

## 개요

보험의 여러 서비스에 사용하기 위해 각 서비스 별로 KEY를 발급해주는 시스템을 개발해야 합니다.  
(이하 "KEY 시스템"이라 칭합니다.)

KEY를 발급 받기 위해서 최초에 "KEY시스템"에 관련 정보를 등록해야 합니다 (필수 API 1번)

이 후 GET API를 호출하면 1개의 KEY를 발급 받을 수 있습니다 (필수 API 2번),  
이 KEY는 DB TABLE의 PK로 이용 할 수 있는 NOT NULL, UNIQUE 특징을 가져야합니다.

Key는 문자형 또는 숫자형 둘 중 하나로 등록 할 수 있습니다.

### 문자형

- 대문자+숫자로 이루어진 4자리 문자열이 dash(-)로 4개 연결된 형태
- 길이는 고정이며, 더 이상 발급이 불가능 할 경우는 고려하지 않습니다
- 예) AA12-BB33-CCCC-00DD

### 숫자형

- 서비스 등록시 입력 받는 최소 자릿수(min-length)를 가지는 '정수'이며 99처럼 해당 자릿수에서 더 이상 발급이 불가능 할 경우 자릿수를 증가 시켜야 합니다.
- max-length는 고려하지 않으셔도 됩니다.
- 또한 시스템이 MySQL을 이용한다면 MySQL에 최적화된 생성 로직을 사용 할 수 있습니다.
  - MysqlKeyGenerator - MYSQL에 최적화된 KEY 생성 로직을 가짐
  - GenericKeyGenerator - 인프라에 독립적인 범용적인 KEY 생성 로직을 가짐

따라서 하나의 KEY는 3개의 조합으로 서비스에 등록되어 KEY를 발급 받을 수 있습니다.

- a) 문자형
- b) 숫자형 + MysqlKeyGenerator
- c) 숫자형 + GenericKeyGenerator

KEY 생성시 필요한 정보는 필수 API 1번에서 KEY를 등록할 때 같이 제공해야 합니다.

## 제약 사항 및 그 외

- 프로그래밍 언어는 **Java 또는 Kotlin**으로 진행해주세요.
- 그 외 프레임워크, 라이브러리, 빌드환경등은 전부 자유입니다.
- 설명이 부족하거나, 이해 안되는 부분이 있다면 새롭게 문제 정의를 내리거나 이해한 방향으로 진행 하셔도 됩니다.
- **README.md** 파일에 고민 거리, 구현의 어려움, 신경 쓴 부분 등을 작성해 주세요.

## 필수 구현 API

### 1. KEY 정보를 등록하는 API

예시1) “증권 번호”에 사용할 용도의 KEY 를 시스템에 등록하고 정상 응답을 받는다.

REQUEST

```
POST /api/key/register
{
  key: "policy-number"
  description : "보험 증서 번호에 사용할 KEY 값으로 테이블 PK 로 사용"
  type : number
  generator: "mysql"
  min-length : 10
}
```

RESPONSE : 200 OK

예시2) 고객센터에서 사용될 용도의 KEY를 등록하지만 이미 등록된 KEY 라서 등록 거부 됨

REQUEST

```
POST /api/key/register
{
  key: "claim-number"
  description: "고객센터에서 고객 문의사항이 접수될 때 사용하는 KEY"
  type: string
}
```

RESPONSE : 이미 등록된 키를 표현하는 에러 메시지와 함께 에러 반환 (자유롭게 구현)

### 2. 각 KEY 별로 새로운 KEY를 하나 발급 받는 API

예시1) 새로운 ‘증권 번호’를 위해서 숫자형의 KEY를 하나 발급 받는다.

- REQUEST
  - GET /api/key/policy-number
- RESPONSE
  - { value: "2755371233" }

예시2) 고객 문의사항이 접수되어서 Claim Table에서 사용 할 PK 용도로 KEY를 1개 받는다.

- REQUEST
  - GET /api/key/claim-number
- RESPONSE
  - { value:"UCAA-E22A-OKP-0021" }

## 우대 사항

### 전부 구현 하실 필요 없습니다

많은 주제를 조금씩 구현 하기 보다는 1-2개를 선정하여 다양한 고민을 하고, 표현해 주시는 것을 선호합니다.

( 목록에 없는 내용도 기능을 정의하고 구현 해 주셔도 됩니다. )

### 1. KEY 시스템을 위한 `spring-boot-starter` 를 추가로 구현해주세요

- 스프링 부트를 이해& 활용 하는 능력이 요구 되므로 해당 부분을 확인 할 예정입니다.
- git repository 또는 maven/gradle module 구성은 자유롭게 하시면 됩니다.

### 2. KEY 시스템이 scale-out 가능하도록 설계해주세요.

- 시스템 설계에 관한 능력이 요구됩니다.
- 추후에 KEY 발급 트래픽이 증가하더라도 버틸수 있는 확장 가능한 서비스를 구축하고 싶습니다.

### 3. KEY 시스템의 서비스 상황, 메트릭을 측정할 수 있는 방법을 추가해주세요.

- 메트릭, 모니터링에 대한 고민과 해결 능력을 보고 싶습니다.
- 각 KEY의 발급 속도, 최대 발급 속도 등을 확인 하고 싶습니다 (dashboard 는 생략)
- 호출 하는 IP, 서비스 등의 조건들로도 Metrics을 디테일하게 확인 하고 싶습니다.

### 4. 풍부하고 의미있는 테스트 커버리지

- 유닛테스트, 인테그레이션 테스트 작성 능력과, 의미있는 테스트에 대한 생각을 보고 싶습니다.
- 100% 커버리지를 목표로 하실 필요는 없습니다.

## FAQ

- 숫자형 KEY 에 대해서 좀 더 설명해 주세요.
  - Unique 한 KEY 가 발급되는 것을 기본으로 해주시고 min-length(최소 길이) 만 준수해주시면 됩니다.
  - 예를 들어 number type min-length:2 이라고 등록 했을 때 발급되는 KEY 는 10 ~ 99 이며, 더 이상 발급 할 수 없을 경우 자릿수가 증가하여 100 ~ 999 가 발급되면 됩니다. (max-length 에 대한 고민은 옵션입니다)
- GenericKeyGenerator 란 MysqlKeyGenerator의 차이점이 무엇인가요?
  - MysqlKeyGenerator 는 Mysql 에서 사용할 수 있는 쿼리나, 디비 엔진의 특성까지 이용하여 KEY 발급 속도를 증가시킬 수 있습니다.  
그에 반해 GenericKeyGenerator 는 범용 인프라(Redis, Kafka, ColumnDB, InMemory) 환경에서 범용적인 KEY 생성 로직을 가지고 있어야 합니다.
- GenericKeyGenerator 대신 Oracle 전용으로 만들어도 되나요?
  - 아니요. GenericKeyGenerator 는 꼭 범용적으로 구현 해 주세요.