Microservices & Spring boot application

RIOT GAMES
Software engineer
예 재 형

강사 소개

LOL (5 시즌)

- 3시즌 Diamond
- 2시즌 Platinum
- 이렐리아, 케일, 람머스

개발자경력

- 정부 프로젝트 2년
- Tmaxsoft 1년
- Riotgames 2년

문의는 어디로?

- Slack (예재형)
- iye@riotgames.com

Micro services & Spring boot

1 Microservices

1. microservices 란?
2. League of Legends 와 함께 살펴보는 Monolithic vs Microservices

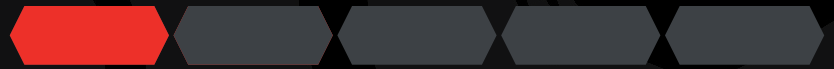
2 Spring boot - 1

1. Spring MVC and now ..
2. Spring vs Spring boot
3. let’s create Spring boot application !
4. HTTP
5. REST & RESTful
6. let’s expose endpoint in your Spring boot
7. let’s support Swagger in your Spring boot

3 Spring boot - 2

1. Concept of DI (Dependency Injection)
2. let’s use open API in your Spring boot
3. let’s use database in your Spring boot

Microservices



What is the microservices

- An architectural style
- An application as a collection of services
- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Comparable to monolithic service

What is the microservices

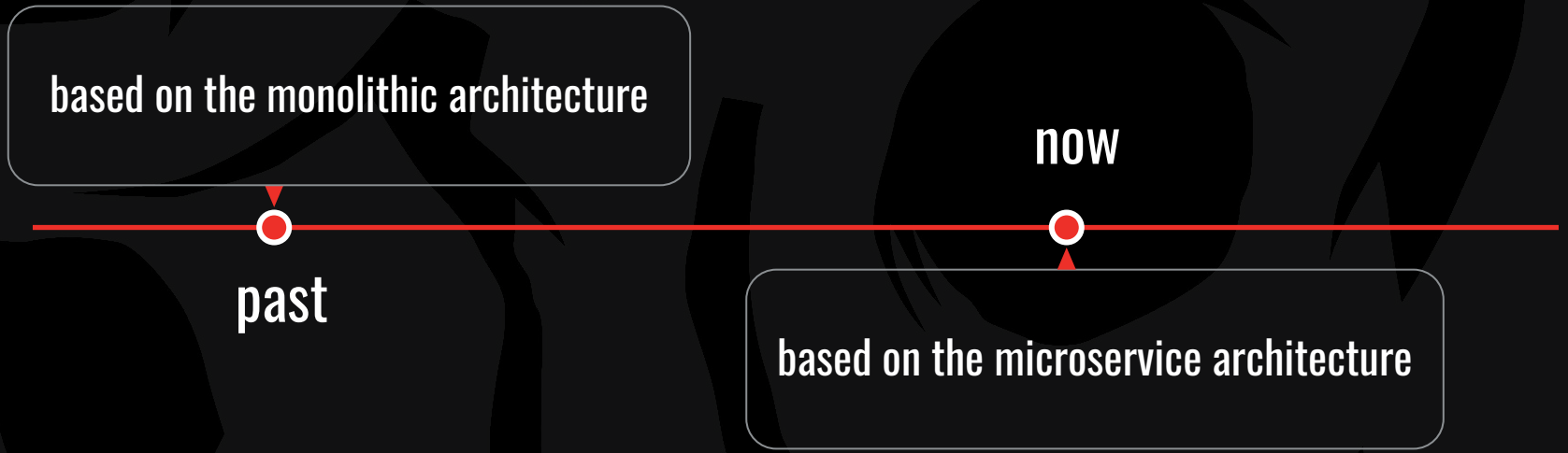
- An architectural style
- An application as a collection of services
- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Comparable to monolithic service

League of legends (platform)

based on the monolithic architecture

past

League of legends (platform)



LOL Basic scenario (Client & Platform)

LOL Basic scenario (Client & Platform)

LOL Basic scenario (Client & Platform)

LOL Basic scenario (Client & Platform)

login request

LOL Basic scenario (Client & Platform)

login request

Authentication

LOL Basic scenario (Client & Platform)

login request

login succeed

Authentication

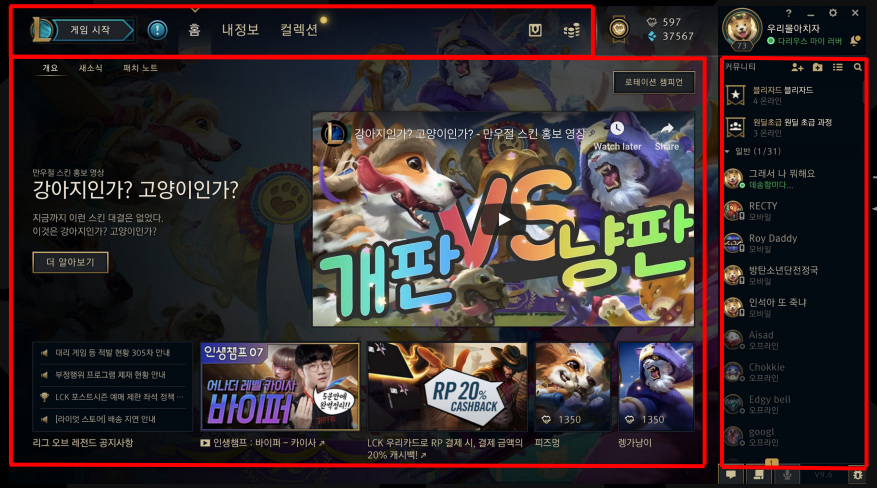
LOL Basic scenario (Client & Platform)

LOL Basic scenario (Client & Platform)

game match service

store service

loot service



lobby service

chat service

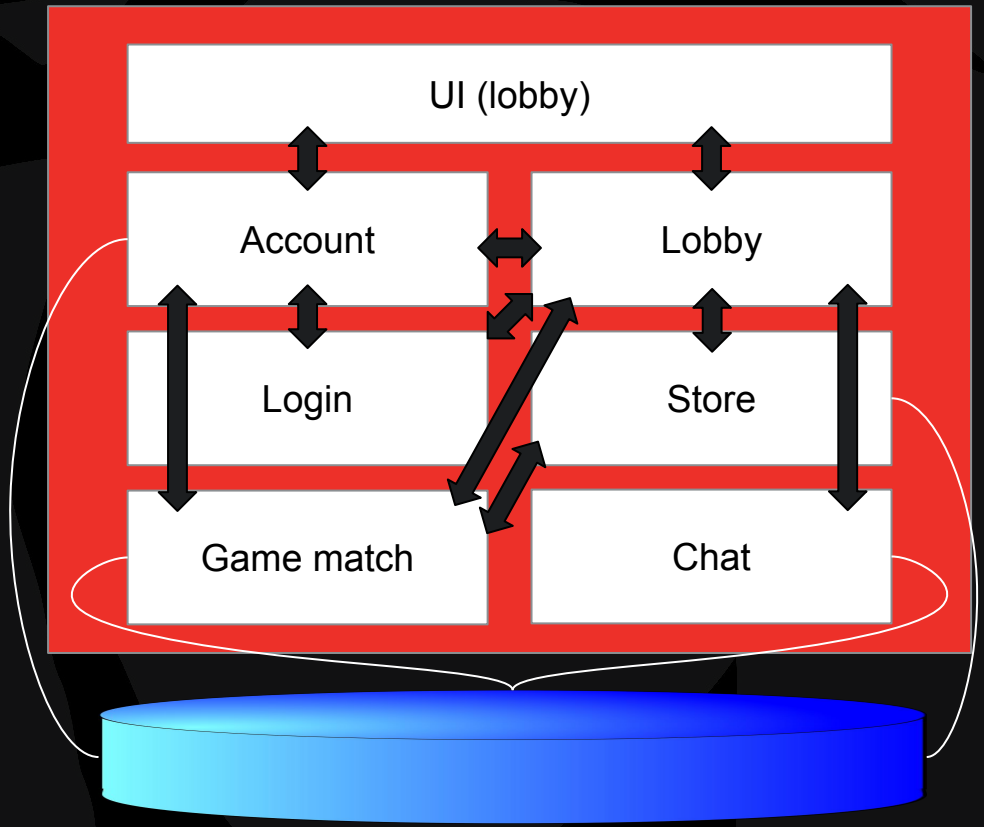
Platform (based on monolithic)

다수의 서비스,

하나의 프로세스,

하나의 데이터베이스

하나의 서버

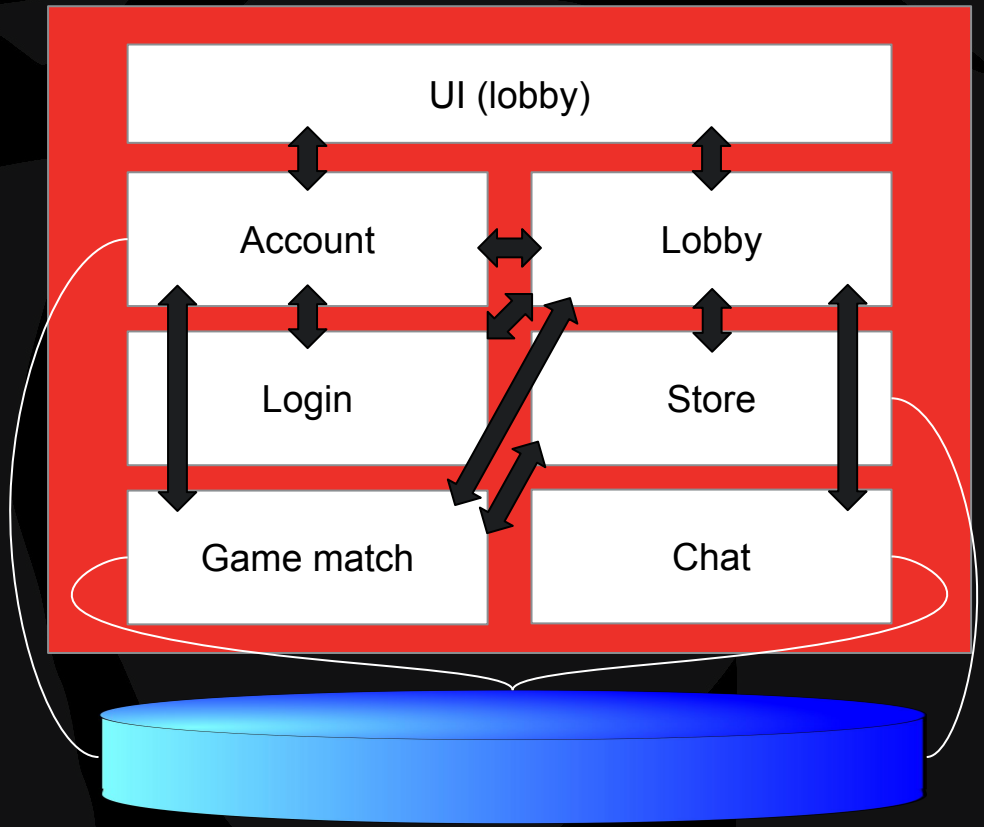


What if ... (runtime)

- ***DB 장애 발생***

어떤 일들이 벌어질까요?

어떻게 해결하죠?

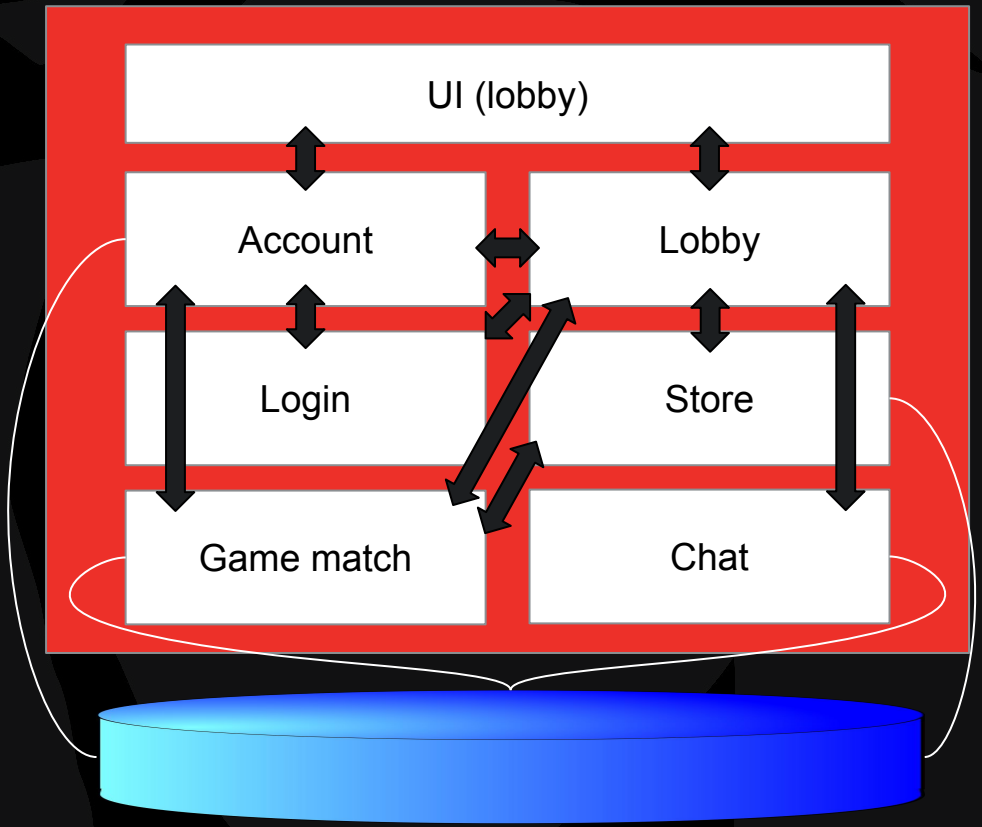


What if ... (runtime)

- *로그인 서비스 장애 발생*
  (by critical bug or library)

어떤 일들이 벌어질까요?

어떻게 해결하죠?

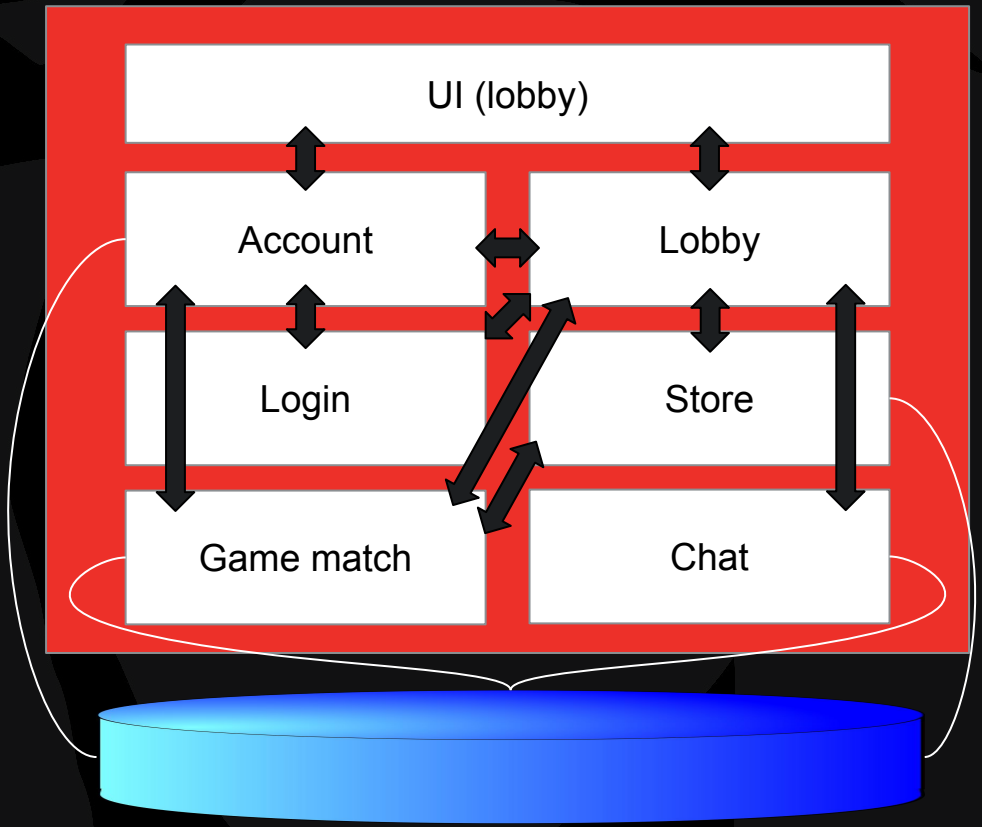


What if ... (development, sustaining)

- ***Game match team에서 신규 기능을 개발해야 함! (조건: Store에서 새롭게 추가될 기능을 사용해야 함)***

*새로운 **Message Queue**를 사용하고 싶다면?*

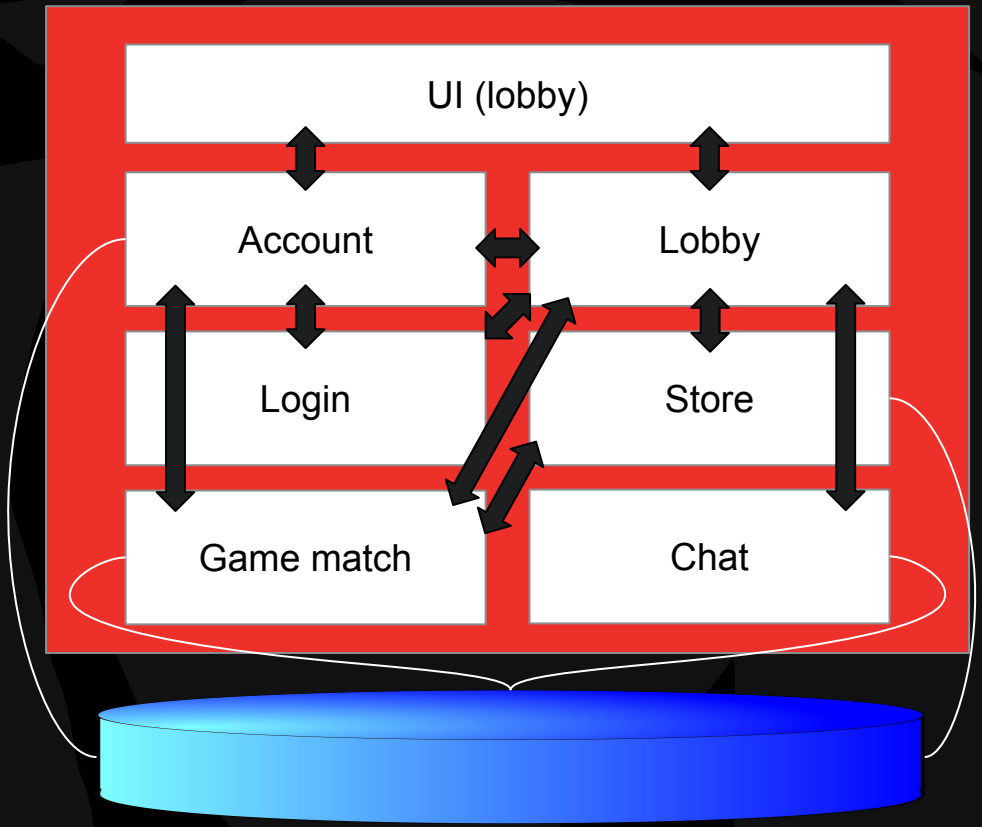
*실컷 기다려서 기능개발을 완료했는데, **Store**의 기능이 크게 변경되면?*



What if ... (development, sustaining)

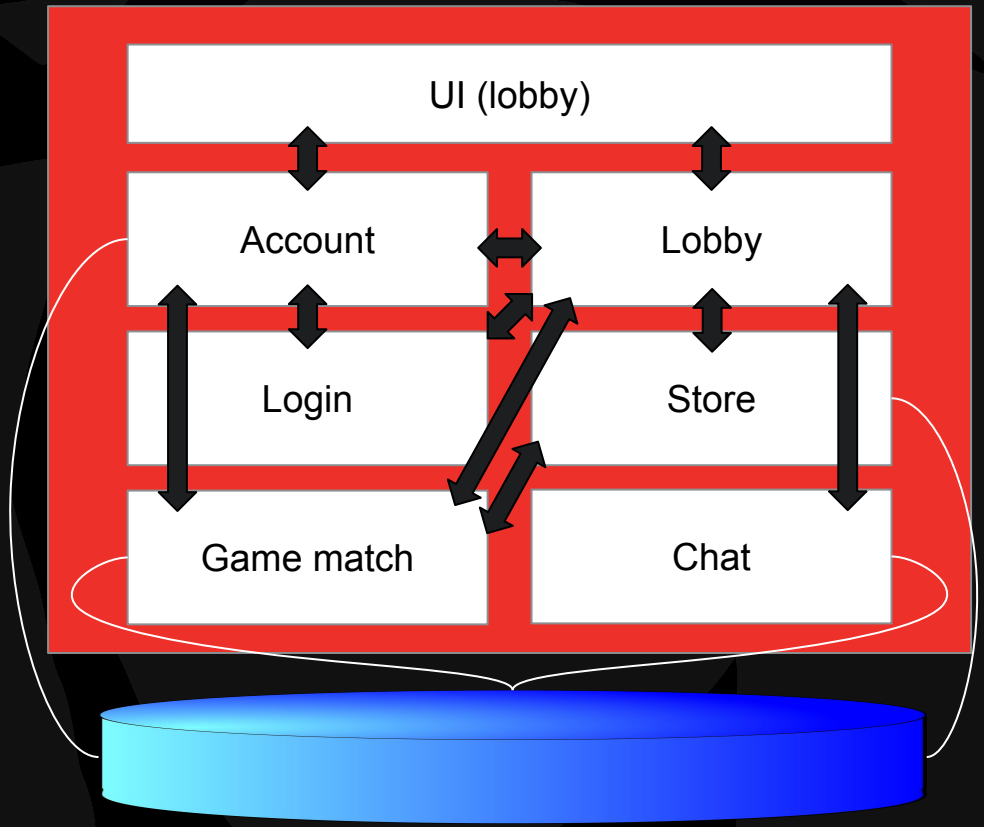
- *신규 개발자 채용 및 합류*

Programming language ?



What if ... (development, sustaining)

... 전부다 단점 뿐인가요 ?



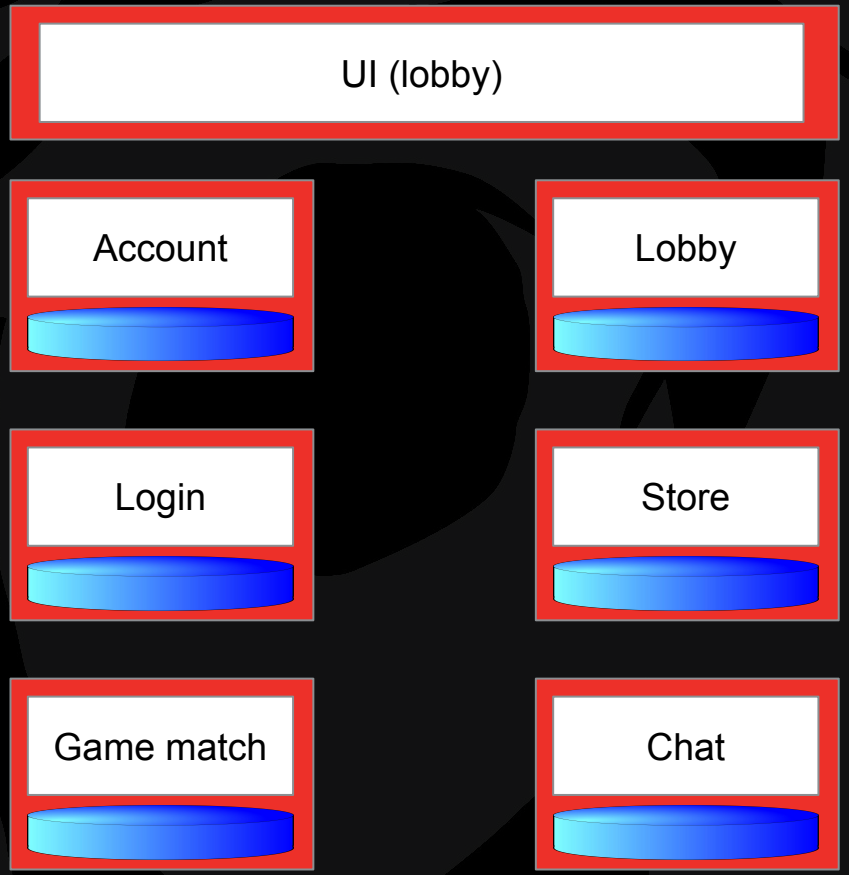
Platform (based on microservices)

다수의 서비스,

다수의 프로세스,

다수의 데이터베이스

다수의 서버



Platform (based on microservices)

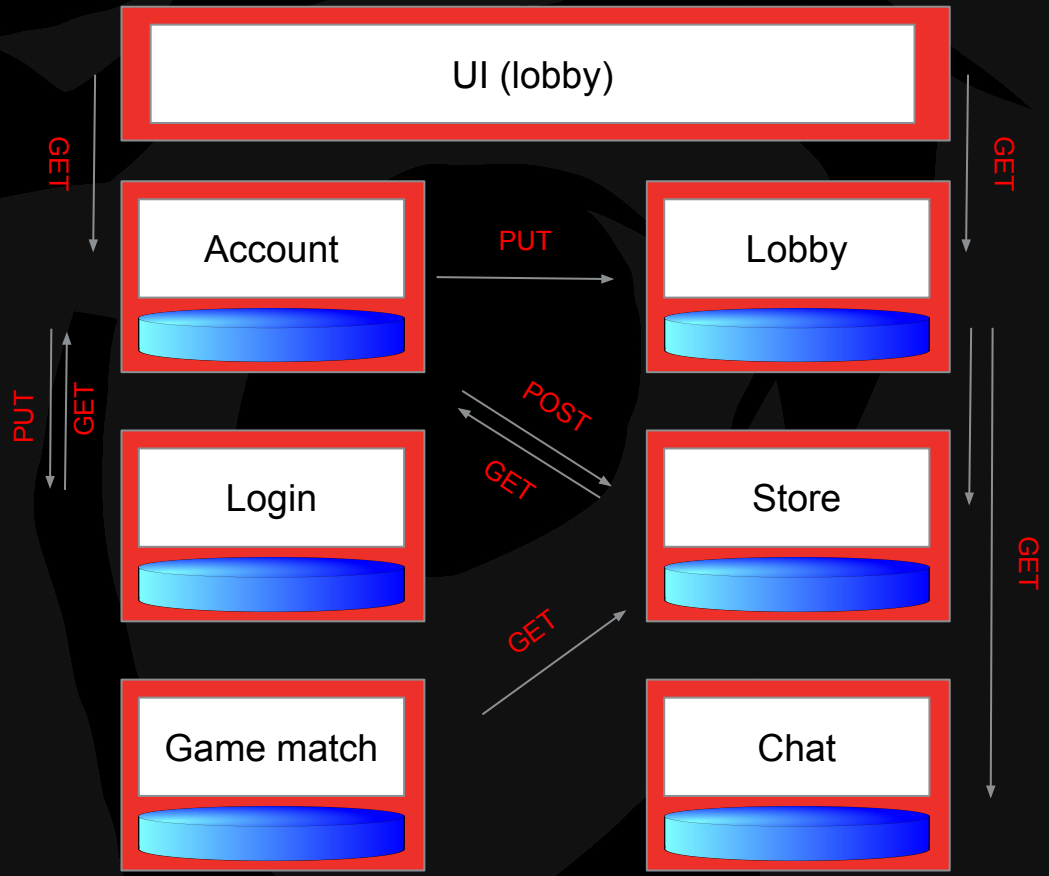
다수의 서비스,

다수의 프로세스,

다수의 데이터베이스

다수의 서버

Well-defined protocol
(ex, HTTP, RPC, RMI … etc)

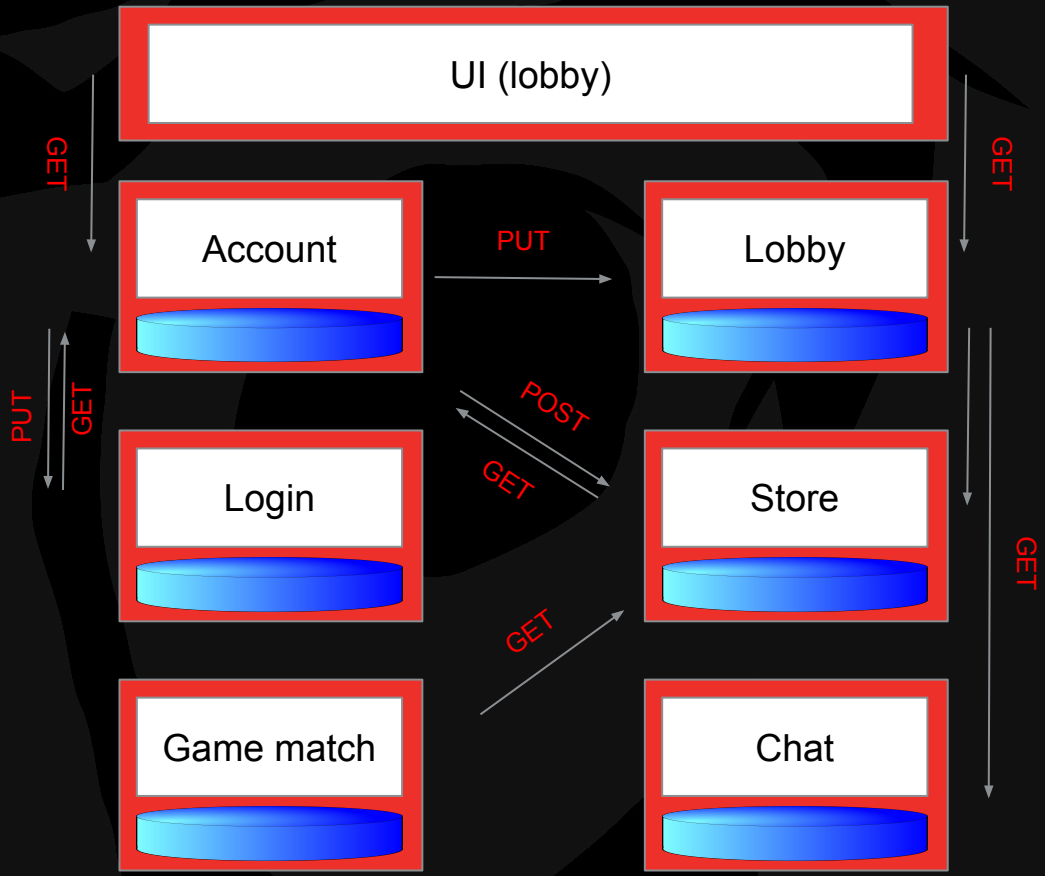


What if ... (runtime)

- ***DB 장애 발생***

어떤 일들이 벌어질까요?

어떻게 해결하죠?

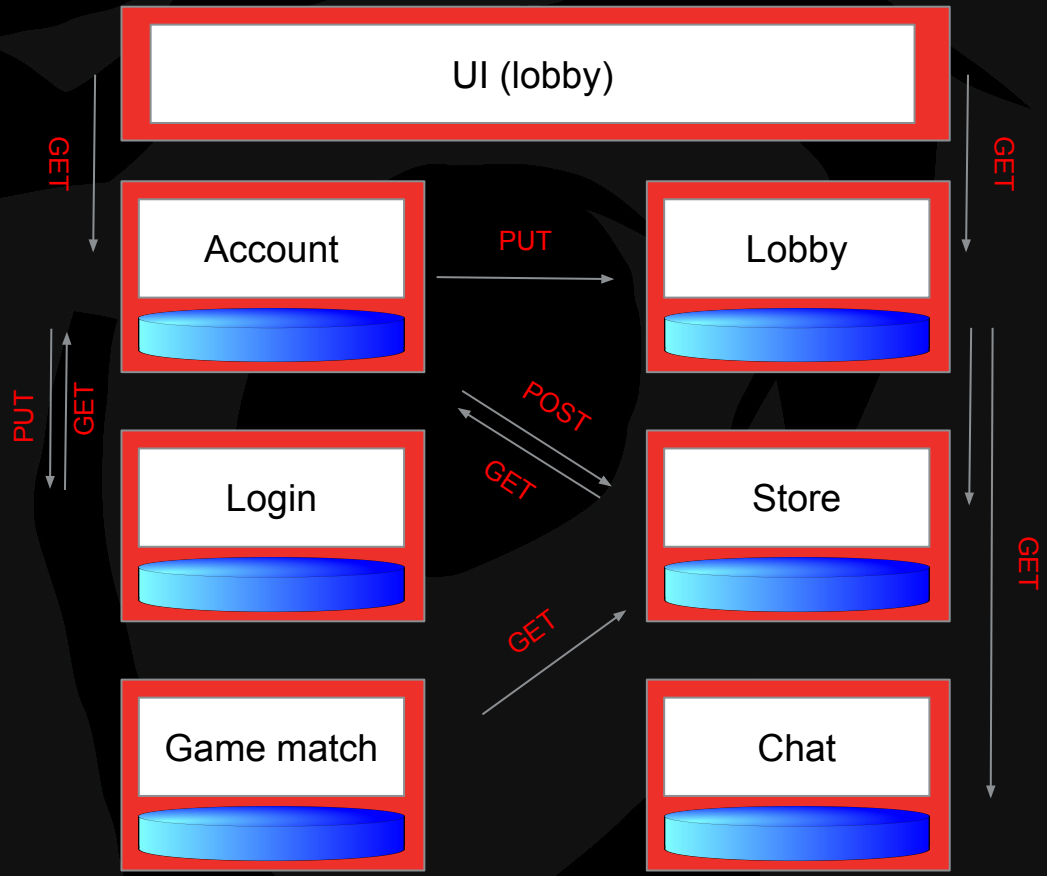


What if ... (runtime)

- *로그인 서비스 장애 발생*
  (by critical bug or library)

어떤 일들이 벌어질까요?

어떻게 해결하죠?

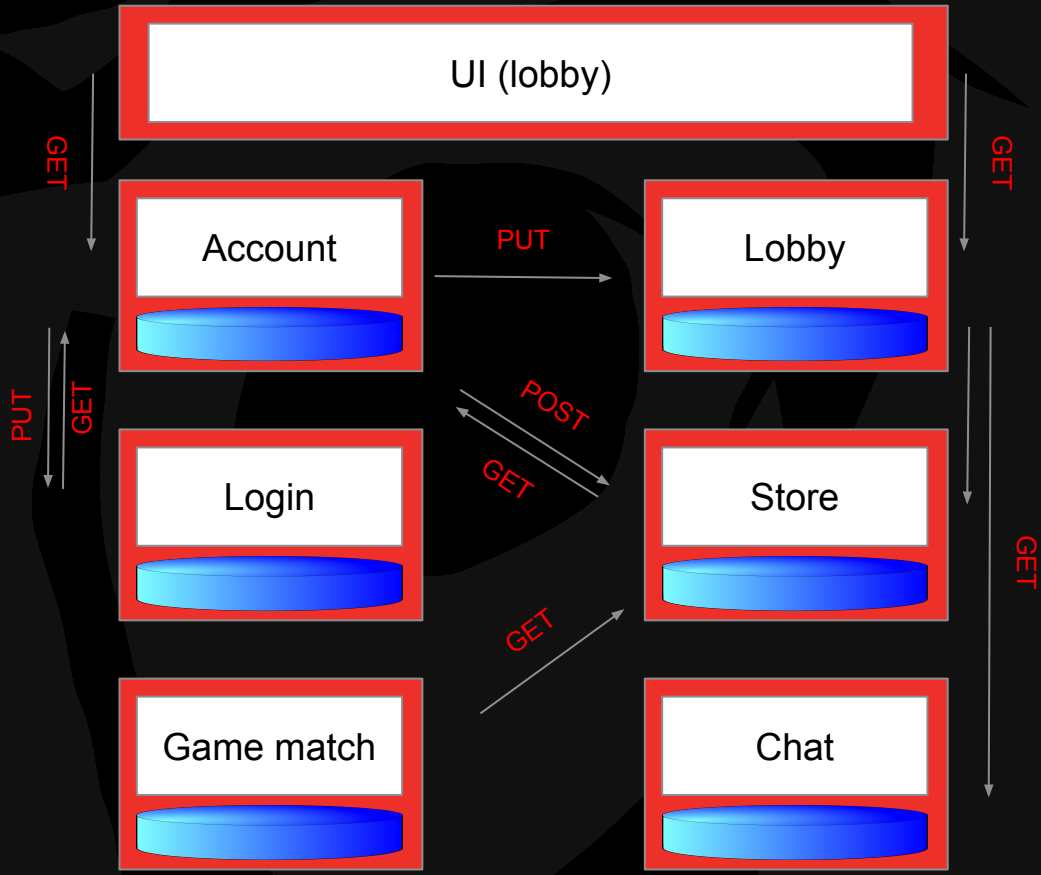


What if ... (development, sustaining)

- ***Game match team에서 신규 기능을 개발해야 함! (조건: Store에서 새롭게 "추가될" 기능을 사용해야 함)***

새로운 Message Queue를 사용하고 싶다면?

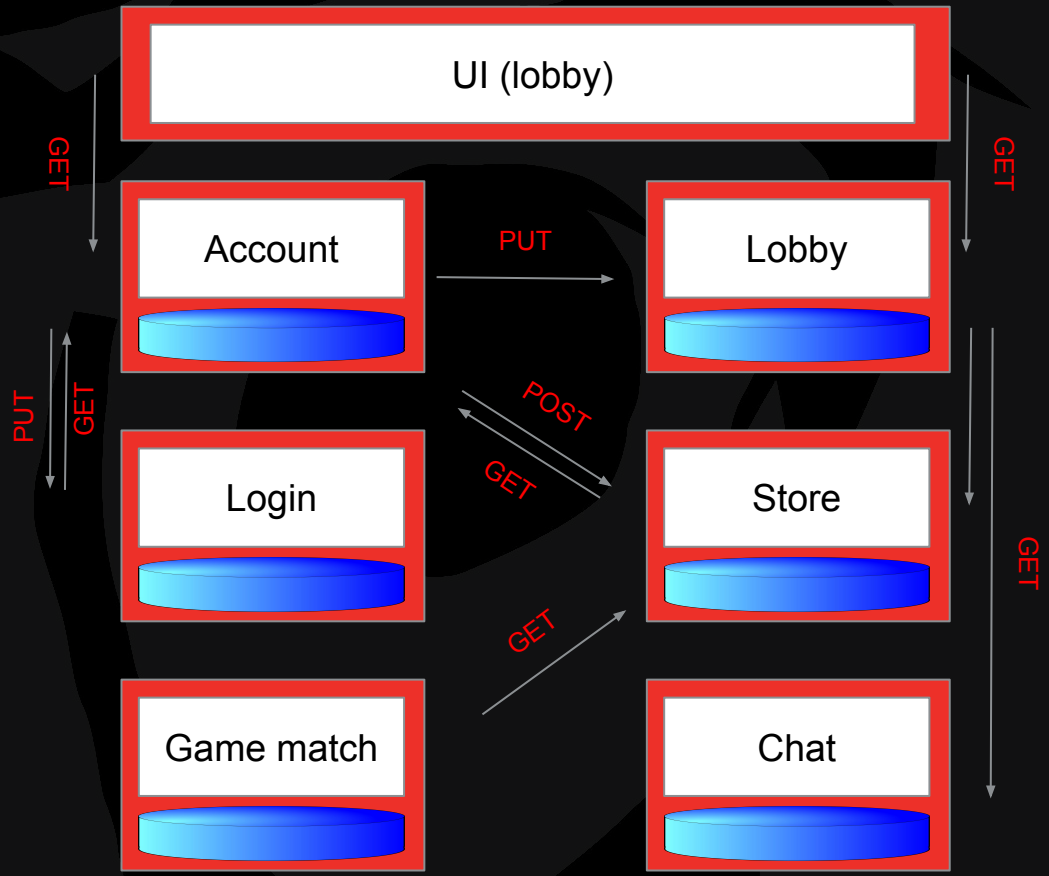
실컷 기다려서 기능개발을 완료했는데, Store의 기능이 크게 변경되면?



What if ... (development, sustaining)

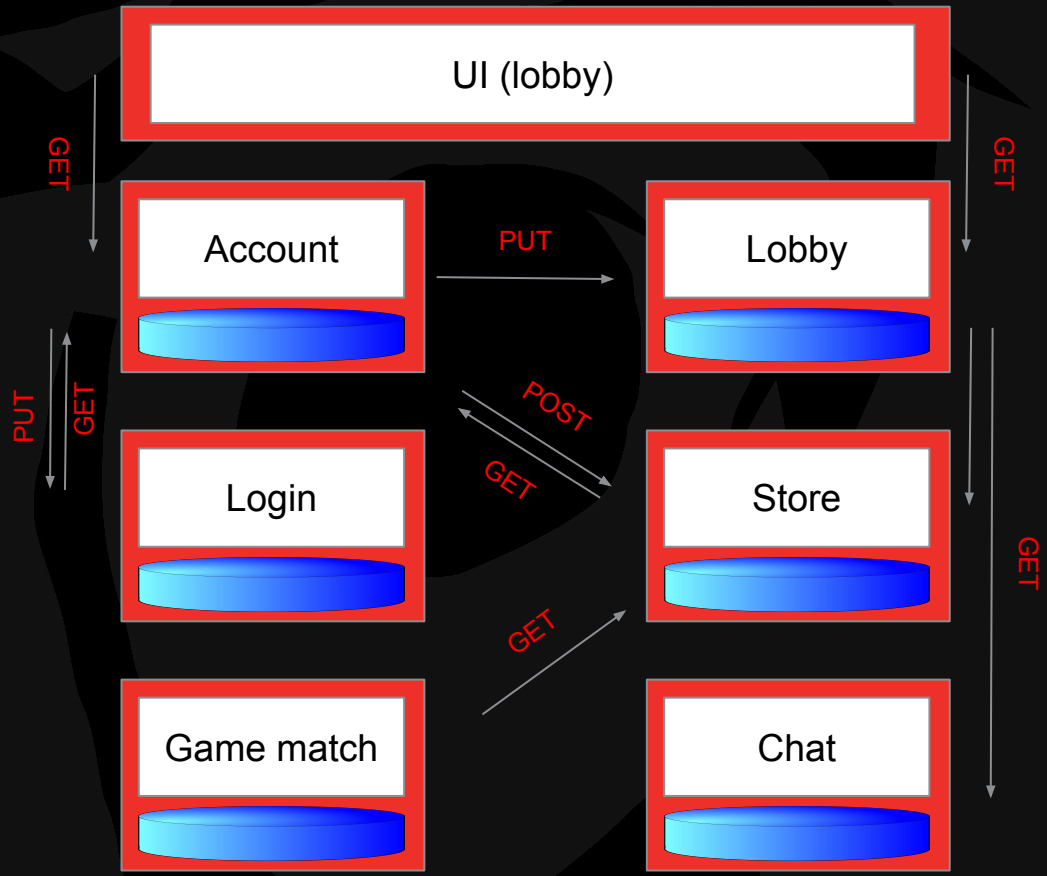
- *신규 개발자 채용 및 합류*

Programming language ?

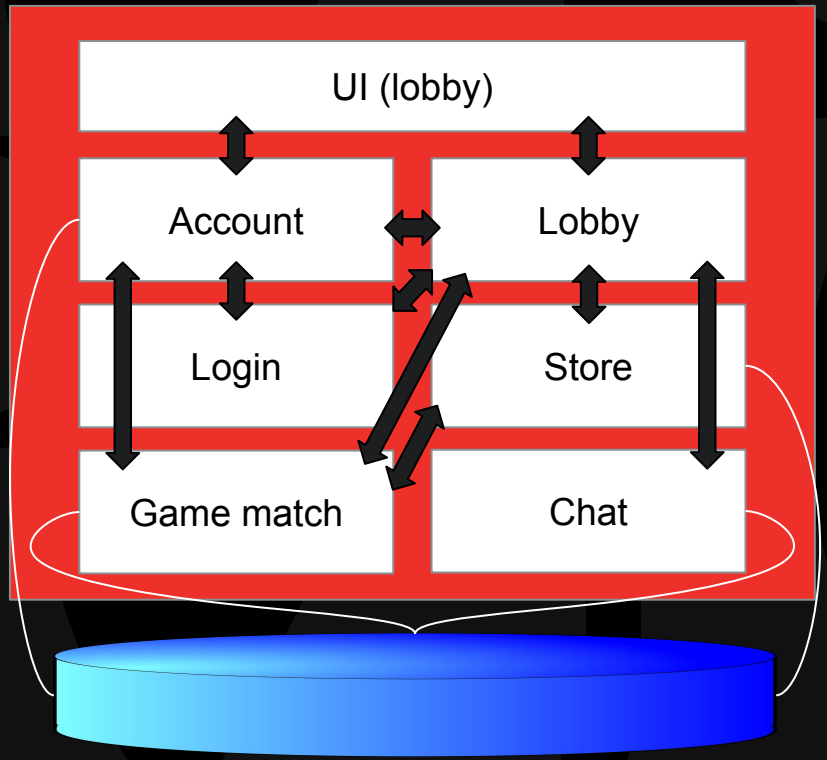


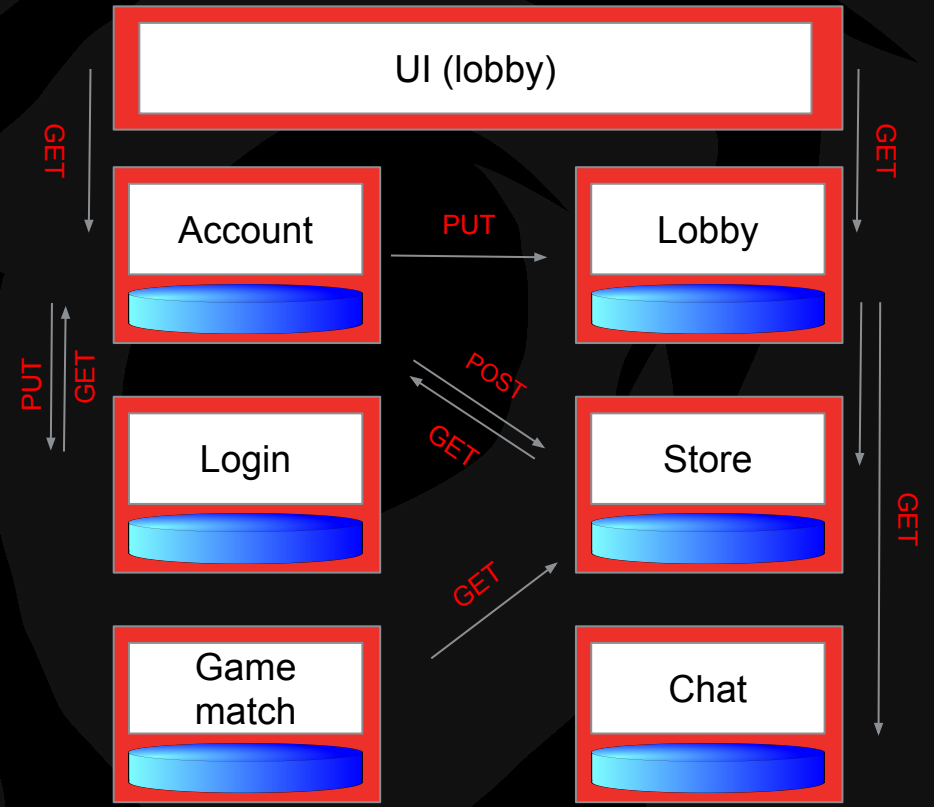
What if ... (development, sustaining)

... 전부다 장점 뿐인가요 ?



Monolithic vs Microservices





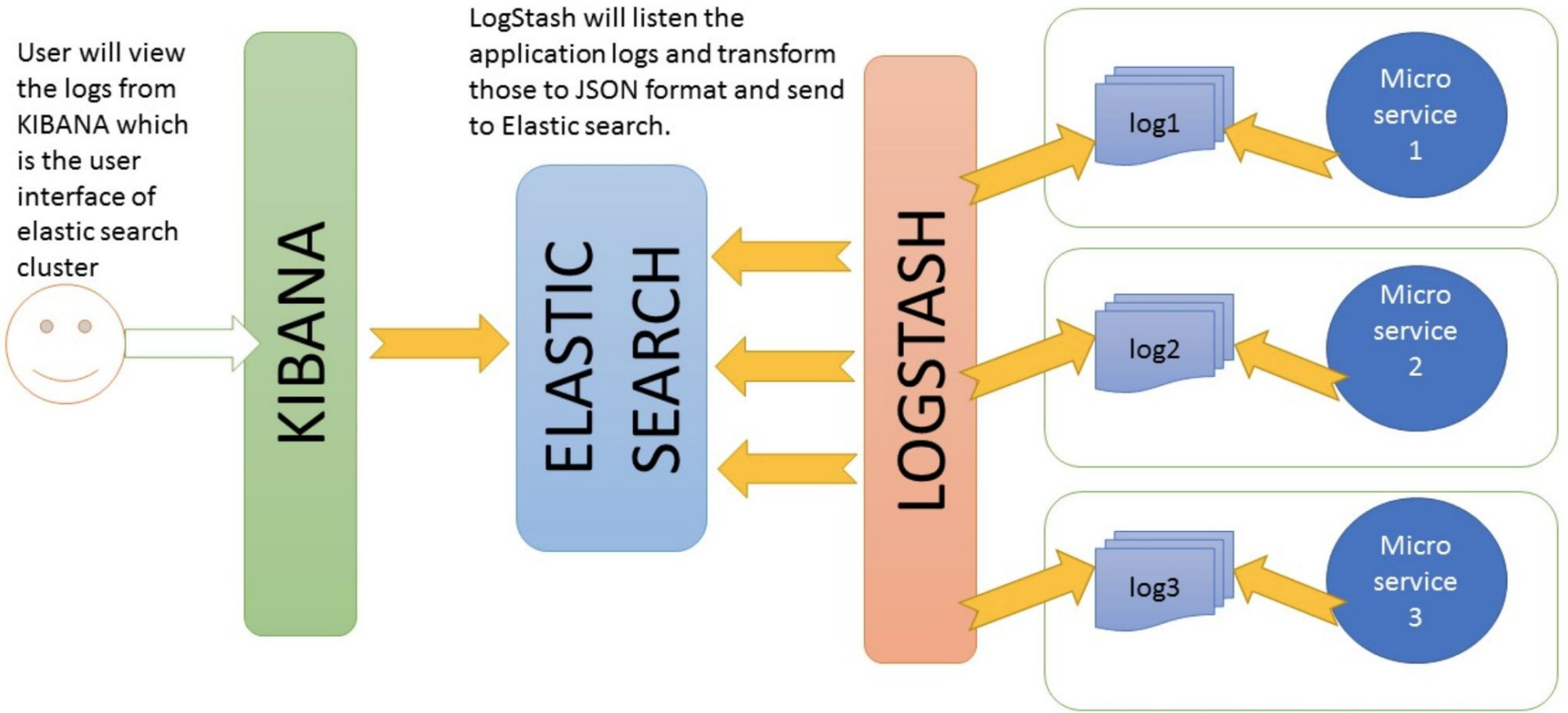
Microservice Pros

- Independently deployable
- Independently scalable
- Independent tech stack
- Faster development cycle
- Independence in case of failure - does not crash whole thing

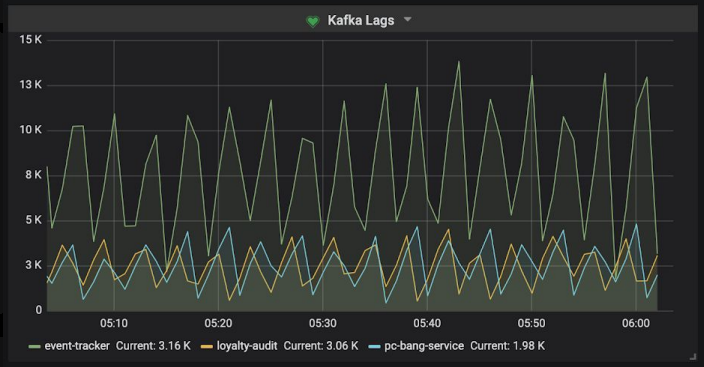
Microservice Cons

- Servers to manage
- Performance (process vs network)
- Transaction is hard (DB scattered)
- Network security
- Logging

how to overcome these cons?



ELK stack interaction with different applications based on Log file



Kafka disk space usage (%) — Last 15 minutes

Time	Usage (%)	Usage (%)	Usage (%)
2019-03-31 05:54	-	38.38	-
2019-03-31 05:53	40.20	-	39.39

Audit mongo disk space usage (%) — Last 15 minutes

Time	Usage (%)	Usage (%)	Usage (%)
2019-03-31 06:00	17.64	-	-
2019-03-31 05:59	-	-	-

Audit mongo disk space usage (%) — Last 15 minutes

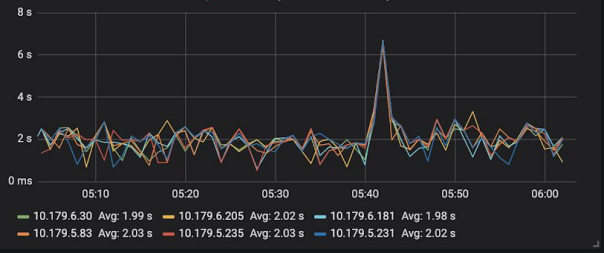
Time	Usage (%)	Usage (%)	Usage (%)
2019-03-31 06:02	-	-	1.47
2019-03-31 06:01	-	-	-

time tracker memory usage (%)

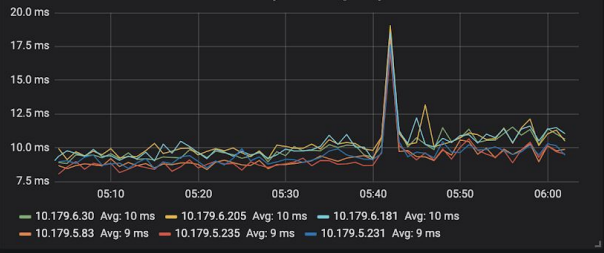
Last 15 minutes

Time	10.179.6.30_Average	10.179.6.181_Average	10.179.5.231_Average	10.179.5.235_Average	10.179.5.83_Average	10.179.6.205_Average
2019-03-31 05:56	13.44	-	-	-	-	13.50
2019-03-31 05:55	-	-	-	12.09	13.48	-

Kafka PlayerEvent Max Delay

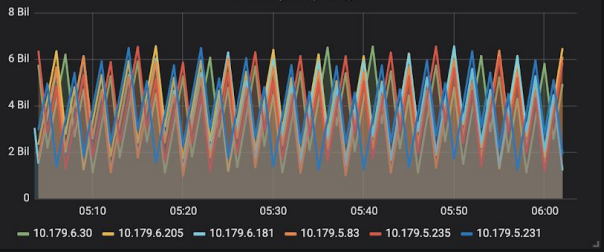


Kafka PlayerEvent Avg Delay

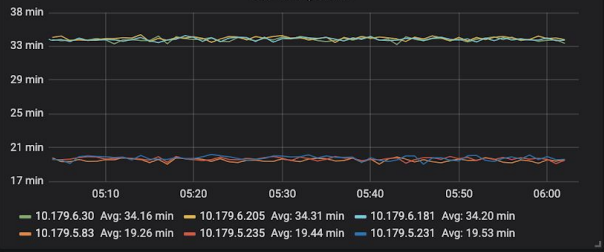


Kafka PlayerEvent Loop TPS

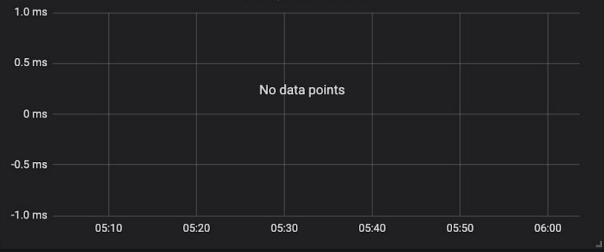
Memory Heap Usage



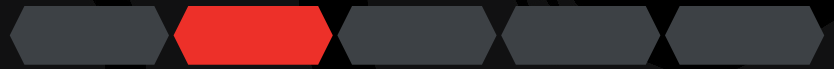
FillPcbangId Min



Event Duplicate Filter Max



Spring boot




How can I use
Spring boot?

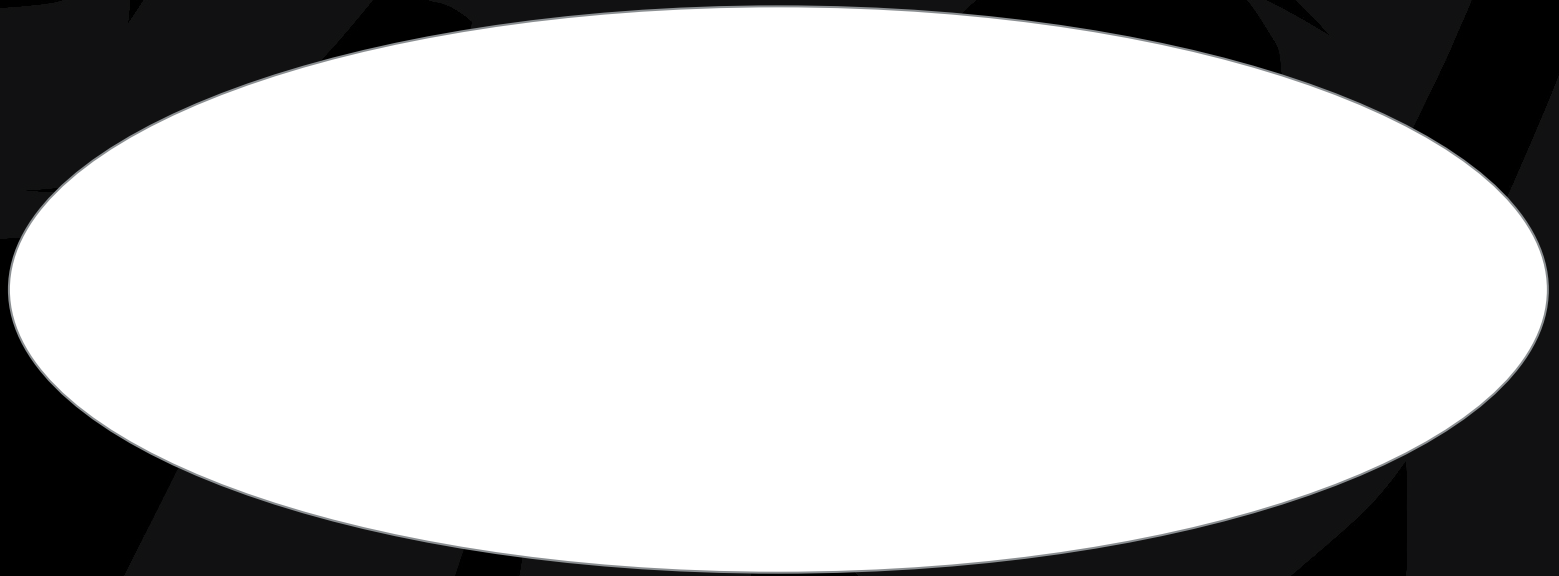
How can I use
Spring boot?

What is the Spring framework ?

- Framework ?
  - skeletons or environment for development
  - a collection of reusable, extendable libraries

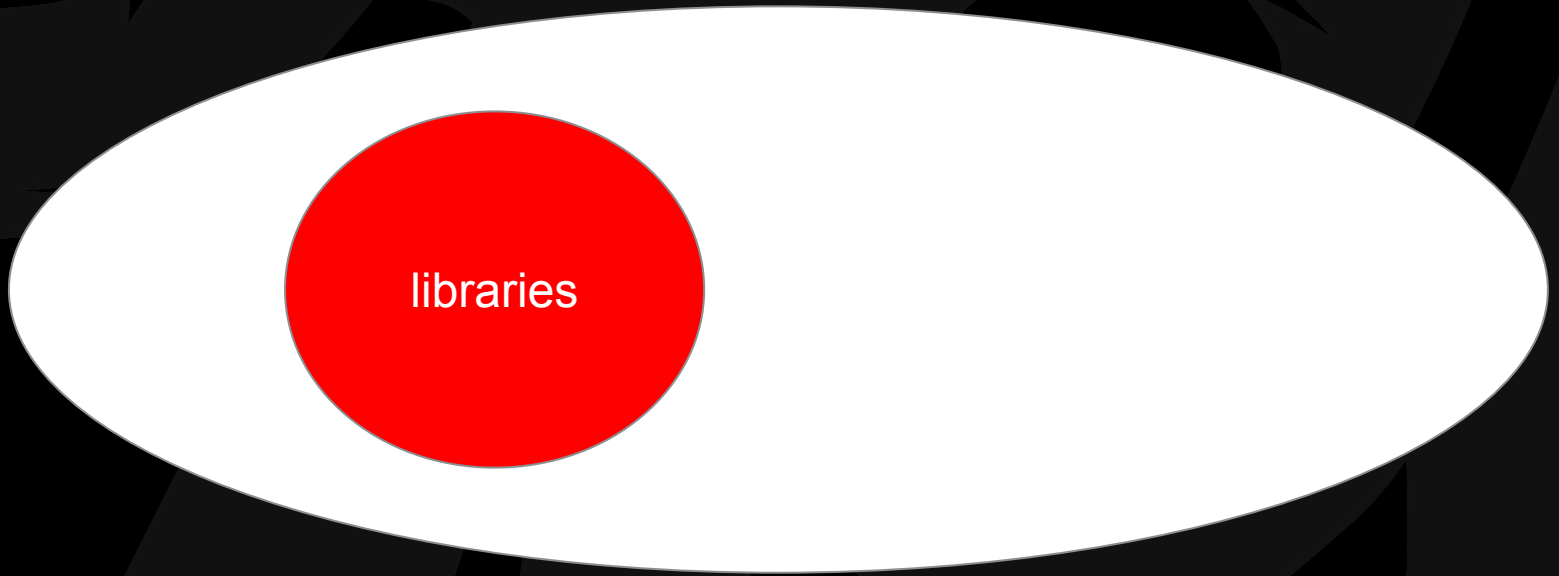
What is the Spring framework ?

- Framework



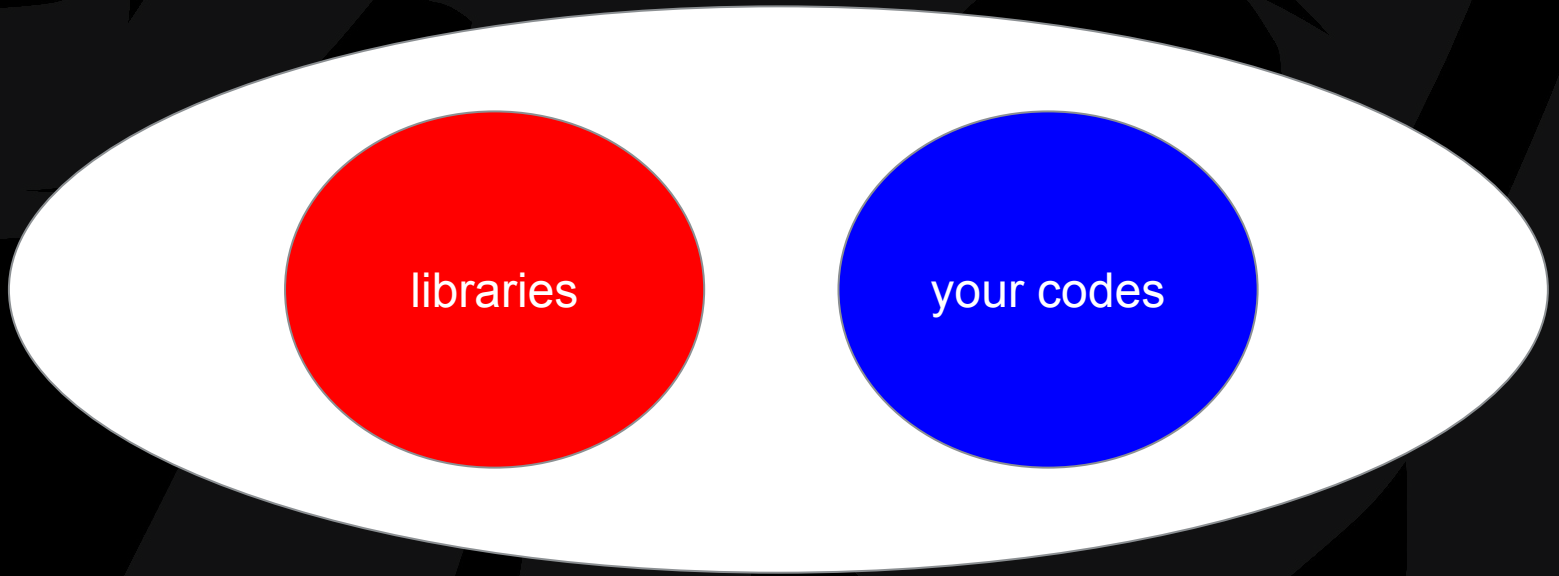
What is the Spring framework ?

- Framework



What is the Spring framework ?

- Framework



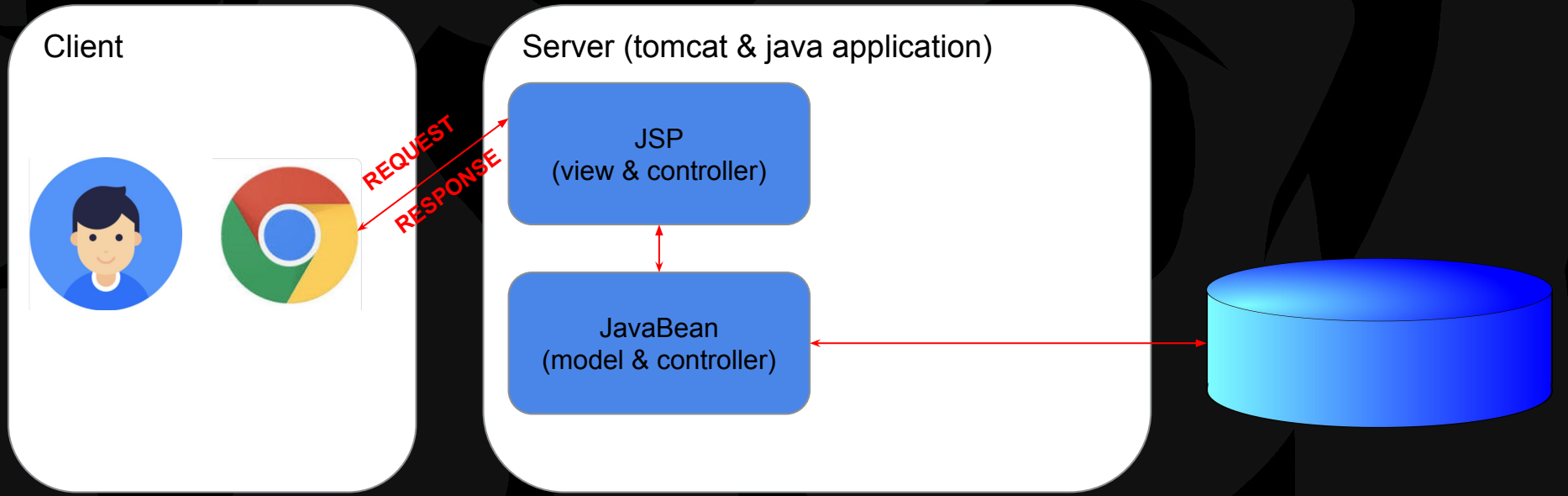
What is the Spring framework ?

- Spring framework ?
  - open source application for java platform
  - based on maven OR gradle
  - IoC (Inversion of Control)
  - DI (Dependency Injection)
  - web server
  - a lot of annotations (@Bean @Service @Repository … etc)
  - suitable to realize microservice architecture

standard framework for web service (Korea government)

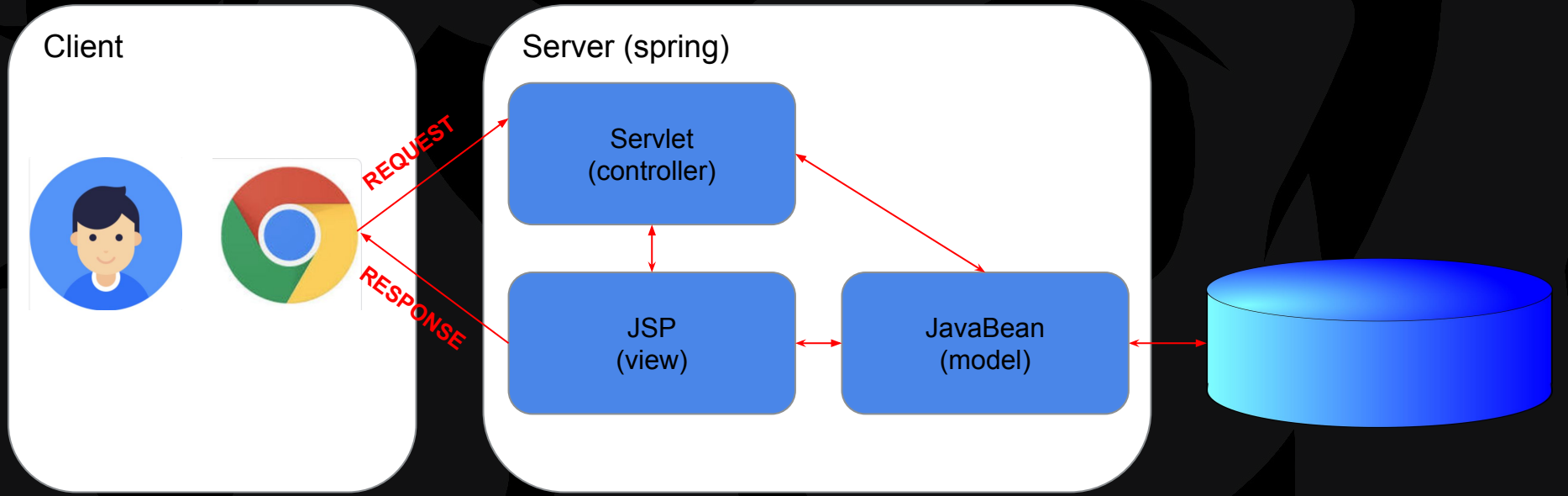
Spring MVC (Model, View, Controller)

- MVC - model 1



Spring MVC (Model, View, Controller)

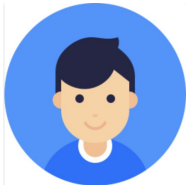
- MVC - model 2

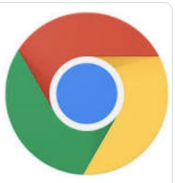


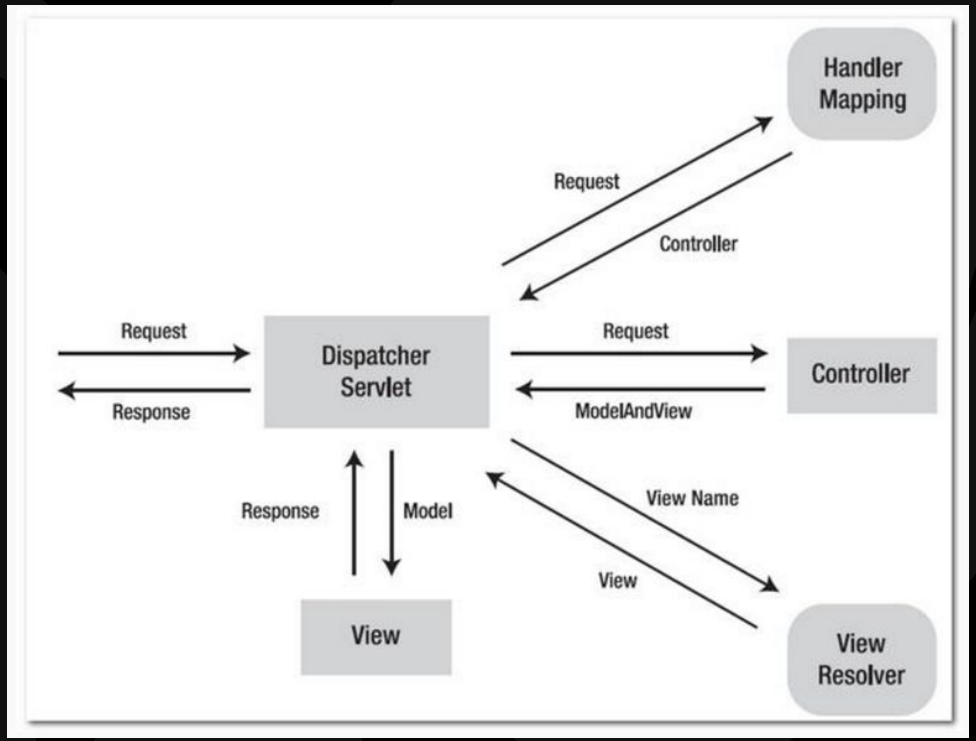
Spring MVC (Model, View, Controller)

- MVC - model 2

Client

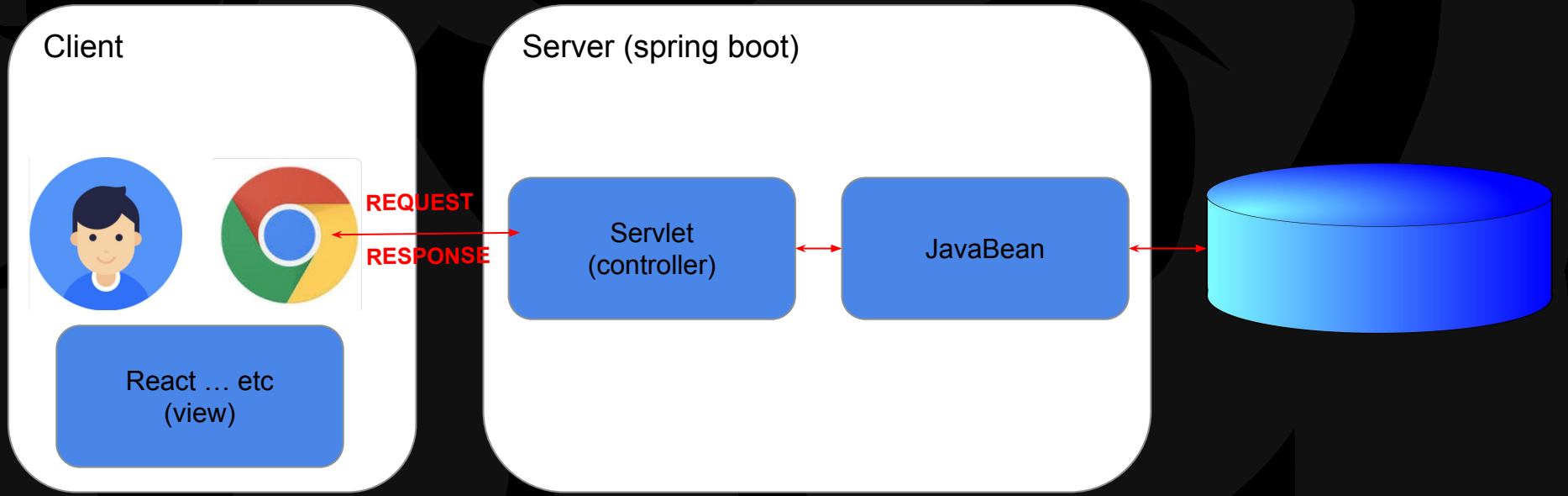






Now days ?

- Microservices



Spring vs Spring boot

Spring

- 앞서 보았던 서버의 역할들이 동작할 수 있도록...
  - 수동으로 dependencies 를 추가해 주어야 함
  - 수동으로 웹 서버를 셋업해야 함..
  - 개발 환경에서 IDE도 웹서버랑 연동해야함..
  - 수동.. 수동.. 수동... 환경 설정 (초/중급 개발자들 뛰어들기 부담) (링크)

Spring boot

- 최소한의 설정만으로! 서버로 즉시 활용! 즉시 배포! 즉시 서비스! 가능
- 웹 서버 내장
- Spring 사용한다 하면 자주 사용되는 라이브러리들!

실습 - let's create spring boot application

- Google search - spring boot initializr

- group : org.cnu.realcoding.weathercrawler
- artifact : weather-crawler

Check your project structures
Remove unnecessary maven configurations
Remove unnecessary resources
Rename application.properties to application.yml

Run your project
http://localhost:8080

잘 안되요…

https://github.com/example0312/weather-crawler

git clone
https://github.com/example0312/weather-crawler.git

git checkout c4e732

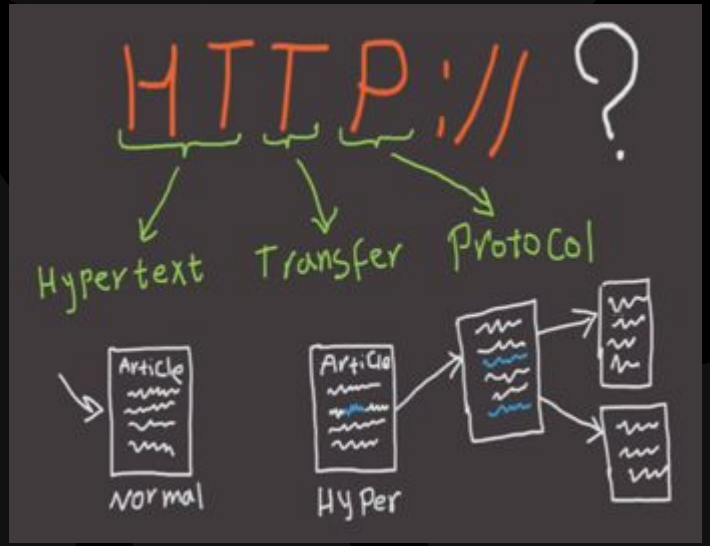
Structure of Spring boot application

- Maven project directory structure
- pom.xml & External Libraries
- spring configuration file (resources/application.yml)
- package naming (follow project’ group name)
- package convention (controller, service, repository, api, config)

REST API with spring

HTTP (HyperText Transfer Protocol)

Protocol to transfer hypertext between computers



HTTP Methods

- POST : Create (생성)
- GET : Select (조회)
- PUT : Update (변경)
- DELETE : Delete (삭제)

HTTP Methods

- POST : Create (생성)
- GET : Select (조회)
- PUT : Update (변경)
- DELETE : Delete (삭제)

- HEAD : Select (조회)
- PATCH : Update (변경)
- OPTIONS : proxy
- TRACE : for testing
- CONNECT : method 종류 확인

REST (Representational state transfer)

- Designed by : Roy Fielding
- HTTP spec 정의
- 개발자들이 HTTP를 이용한 Web Architecture를 제대로(RESTful하게) 활용하지 못하고 있구나…

~ _ ~ Confused examples ~ _ ~;;

- http://www.dogs.com/get/dog/bella
- http://www.dogs.com/dog/dogName=bella
- http://www.dogs.com/dogs/bella
- http://www.dogs.com/dog?dogName=bella
- http://www.dogs.com/dogs/names/bella

RESTful ?

- 웹의 장점을 최대로 활용하게 하는 아키텍쳐
  - 과거 다양한 방법(socket, SOAP …)을 사용하면서 client에서 server를 제어하는 기술이 발전

- 구성
  - 자원: URI에 자원이 표현되어야 한다
    - ex) http://localhost:8080/dogs
  - 행위: HTTP Method로 표현 가능하지 않은가? action을 URI에 포함하지 말아라

- 검색 & 공부 하면서 나의 API도 RESTful 하게 만듭시다..

REST basic (Resource, Method, Message)

이름이 lan 인 강아지를 생성하고 싶다!

REST basic (Resource, Method, Message)

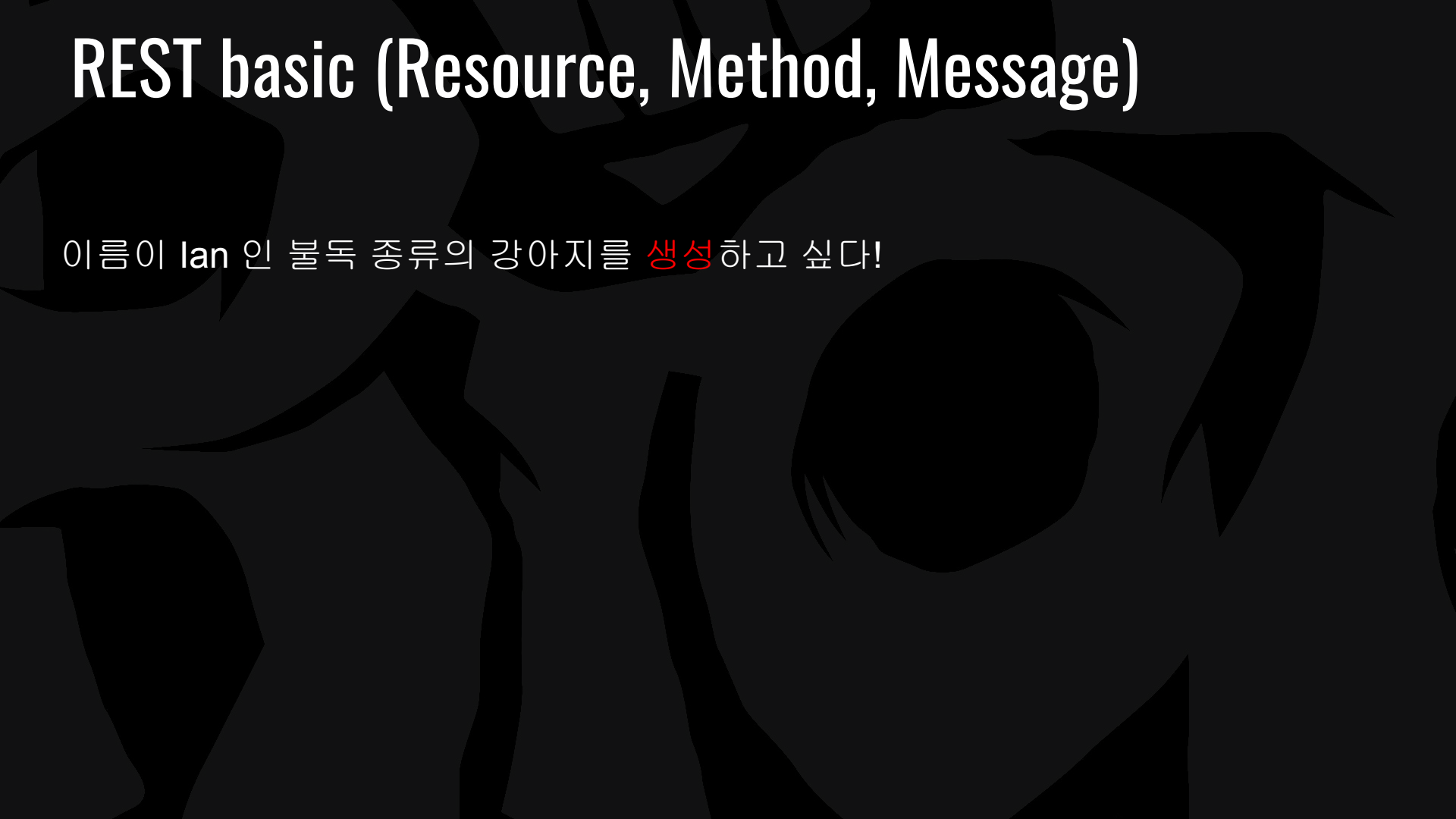
이름이 Ian 인 강아지를 생성하고 싶다!

- Resource : dogs
- Method : create (POST)
- Message : Request Body

HTTP POST방식으로, http://cnu.com/dogs

```
{
   “name”: “Ian”
}
```

REST basic (Resource, Method, Message)



이름이 **Ian** 인 불독 종류의 강아지를 생성하고 싶다!

REST basic (Resource, Method, Message)

이름이 Ian 인 불독 종류의 강아지를 생성하고 싶다!

- Resource : dogs
- Method : create (POST)
- Message : Request Body

HTTP POST방식으로, http://cnu.com/dogs

```
{
   “name”: “Ian”, “type”: “bulldog”
}
```

REST basic (Resource, Method, Message)

이름이 **Ian** 인 강아지를 조회하고 싶다!

REST basic (Resource, Method, Message)

이름이 Ian 인 강아지를 조회하고 싶다!

- Resource : dogs
- Method : selete (GET)
- Message : RequestParam or PathVariable

HTTP GET방식으로, http://cnu.com/dogs?name=david
HTTP GET방식으로, http://cnu.com/dogs/david

REST basic (Resource, Method, Message)

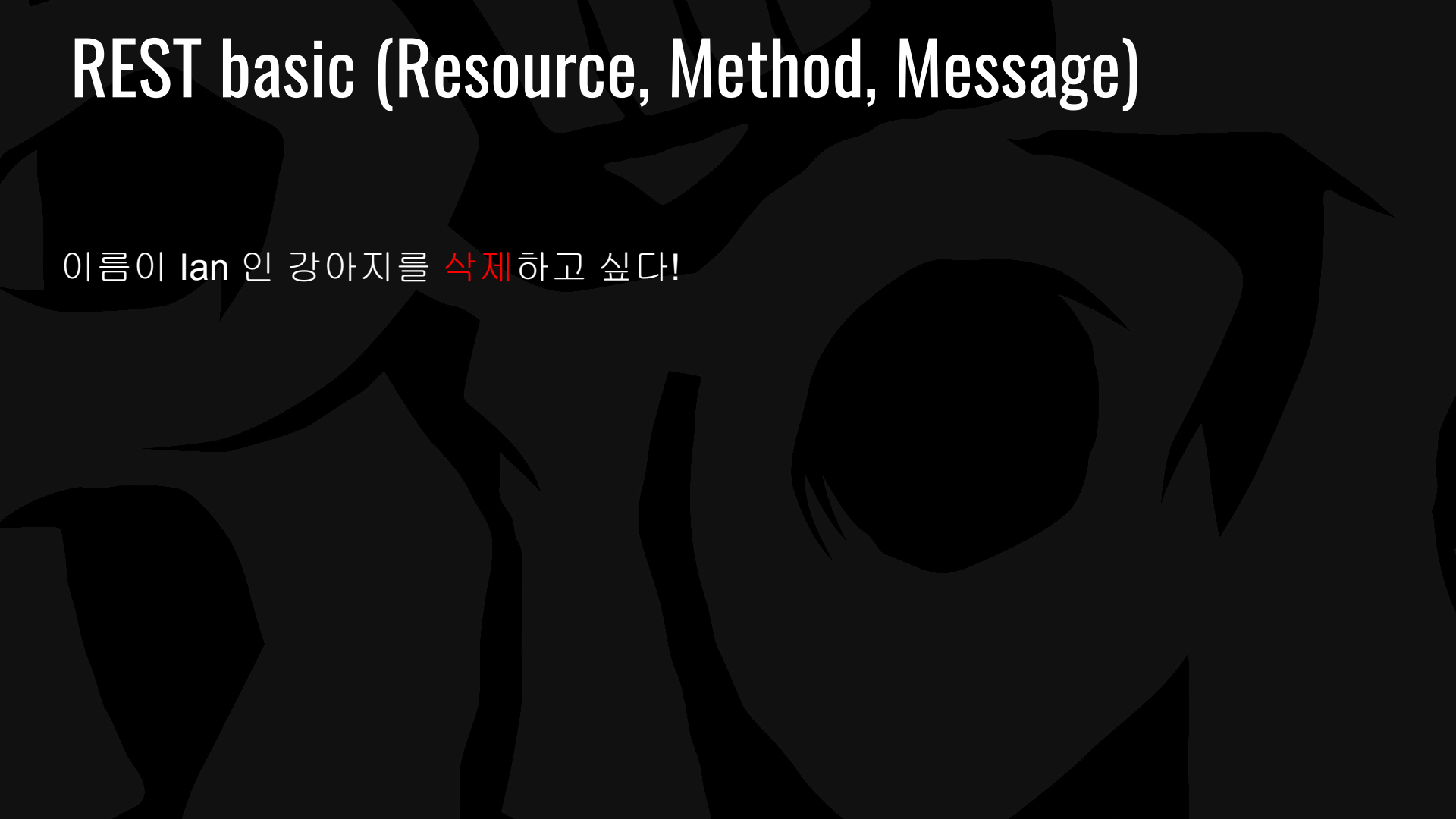
REST basic (Resource, Method, Message)

이름이 Ian 인 강아지의 종을 변경하고 싶다!

- Resource : dogs
- Method : update (PUT)
- Message : PathVariable and Request body

HTTP PUT방식으로, http://cnu.com/dogs/ian
{“type”: “martiz”}

REST basic (Resource, Method, Message)



이름이 Ian 인 강아지를 삭제하고 싶다!

REST basic (Resource, Method, Message)

이름이 Ian 인 강아지를 삭제하고 싶다!

- Resource : dogs
- Method : delete (DELETE)
- Message : PathVariable or RequestParam

HTTP DELETE방식으로, http://cnu.com/dogs/ian
HTTP DELETE방식으로, http://cnu.com/dogs?name=ian

REST basic (Resource, Method, Message)

Quiz)
Web Browser는 어떤 HTTP Method만을 사용?

IoC & DI

- Spring container (IoC, Inversion of Control)

spring framework 안에서 사용하는 자바 객체를 어떻게 관리?
- SpringContainer가 객체를 singleton패턴으로 관리하고, 필요한 순간에 사용 가능토록 주입(Dependency Injection)

IoC example
- @Configuration @Service @Repository @Controller @Bean … etc

DI example
@Autowired “spring container야, 내가 원하는 객체를 사용 가능하도록, 주입해 줘”

실습 - let's expose endpoints in your spring boot

- **Expose POST method for dogs**
- **Expose GET method for all dogs**
- **Expose GET method for one dog by name**
  (Use PathVariable, localhost:8080/dogs/ian)
- **Expose PUT method for one dog**
- **Expose DELETE method for one dog**
  (Use RequestParam, localhost:8080/dogs?name=ian)

잘 안되요…

https://github.com/example0312/weather-crawler

git clone https://github.com/example0312/weather-crawler.git

git checkout b37876

실습 - let's support Swagger in your spring boot

FE / BE 의 분리

FE engineer ←→ BE engineer
- 개발 환경에서 swagger를 활용하여 소통

나의 Webserver 에서
- 제공하는 api들을 명세
- 자유롭게 사용해 볼 수 있도록 UI 제공

http://localhost:8080

잘 안되요…

https://github.com/example0312/weather-crawler

git clone https://github.com/example0312/weather-crawler.git

git checkout f60cab

과제

- 강사를 조별 채널에 초대하기
- 조별로 조원들끼리 의견을 취합해서 오늘 강의에서 좋았던점 또는 나빴던점을 강사에게 피드백하기
- 조별로 조원들끼리 의견을 취합해서 오늘 강의에서 학습했던 주제들 중 하나를 선택하고,
  궁금한 점을 강사에게 질문해서 답변받기
- 마감 : 4월 5일 금요일 23시 59분 59초 까지

지난시간 피드백

- 강의의 목적은 대체……?

궁극적으로 이 수업이 목적이 무엇인지 잘 모르겠다!?

학생으로 학교에서 배우는 것, 하는 것

개발자로 회사에서 배우는 것, 하는 것

학생들이 강사들에게 수업듣는 이유?

강사들이 학생들에게 강의하는 이유?

지난시간 피드백

- 지난번 과제

여러분에게 필요한 것? (토론은 성장의 밑거름이 됩니다)

각 조별로 궁금했던점 / 강사가 답변한 내용 (강사 말이 전부 정답인가요?)

https://docs.google.com/document/d/1rhTamH5yph-GiJfmtaSqfi1I-K1CNxKG6DtH5_5SKtI/edit?usp=sharing

다시 짚고 넘어가자 #1

이름이 Ian 인 강아지를 조회하고 싶다?

- Resource : dogs
- Method : selete (GET)
- Message : RequestParam or PathVariable

RequestParam

<client request url>
http://cnu.com/dogs?name=david

<구현>

```
@GetMapping("/dogs")
public Dog getDogByNameByRequestParam(@RequestParam String name) {
    return dogService.getDogsByName(name);
}
```

PathVariable

<client request url>
http://cnu.com/dogs/david

<구현>

```
@GetMapping("/dogs/{name}")
public Dog getDogByName(@PathVariable String name) {
    return dogService.getDogsByName(name);
}
```

다시 짚고 넘어가자 #2

- IoC ? DI ? 도대체 뭔가요?

객체지향 언어는 class를 정의할 수 있고 runtime 때에는 객체를 생성함으로써 class 내부의 field들이나 method를 사용할 수 있게 됩니다.

그러나 우리가 실습했던 Spring boot application에서는 객체를 생성할 때 사용하는 예약어인 `new`를 사용한 적이 없습니다.
그 이유는 Spring은 IoC 개념이 적용된 Spring container를 기반으로 동작하기 때문입니다.

IoC는 inversion of control, 제어의 역전 이라는 의미입니다.
즉, 객체를 생성하고 관리하는 등의 제어 역할을 담당하는 주체가 framework를 사용하는 `우리`가 아닌 `Spring framework`에게 `위임`한다는 데서 등장한 용어 입니다.

Spring boot application이 시작될 때, Spring container도 함께 시작됩니다.
Spring container는 DL(Dependency Lookup, 의존성 검색) 기능을 통해 프로젝트 내부의 annotation을 기반으로 객체를 생성하고, container에 보관합니다.
또한, @Autowired annotation이 사용된 곳에는 DI(Dependency Injection, 의존성 주입) 기능을 통해 container에 보관된 객체를 사용할 수 있도록 referencing 합니다.

Spring framework에서 `우리`는 적절한 annotation들을 사용하여 Spring에게 객체를 생성하고 관리하는 모든 일을 위임합니다.

spring-context-5.1.5.RELEASE
org.springframework.context.support.AbstractApplicationContext

Swagger?

- 웹 서버의 사용가능한 endpoint(API) 들을 사용하기 편리하게 Browser 상에서 제공

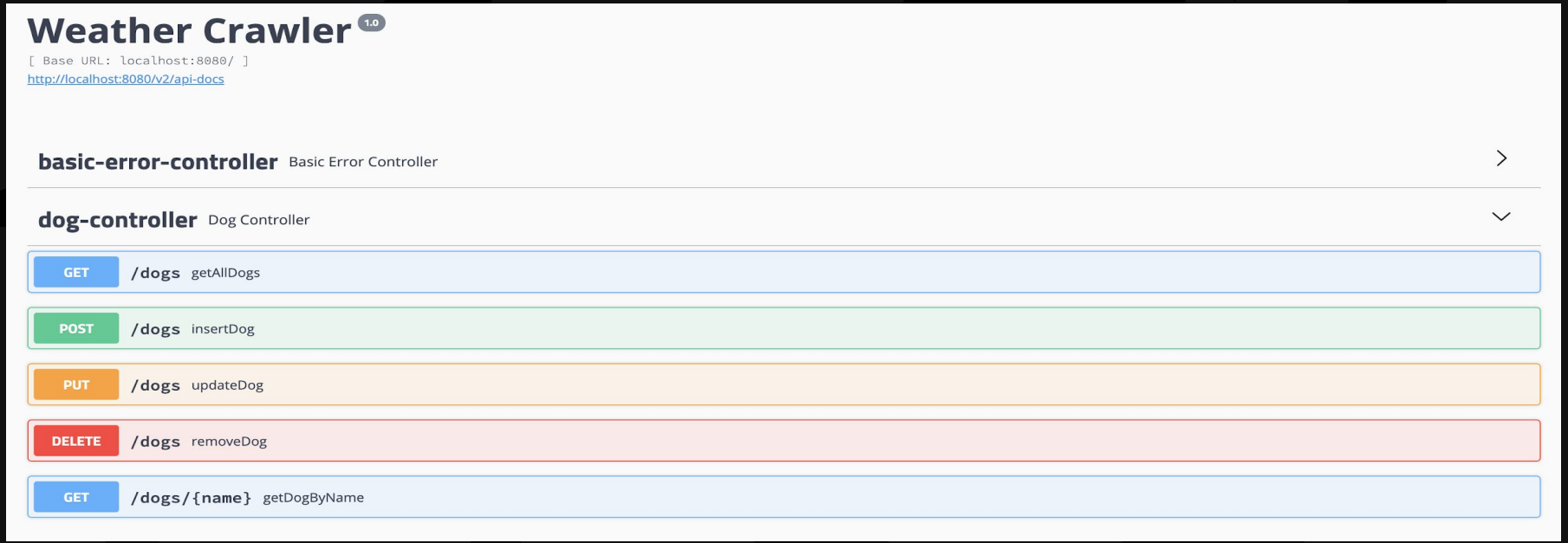
Spring boot application에서는 아래 2가지만 적용하면 http://localhost:8080 으로 즉시 사용 가능!

1. Swagger 관련 dependency 2개 추가 (pom.xml)
   https://github.com/example0312/weather-crawler/commit/00b4666f4be2e1568678cad0aebf40816c481554

2. Swagger 설정 파일을 추가 (config 패키지에 SwaggerConfig 클래스 추가)
   https://github.com/example0312/weather-crawler/commit/f60cab9de1abd7b8b2f0ce1a4c2177152f40cb96

Swagger?

- http://localhost:8080 브라우저에서 접속 시 우리 웹서버에서 제공하는 endpoints(API)들이 나열되고, 사용도 가능하다! (POSTMAN byebye)



개발하기 전 워밍업 (개발 전 미팅)

- 우리는 유저에게 한국 다양한 도시의 현재 날씨 정보를 열람할 수 있는 안드로이드 앱을 만들고자 한다.

FE를 담당하시기로 한 Android app 엔지니어 도창욱님과 날씨 앱을 만들기 위해 회의에 참석했다. 우리는 BE 엔지니어이다.

도창욱: “BE 는 뭐 기반으로 만드실건가요? 날씨 정보는 어디서 가져오실거예요?”

우리: “Spring boot 으로 하려고 하고요, 날씨정보는 OpenWeatherMap 이라는 곳에서 현재 날씨에 대한 OpenAPI 를 제공하더라고요. 거기서 가져오려고 하긴 하는데... 다른데 써도 상관 없으니 좀 더 살펴보려고요. FE에서는 BE로부터 어떤 request들을 하셔야 하나요?”

도창욱: “음.. 일단 도시 이름으로 현재 날씨를 받아오는 endpoint(API)가 있었으면 좋겠고요. 어떤 도시 이름을 쓸 수 있는지 저는 모르니까… 사용 가능한 도시 이름들을 리스트로 제공하는 endpoint도 있었으면 좋겠네요”

우리: “네~ OpenWeatherMap의 API들 중에 도시 이름으로 현재 날씨 가져올 수 있는지는 한번 봐야겠네요. 만약 가능하다면, 사용 가능한 도시 이름들 리스트는 OpenWeatherMap에서 API 제공할 것 같기는 한데, 이건 아직 모르겠어요. 찾아보고 공유 드릴게요~”

도창욱: “네네"

개발하기 전 워밍업

- FE가 BE에게 바라는 요구사항

1. 사용 가능한 도시 이름들을 요청하면, 사용가능한 도시이름의 리스트를 반환하는 기능
2. 도시 이름으로 현재 날씨 정보를 요청하면, 해당 도시의 현재 날씨 정보를 반환하는 기능

- OpenWeatherMap 을 통해 요구사항 다 충족 시킬수 있어?

1. 살펴보자. 찾자. 써보자(PostMan~).
2. 사용가능한 도시정보 어디있지? https://openweathermap.org/current 음? ISO 3166 country code 사용하라고? 도시 이름은 어디있니…?
3. 잘 찾아보니 있음 https://openweathermap.org/current Json format 으로 된 파일 다운로드해서 보라는 식이네.. 도시 이름만 리스트로 주는 API는 없는건가?
4. 아무리 찾아봐도 없음. 생각해보니 없는게 맞음. 세계 전체를 대상으로 하는 날씨 사이트다 보니.. 20만건이 넘는 도시 이름들을 API를 통해 리스트로 주다가는… 응답이… 엄청 늦을것 같기도 하다...
5. 어쩔 수 없이 3번의 파일에서, country code “KR” 인 녀석들만 뽑아서 써야겠군.

Q) “KR” 인 녀석들의 도시 이름을 잘 뽑았다 치자. (실제로 뽑았더니 240여건..)
이를 코드에 박아넣고 쓸까?
프로젝트 루트 디렉토리에 파일로 박아넣고 쓸까?
웹 서버를 하나 뚝딱 파서 API로 제공하게 할까?

A) http://demo6468405.mockable.io/weather-crawlers/cities

개발하기 전 워밍업 (정리)

- 요구사항을 충족시킬 수 있음을 확인했다.

사용가능한 도시
이름 request

사용가능한 도시
이름 response

도시의 현재 날씨
request

도시의 현재 날씨
response

Spring Boot

사용가능한 도시
이름 request

사용가능한 도시
이름 response

도시의 현재 날씨
request

도시의 현재 날씨
response

OpenWeatherMap

본격 개발 #1 (도시 리스트 제공하기)

- 시나리오
  1. FE 에서 도시 리스트 달라고 우리에게 request 날림
  2. 우리 웹서버에서 아까 임시로 만들었다던 웹 서버로 request 날림
  3. 리턴받은 도시 리스트를 FE에 리턴하기

본격 개발 #1 (도시 리스트 제공하기)

- 다른 웹 서버랑 통신할 클래스를 구현하자 (AvailableCityNamesApiClient)

1. api 패키지에 AvailableCityNamesApiClient 클래스 정의
2. @Service 어노테이션 추가
3. RestTemplate 필드 선언
4. RestTemplate 필드 변수에 @Autowired 어노테이션 추가
5. config 패키지에 HttpConfig 클래스 정의
6. @Configuration 어노테이션 추가
7. RestTemplate 반환하는 메소드 구현
8. @Bean 어노테이션을 메소드에 추가
9. public List<String> getAvailableCityNames() 정의
10. RestTemplate을 사용하여 웹 서버에 http request를 날리자. getAvilableCityNames() 을 구현
    http://demo6468405.mockable.io/weather-crawlers/cities
11. 응답받은 Response 의 Body 를 꺼내어 리턴

잘 안되요...

git clone
https://github.com/example0312/weather-crawler.git

이미 clone 받았어요? 그럼..
git stash
git checkout master
git pull

==========

git checkout add-weather-service
git checkout f188c1

본격 개발 #1 (도시 리스트 제공하기)

- 나의 웹 서버에서 내부적으로 필요한 비즈니스 로직을 처리하기 위한 서비스 클래스를 정의하자 (WeatherService)

1. service 패키지에 WeatherService 클래스 정의
2. @Service 어노테이션 추가
3. AvailableCityNamesApiClient 필드 선언
4. AvailableCityNamesApiClient 필드에 @Autowired 어노테이션 추가
5. public List<String> getAvailableCityNames() 정의
6. AvailableCityNamesApiClient 필드를 사용하여 getAvilableCityNames() 호출
7. 호출해서 받은 리턴 결과를 다시 리턴

잘 안되요...

git clone
https://github.com/example0312/weather-crawler.git

이미 clone 받았어요? 그럼..
git stash
git checkout master
git pull

==========

git checkout add-weather-service
git checkout 2e1595

본격 개발 #1 (도시 리스트 제공하기)

- FE에서 나의 웹 서버로 도시 이름들 리스트 달라고 요청할 endpoint(API)를 생성하자

1. controller 패키지에 WeatherController 클래스 정의
2. @RestController 어노테이션 추가
3. @RequestMapping 어노테이션 추가
4. WeatherService 필드 선언
5. WeatherService 필드에 @Autowired 어노테이션 추가
6. @GetMapping 어노테이션 추가하고, 메소드 구현.
7. 메소드는 WeatherService의 getAvilableCityNames() 를 호출하고, 리턴받은 결과를 다시 리턴

잘 안되요...

git clone
https://github.com/example0312/weather-crawler.git

이미 clone 받았어요? 그럼..
git stash
git checkout master
git pull

==========

git checkout add-weather-service
git checkout 4b8584e

본격 개발 #1 (점검해 보세요)

- http://localhost:8080/weather-crawler/available-cities

사용가능한 도시
이름 request

사용가능한 도시
이름 response

도시의 현재 날씨
request

도시의 현재 날씨
response

Spring Boot

사용가능한 도시
이름 request

사용가능한 도시
이름 response

도시의 현재 날씨
request

도시의 현재 날씨
response

OpenWeatherMap

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- 시나리오
  1. FE 에서 특정 도시의 현재 날씨 정보를 달라고 우리에게 request 날림
  2. 우리 웹서버에서 OpenWeatherMap 웹 서버로 request 날림
  3. 리턴받은 도시 리스트를 FE에 리턴하기

- 앗 그런데!?

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- 시나리오
  1. FE 에서 특정 도시의 현재 날씨 정보를 달라고 우리에게 request 날림
  2. 우리 웹서버에서 OpenWeatherMap 웹 서버로 request 날림
  3. 리턴받은 도시 리스트를 FE에 리턴하기

- 앗 그런데!?

https://openweathermap.org/guide#plans (돈 안내는 사람은 분당 60회로 제한함)

Q) FE(안드로이드 날씨 앱)를 사용하는 사용자가 다수일 경우?
Q) FE(안드로이드 날씨 앱)를 사용하는 사용자가 화가나서 조회 버튼을 광클릭했을 경우?

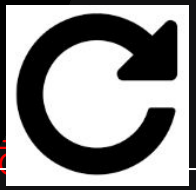
본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- 생각을 다시 정리하자

우리 웹 서버는 현재 날씨를 조회할 수 있는 도시 이름들의 리스트를 가져올 수 있다.

우리 웹 서버가 시작되면 일단

도시 이름들의 리스트를 가져오자. (한 2분에 한번씩 ~~



이 리스트를 순차적으로 탐색하면서 OpenWeatherApi 에 request를 날려 현재 날씨 정보를 가져온 후, 이를 DB에 저장하자.

FE에서 특정 도시의 날씨 정보 달라고 하면, DB에서 해당 도시의 최근의 데이터를 꺼내주자.

앗 근데?

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- 반복할 부분을 좀 손봐야 겠다... 이렇게 하면 어떨까?

가져온 도시 이름들 리스트를 Queue에 넣기

1.5초? 2초? 마다 한번씩 Queue에서 도시 이름을 꺼내서 request를 날린 후 DB에 저장
(저장했으면 도시 이름은 다시 Queue에 넣어야겠죠?)

Q) 분당 60회 안전?

A) 분당 30~45회니까 안전..... 대신...

사용가능한 도시
이름 request

사용가능한 도시
이름 response

도시의 현재 날씨
request

도시의 현재 날씨
response

Spring Boot

사용가능한 도시
이름 request

사용가능한 도시
이름 response

도시의 현재 날씨
request

도시의 현재 날씨
response

OpenWeatherMap

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

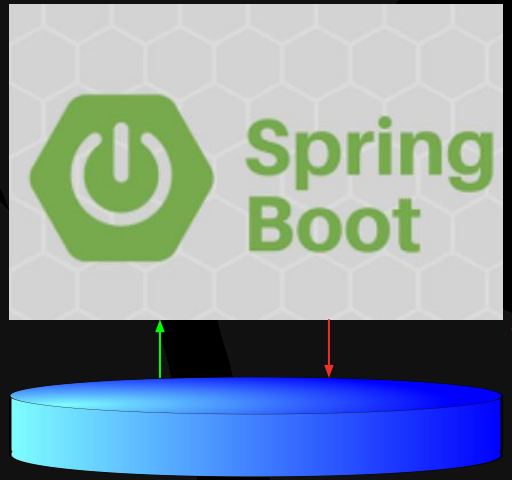
변경된 부분

사용가능한 도시 이름 request

사용가능한 도시 이름 response

도시의 현재 날씨 request

도시의 현재 날씨 response



사용가능한 도시 이름 request

사용가능한 도시 이름 response

도시의 현재 날씨 request

도시의 현재 날씨 response

OpenWeatherMap

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- OpenWeatherMap 웹 서버랑 통신할 클래스를 구현하자 (OpenWeatherMapApiClient)

1. api 패키지에 OpenWeatherMapApiClient 클래스 정의, @Service 어노테이션 추가
2. RestTemplate 필드 선언 @Autowired 어노테이션 추가
3. ApiKey로 사용할 String 필드를 하나 선언 (자신의 Key값을 assign https://home.openweathermap.org/api_keys), 호출할 URI 필드 선언
4. public String getCurrentWeather(String cityName) 정의
5. RestTemplate을 사용하여 OpenWeatherMapApiClient 웹 서버에 http request를 날리자. getCurrentWeather(String cityName) 을 구현
6. 응답받은 Response 의 Body 를 꺼내어 리턴
7. 응답받은 Body String을 그냥 돌려준다..? 담을 그릇을 하나 만드는게 좋지 않을까?
8. domain 패키지에 CurrentWeather 클래스 정의
9. CurrentWeather 클래스에 적절한 field들을 선언 (Response body의 JSON 형태를 보면서.. https://openweathermap.org/current)
10. OpenWeatherMapApiClient로 돌아가서 public String getCurrentWeather(String cityName) 의 리턴타입을 CurrentWeather 클래스로 변경

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- 방금 전까지 잘 못따라갔어요..

https://github.com/example0312/weather-crawler 참고하시거나,

아직 클론을 안받았다면 적절한 폴더에 클론을 받기
git clone https://github.com/example0312/weather-crawler.git

이미 클론 받았다면,
git stash
git checkout master
git pull

그런다음.....

git checkout add-weather-service
git checkout 04e6e41

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- 개발 #1에서 정의했던 WeatherService에 반복작업을 위한 메소드를 정의하자

1. OpenWeatherMapApiClient 필드 선언, @Autowired 어노테이션 추가
2. AvailableCitiesApiClient 필드 선언, @Autowired 어노테이션 추가 (이미 개발 #1에서 추가 되어 있을 거예요)
3. CityNamesQueue 필드 선언, Queue 할당

4fcb2fd

4. public void getCurrentWeatherPeriodicallyByCityName() 정의, 구현
5. queue 가 비었으면? AvailableCitiesApiClient 사용해서 도시 이름들 가져오고 큐에 넣기
6. queue 에서 도시 이름을 꺼내고 다시 넣기, OpenWeatherMapApiClient 사용해서 현재 날씨정보 가져오기
7. 가져온 현재 날씨 정보를 DB에 저장하기 위해, DB작업을 담당할 CurrentWeatherRepository 필드 선언

50e06a9

8. repository 패키지에 CurrentWeatherRepository 클래스를 생성하고 @Repository 어노테이션 추가
9. MongoTemplate 필드를 선언, @Autowired 어노테이션 추가, MongoTemplate이 구현된 maven dependency 추가

618053e

10. config 패키지에 MongoConfig 클래스 선언, @Configuration 어노테이션 추가
11. MongoTemplate 반환하는 메소드 구현, @Bean 어노테이션을 메소드에 추가

410ee21

12. MongoTemplate 필드를 사용하여 데이터를 DB에 삽입하는 insertCurrentWeather(CurrentWeather currentWeather) 구현

4b3d54a

13. 다시 WeatherService로 돌아와서, 가져온 현재 날씨 정보를 저장하기위해 CurrentWeatherRepository 에 구현한 메소드를 insert 메소드를 호출
14. insert된 객체를 로깅
15. getCurrentWeatherPeriodicallyByCityName 메소드에 반복 작업을 위한 @Scheduled 어노테이션 추가
16. 메인함수 존재하는 클래스에 @EnableScheduling 어노테이션 추가

d9f96db

본격 개발 #2 (점검해볼까요?)

- Spring boot application 시작.
- Exception 또는 Error 없이, DB에 성공적으로 2초마다 저장하는지 로그 확인
- Robo 3T 활용해서 저장된 데이터 확인
  - 마우스 더블클릭 -> localhost:27017 접속
  - weather-crawler database 선택
  - currentWeather collection 더블클릭
  - 데이터 확인

본격 개발 #2 (도시별 현재 날씨 정보 제공하기)

- FE가 도시 이름을 가지고 현재 날씨를 request 할 수 있도록, endpoint(API)를 제공하자

1. 개발 1단계에서 controller 패키지에 WeatherController 클래스가 어느정도 구현되어 있을 것이다.
2. @GetMapping 어노테이션 추가하고 PathVariable로.. 메소드 정의.
3. 메소드 구현. 메소드는 WeatherService에 아직 구현되지 않았지만, weatherService.getCurrentWeatherByCityName(cityName) 호출
4. Alt + Enter 사용해서 WeatherService에 아직 구현되지 않은 메소드 생성
5. 메소드 구현, CurrentWeatherRepository 사용해서, DB에서 현재 날씨를 가져오는 메소드를 호출 (findRecentCurrentWeatherByCityName(), 아직 없음)
6. Alt + Enter 사용해서 CurrentWeatherRepository 아직 구현되지 않은 메소드 생성..
7. 생성된 메소드를 구현. MongoTemplate 사용해서 insert 했듯이, find 하자.

잘 안되요…
https://github.com/example0312/weather-crawler

git clone
https://github.com/example0312/weather-crawler.git
or
git pull

git checkout add-weather-service
git checkout 3a48d20

본격 개발 #2 (점검해볼까요?)

- FE가 현재 날씨를 도시 이름으로 호출하는 API 를 점검하기
  (postman or browser 활용해서...)

본격 개발 #2 (점검해볼까요?)

- 간단 Mongo query 실습 (C R U D)

과제 (Riot Games API 호출하고 결과를 DB에 저장)

- https://developer.riotgames.com/ 방문
  로그인 (롤 계정이 없으면 이 기회에 하나 만드세요..)
- 자신만의 API 키를 생성하기
- 상단 메뉴바의 API DOCUMENTATION 으로 이동 (Swagger page 가 제공됨)

(optional) 롤을 좋아하시는 분이라면.. 좌측의 다양한 API들을 살펴보기..

- 좌측의 SUMMONER-V4 선택
- /lol/summoner/v4/summoners/by-name/{summonerName} 프로게이머의 소환사 이름(hide on bush)을 가지고 request 해보기
  팁: Query Param(RequestParam) 방식으로 API KEY를 INCLUDE 하세요!
  이때 Response Body에 존재하는 "id" 값이 encryptedSummonerId 입니다.

- 계속해서 좌측 LEAGUE-V4 선택
- 아까 응답받은 encryptedSummonerId 를 사용하여 /lol/league/v4/positions/by-summoner/{encryptedSummonerId} request 해보기

다 잘 되면?

과제 (Riot Games API 호출하고 결과를 DB에 저장)

과제 (5월 3일 23시 59분 59초 까지)

우리 Spring boot application에 "소환사이름"을 파라미터로 request 할 수 있는 GET방식의 endpoint(API)를 하나 구현하세요.

이 API는 Request를 받으면, 소환사의 league position(소환사의 게임 성적 정보)을 response합니다.

구현 방법

- SUMMONER-V4 의 /lol/summoner/v4/summoners/by-name/{summonerName} API 를 호출하여 encryptedSummonerId 를 얻으세요.
  encryptedSummonerId는 호출한 결과의 "id"에 들어있습니다.

- SUMMONER-V4 의 /lol/league/v4/positions/by-summoner/{encryptedSummonerId} API 를 호출하여 결과를 DB에 저장하고, 결과를 response 하세요.

가산점 (3개중 택일해서 구현)

- (0점) DB에 이미 소환사의 게임 성적 정보가 들어있던지 말던지 새로 얻은 결과를 삽입하세요

- (1점) DB에 이미 소환사의 게임 성적 정보가 들어있다면, 지워버리고 새로 얻은 결과를 삽입하세요
  힌트: mongoTemplate에 delete() 메소드가 있음

- (2점) DB에 이미 소환사의 게임 성적 정보가 들어있다면, 이를 업데이트하세요
  힌트: mongoTemplate에 update() 메소드가 있음

가산점

- (1점~2점) git 에 작업 과정을 남길 것 (중간 중간 커밋을 하셔야겠죠?)

과제 (Riot Games API 호출하고 결과를 DB에 저장)

주의 사항

- 새로 spring boot project 를 생성해서 과제를 수행할 것
- 과제 수행중 문제가 발생해서 해결이 어려우면 아무때라도 연락 @예재형(강사)
  참고) 과제이기때문에 질문이 들어오면 정답을 주지는 않음

제출 방법

- 과제가 완성되면 git url을 조별 채널에 제출