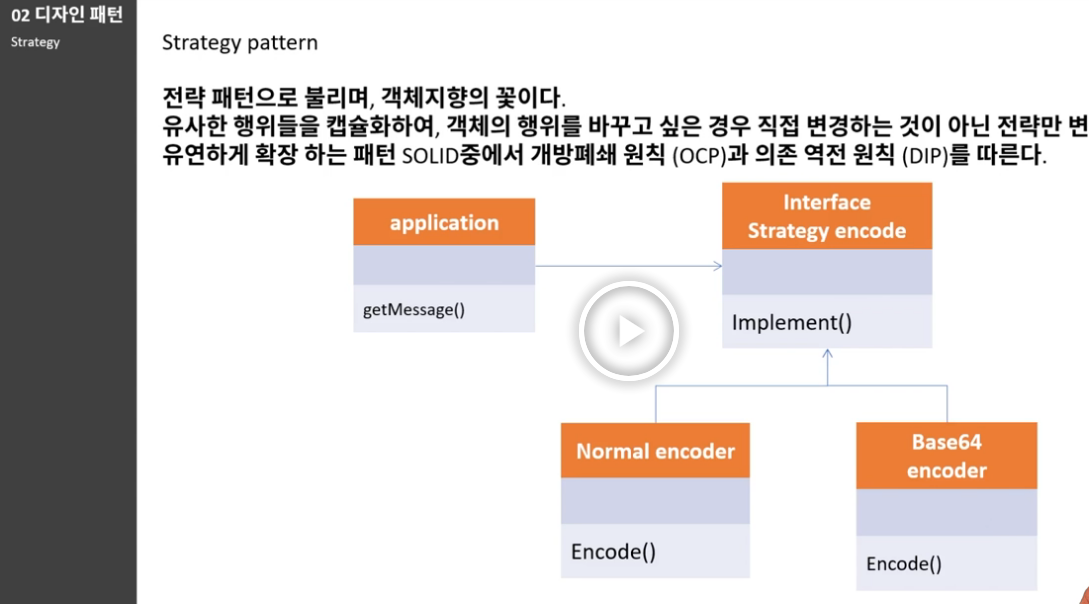
Cache 기능이나 AOP 기능 구현을 Proxy Pattern으로 하는 듯 하다.



안드로이드 같은 곳에서 event listener 같은 기능이 옵저버 패턴의 사용 예시이다.







# 3. 웹 개발 개론

## REST

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

URI에 자원을 식별할 수 있는 정보가 담겨 있어야 한다.

100 번 째 유저

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 실내, 스크린샷, 영수증이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

## Uri 설계 패턴

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

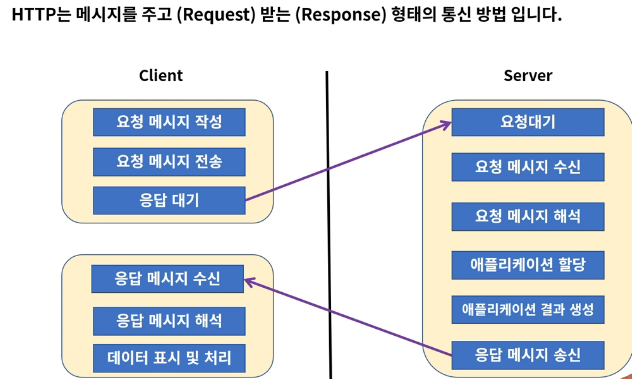
텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

## HTTP Protocol



테이블이(가) 표시된 사진

자동 생성된 설명

멱등성 : 몇 번을 호출하던 같은 정보가 내려오는 지 여부

POST API

테이블이(가) 표시된 사진

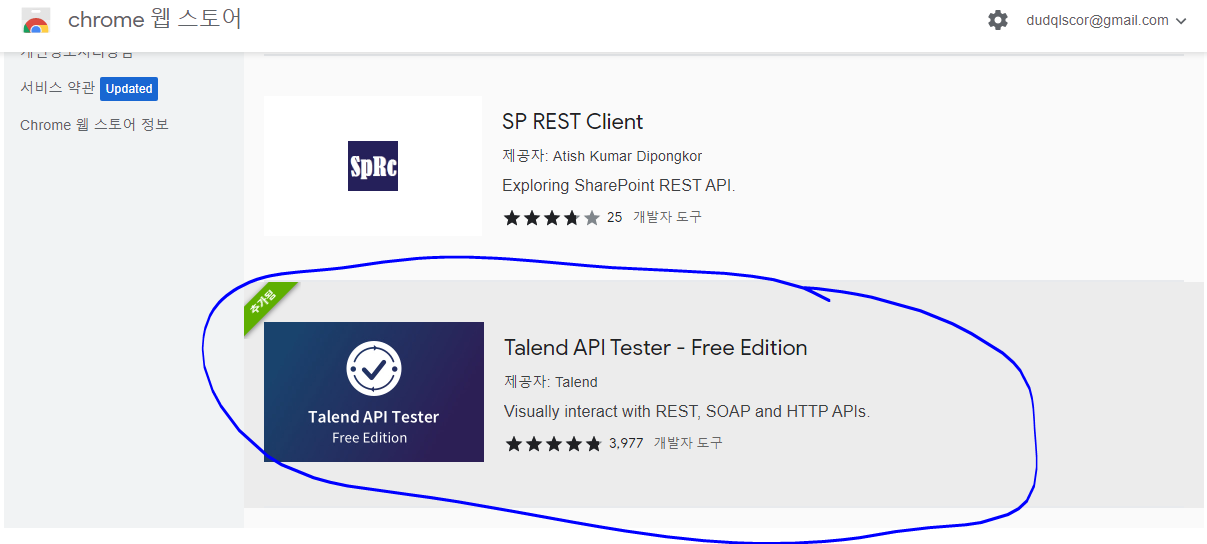
자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

# 스프링 부트 시작하기

### Rest Client 설치하기



저 프로그램 추가해서 테스트 하는데 사용

### GET API

### POST API

JSON 구조

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

{

"accoun":"user01",

"email":"adad@ada.com",

"address":"패스트캠퍼스",

"password":"abcd"

}

테이블이(가) 표시된 사진

자동 생성된 설명

### PUT API

Body JSON 예시 데이터

{

"name":"user01",

"age": 3,

"car\_list":[{"name":"쏘나타",

"car\_number":"1111"

},

{"name":"아반떼",

"car\_number":"121"

}

]

}

테이블이(가) 표시된 사진

자동 생성된 설명

### Delete api

테이블이(가) 표시된 사진

자동 생성된 설명

### Response 내려주기 및 모범사례

//Response 생성하는 동작 순서 : req -> object mapper -> object -> method -> object -> objectmapper -> json -> response

## Object Mapper

@SpringBootTest  
class MvcApplicationTests {  
  
 @Test  
 void contextLoads() throws JsonProcessingException {  
 System.*out*.println("-----------");  
  
 *// Text JSON -> Object  
 // Object -> Text JSON  
  
 //controller req json(text) -> object  
 // response object -> json(text)* var objectMapper = new ObjectMapper();  
  
 *// object -> text  
 // object mapper get method를 활용한다.* var user = new UserReq("aaa","aaa#asda.com", 10);  
 var text = objectMapper.writeValueAsString(user);  
 System.*out*.println(text);  
  
 *// text -> object  
 // object mapper는 default 생성자를 필요로 한다.* var objectUser = objectMapper.readValue(text, UserReq.class);  
 System.*out*.println(objectUser);  
 }  
}

Object Mapper는

1. Object를 text로 바꿀 수 있다.   
   이 과정에서 각 속성의 get method 사용
2. Text를 Object로 바꿀 수 있다.  
   이 과정에서 default 생성자가 필요하다. (set 메서드는 없어도 되는 것 같은데 정확히 왠지는 잘 모르겠음.)

## 스프링의 핵심

텍스트, 명함이(가) 표시된 사진

자동 생성된 설명

## IOC, DI(1)

텍스트이(가) 표시된 사진

자동 생성된 설명

IoC 개념은 내가 객체를 관리하는 것이 아니라 스프링 컨테이너가 객체의 생명주기를 관리해 주는 것

텍스트이(가) 표시된 사진

자동 생성된 설명