

차세대 챗봇 Lab I

# Lecture-5 RESTful and JPA 연동

Sungja Choi,  
School of Computing, Gachon University  
2023-1



# 학습내용

RESTful 지원을 위한 JPA,에 대해 학습합니다.

Rest Repositories

Lombok

Data JPA

MySQL 연동

# Lecture-5. RESTful and API 연동

Rest Repositories 와 JPA의 기술을 익히고, MySQL과 연동하여 RESTful 서비스를 제공합니다.

# Practice

## >> ORM

(Object Relational Mapping)  
객체와 데이터베이스의  
관계를 자동으로  
매핑해주는 기술

## >> JPA

(Java Persistence API)  
ORM 기술표준의  
인터페이스 모음  
관계형 데이터베이스 사용  
방식을 정의  
Hibernate, OpenJPA등의  
인터페이스 제공  
CRUD SQL 처리

# 실습-1 아이템생성 및 확인(데이터베이스연동)

ORM 기술 표준 인터페이스를 모아 놓은 JPA를 사용하여 RESTful 서비스를 쉽게 구현합니다.

[실행화면]

http://localhost:8080/api/item/3

GET http://localhost:8080/api/item/3

```
{
  "name": "gcu-smartbrain",
  "price": 5000,
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/item/3"
    },
    "item": {
      "href": "http://localhost:8080/api/item/3"
    }
  }
}
```

mysql> select \* from item;

id	name	price
1	gcu-smartphone	2000
2	gcu-notebook	3000
3	gcu-smartbrain	5000

# Practice

>> start.spring.io

>> Lombok

자바 클래스의  
Setter/Getter 등의 메소드  
자동 생성지원

>> Rest Repositories

VC(View and Controller)를  
생략하고, 오직 Domain과  
Repository만을 가지고  
아주 쉽게 REST API  
서버를 구현하도록 지원

>> Spring Data JAP

CRUD 를 데이터베이스와  
자동 매핑하도록 지원

# 프로젝트 생성

The screenshot shows the Spring Initializr web application interface. The browser tab is titled "Spring Initializr" and the address bar shows "start.spring.io". The interface is divided into several sections:

- Project:** Includes radio buttons for "Gradle - Groovy" (selected), "Gradle - Kotlin", and "Maven".
- Language:** Includes radio buttons for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot:** Includes radio buttons for "3.1.0 (SNAPSHOT)", "3.1.0 (M2)", "3.0.6 (SNAPSHOT)", and "3.0.5" (selected). There is also a "2.7.11 (SNAPSHOT)" and "2.7.10" option.
- Project Metadata:** Includes input fields for "Group" (gcu.backend), "Artifact" (gcurestdata), "Name" (gcurestdata), "Description" (Demo project for Spring Boot), and "Package name" (gcu.backend.gcurestdata). There are also radio buttons for "Packaging" (Jar selected, War) and "Java" version (20, 17 selected, 11, 8).
- Dependencies:** A list of dependencies with checkboxes and category tags: "Lombok" (DEVELOPER TOOLS), "Spring Web" (WEB), "Rest Repositories" (WEB), "Spring HATEOAS" (WEB), "Spring Data JPA" (SQL), "MySQL Driver" (SQL), and "Spring Boot DevTools" (DEVELOPER TOOLS). Each dependency has a brief description.

At the bottom, there are three buttons: "GENERATE CTRL + G", "EXPLORE CTRL + SPACE", and "SHARE...".

# Practice

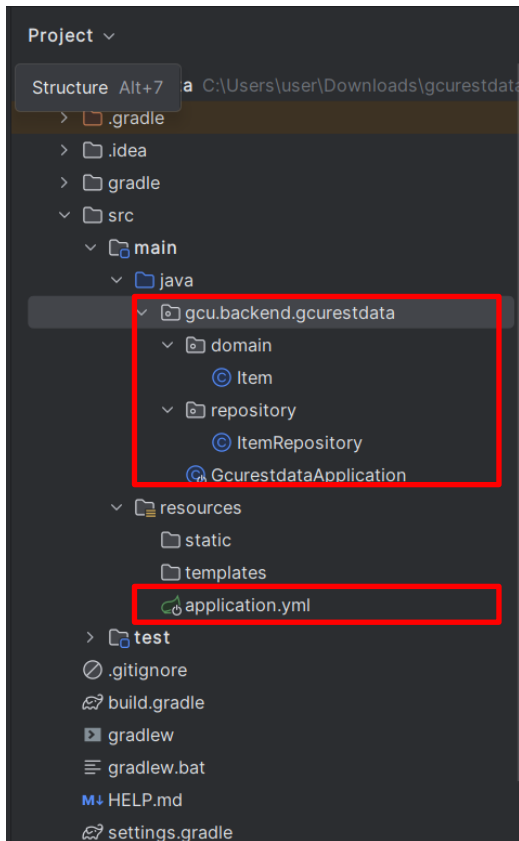
## >> domain

RESTful 서비스를 위한  
도메인 영역의 자바 클래스  
폴더

## >> repository

데이터베이스와 JPA 연동  
레포지토리 인터페이스

# 프로젝트 구조

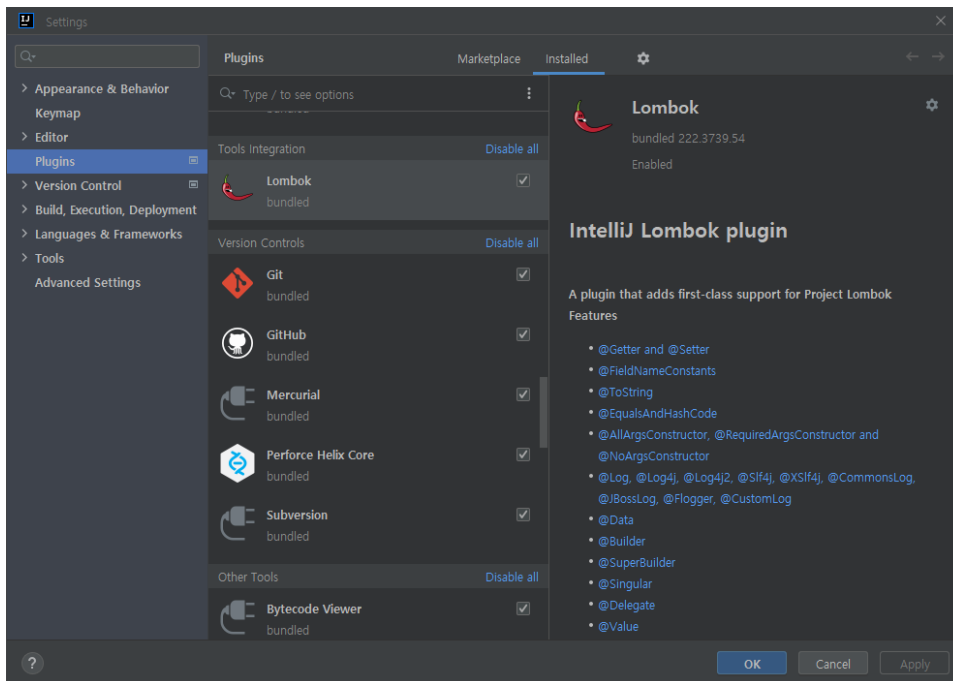


# Practice

>> **Lombok**에서 제공하는 어노테이션을 사용해서 Getter, Setter, Equals 및 ToString과 같은 메소드들을 자동 완성할 수 있습니다.

## [REF .] Lombok

### Setting - Plugins



## Practice

## 스프링 부트 어플리케이션

SpringDataRestApplication.java

**@SpringBootApplication**

```
public class SpringDataRestApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(SpringDataRestApplication.class, args);
```

```
    }
```

```
}
```



# Practice

>> **@Getter**: 프로퍼티의 Getter를 자동 생성  
**@NoArgsConstructor**(access=AccessLevel.PUBLIC):  
인수가 없는 기본 생성자의 접근 레벨을 PUBLIC으로 지정  
**@Entity, @Table**:  
데이터베이스의 관계와 테이블 매핑  
**@Id, @Column, @GeneratedValue**:  
테이블의 컬러에 해당하고, id 프로퍼티를 자동 증가하도록 지정하고 primary key로 사용  
**@Builder**: 객체 인스턴스를 생성해주는 어노테이션

# 도메인 지정

domain/Item.java

**@Getter**

**@NoArgsConstructor**(access = AccessLevel.PUBLIC)

**@Entity**

public class Item implements

**Serializable** {

**@Id**

**@Column**

**@GeneratedValue**(strategy = GenerationType.IDENTITY)

private long id;

**@Column**

private String name;

**@Column**

private int price;

**@Builder**

public Item(String name, int price) {

this.name = name;

this.price = price;

}

}

## Practice

>> **@RepositoryRestResource**

**urce:** 레스트 요청을  
처리해주기 위한 리소스를  
연결하는 레포지터리  
경로는 item으로 지정

>> **JpaRepository**를  
사용해서 Item 객체와 id의  
타입을 데이터베이스의  
테이블과 매핑하도록 함.

## 레스트 레포지터리 지정

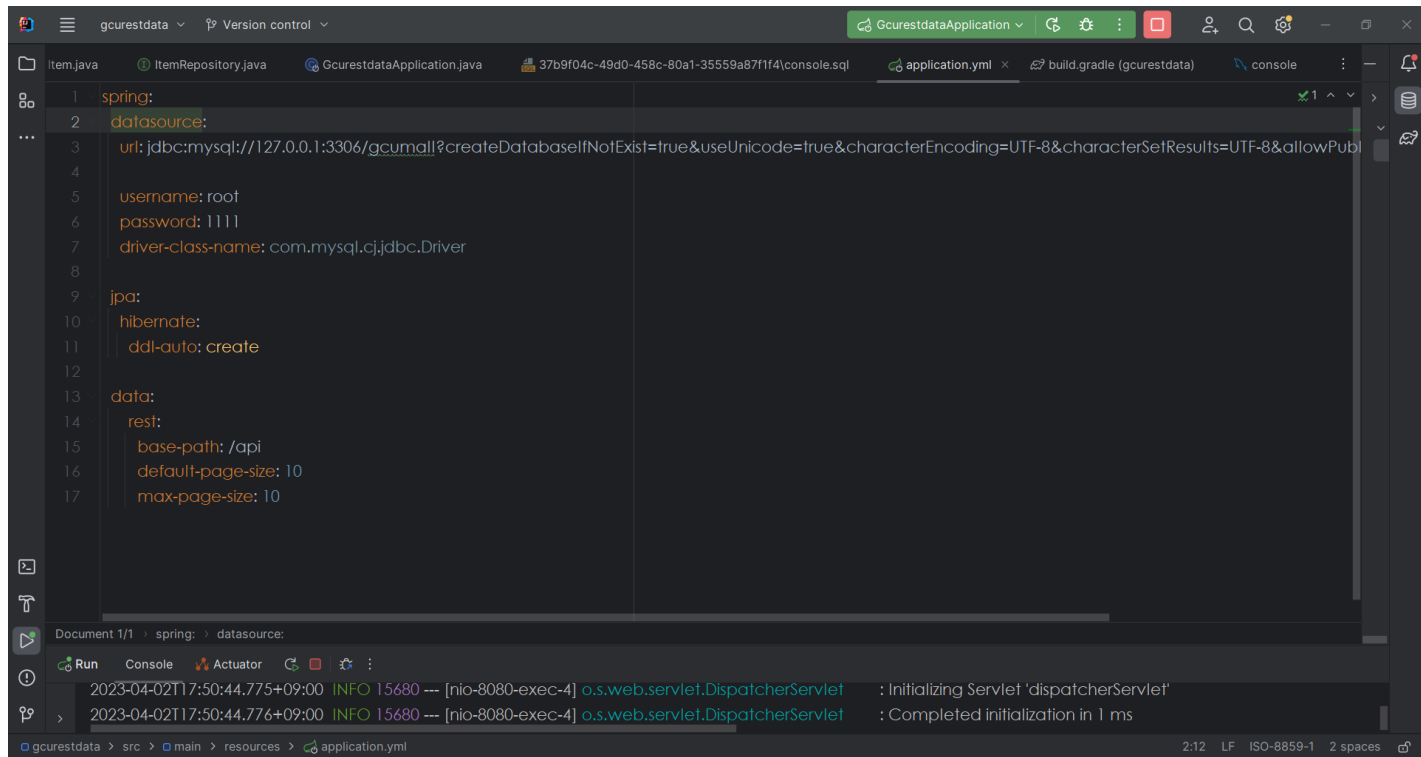
repository/ItemRepository.java

```
@RepositoryRestResource (collectionResourceRel = "item", path="item")  
public interface ItemRepository extends JpaRepository<Item, Long> {  
}
```

# Practice

>> 탭이 맞지 않으면  
테이블이 생성되지  
않습니다. 주의하시기  
바랍니다.

## application.yml 스크린 샷



```
1 spring:
2   datasource:
3     url: jdbc:mysql://112.0.0.1:3306/gcucmall?createDatabaseIfNotExist=true&useUnicode=true&characterEncoding=UTF-8&characterSetResults=UTF-8&allowPublic
4
5     username: root
6     password: 1111
7     driver-class-name: com.mysql.cj.jdbc.Driver
8
9   jpa:
10     hibernate:
11       ddl-auto: create
12
13   data:
14     rest:
15       base-path: /api
16       default-page-size: 10
17       max-page-size: 10
```

Document 1/1 · spring: > datasource:

Run Console Actuator

2023-04-02T17:50:44.775+09:00 INFO 15680 --- [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'

2023-04-02T17:50:44.776+09:00 INFO 15680 --- [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms

gcurestdata > src > main > resources > application.yml

2:12 LF ISO-8859-1 2 spaces

## Practice

>> 데이터소스연결

❶ URL을 통해 MySQL의 gcumall 데이터베이스에 연결

<조건>

- ✓ 해당 DB가 없으면 생성
- ✓ 유니코드를 사용하고 인코딩은 UTF-8
- ✓ 결과값 UTF-8 사용
- ✓ 공개키 검색을 허용
- ✓ SSL미사용

❷ 데이터베이스 접속을 위한 사용자 계정과 비밀번호 지정

❸ 데이터베이스 드라이버 이름 지정

## 데이터 소스 연결

spring:

datasource:

❶ url: jdbc:mysql://127.0.0.1:3306/gcumall?

createDatabaseIfNotExist= true

&useUnicode=true&characterEncoding=UTF-8

&characterSetResults= UTF-8

&allowPublicKeyRetrieval=true

&useSSL=false

❷ username: root

password: 1111

❸ driver-class-name: com.mysql.cj.jdbc.Driver

Resources/application.yml

## Practice

>> JPA 지정  
hibernate를 사용해서  
create쿼리를 수행하도록 함  
>> hibernate: SQL 쿼리를  
매핑해주는 도구입니다.

## JPA 연결

```
jpa:  
  hibernate:  
    ddl-auto: create
```

Resources/application.yml

## Practice

>> REST DATA

기본 루트 경로 지정

기본페이지와 최대 페이지  
크기를 지정

## REST DATA

data:

Resources/application.yml

rest:

base-path: /api

default-page-size: 10

max-page-size: 10

## Practice

>> JPA 레포지토리의  
save메소드를 사용해서  
POST 요청을 처리합니다.

# RestController와 JPA 연동

RESTful 요청을 직접작성하기 위해 Repository와 연동해봅니다.

## 레포지토리 사용

@RestController

controller/ItemController.java

```
public class ItemController {
```

```
    @Autowired
```

```
    ItemRepository itemrepository;
```

```
    @PostMapping("/api/item")
```

```
    public String add(@RequestBody Item item){
```

```
        itemrepository.save(item);
```

```
        return "update OK";
```

```
    }
```

## Practice

>> JPA 레포지토리에서 제공하는 `findAll()`과 `findById(id)` 등의 다양한 메소드를 사용할 수 있습니다.

## 레포지토리 사용

```
@GetMapping("/api/item")
public List< Item > get(){
    return itemrepository.findAll();
}
```

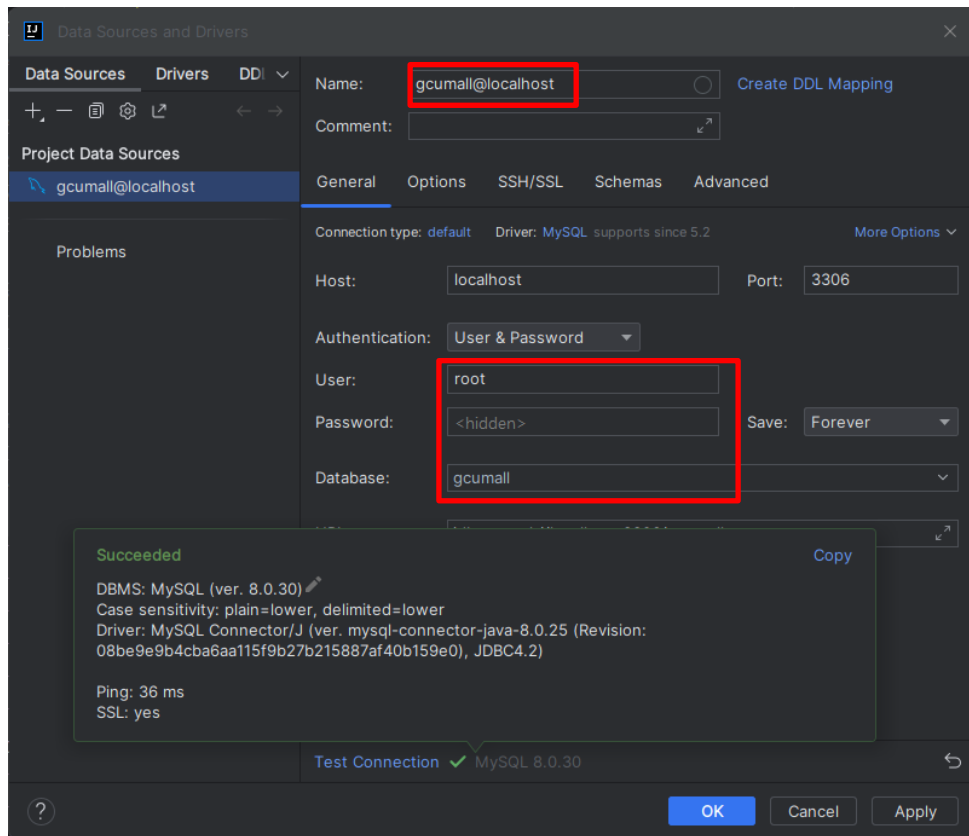
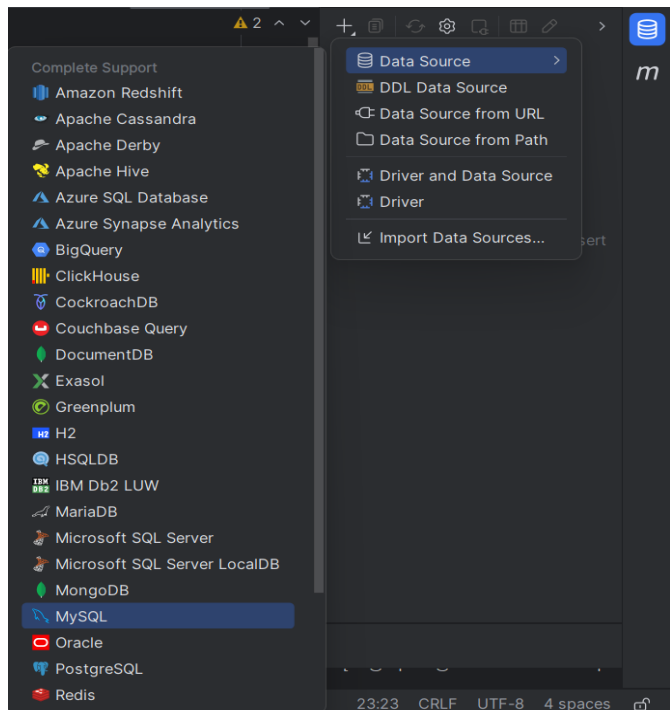
controller/ItemController.java

```
@GetMapping("/api/item/{id}")
public Optional<Item> getId(@PathVariable("id") Long id){
    return itemrepository.findById(id);
}
}
```



# Practice

# DB CONNECTION TEST



# Practice

>> POSTMAN 도구를  
사용해서 POST 요청을  
서버에 보내고 결과를  
확인합니다.

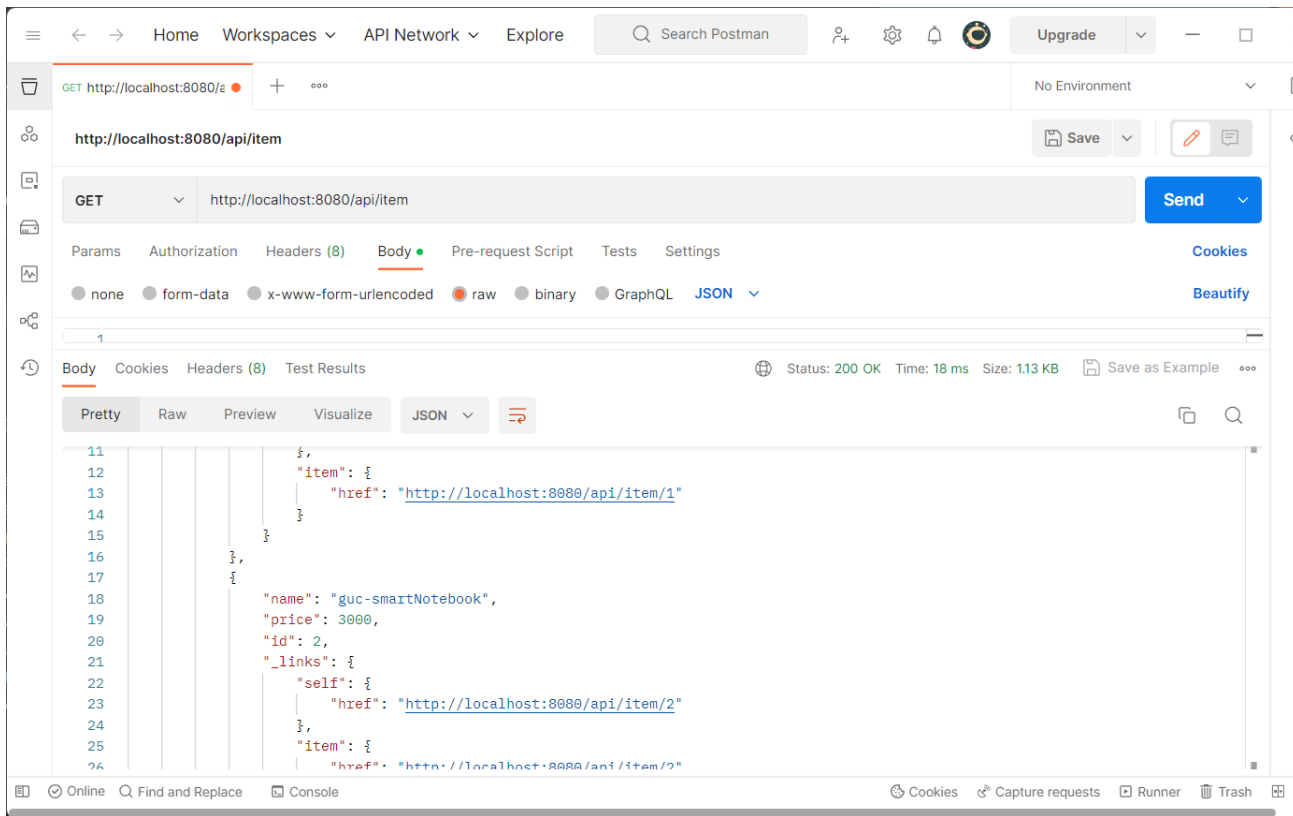
## RESTful TEST (1)

The screenshot displays the Postman application interface. At the top, the URL bar shows a POST request to `http://localhost:8080/api/item`. Below this, the 'Body' tab is selected, showing a raw JSON payload: `{ "name": "gcu-smartPhone", "price": 2000 }`. The bottom section shows the 'Test Results' tab with a status of 201 Created, and the 'Body' tab displaying the JSON response: `{ "name": "gcu-smartPhone", "price": 2000, "id": 2, "_links": { "self": { "href": "http://localhost:8080/api/item/2" }, "item": { "href": "http://localhost:8080/api/item/2" } } }`. Red rectangular boxes highlight the request URL bar, the request body, and the response body.

# Practice

>> POSTMAN 도구를  
사용해서 GET요청을  
서버에 보내고 결과를  
확인합니다.  
데이터베이스에 저장된  
항목들을 보여줍니다.

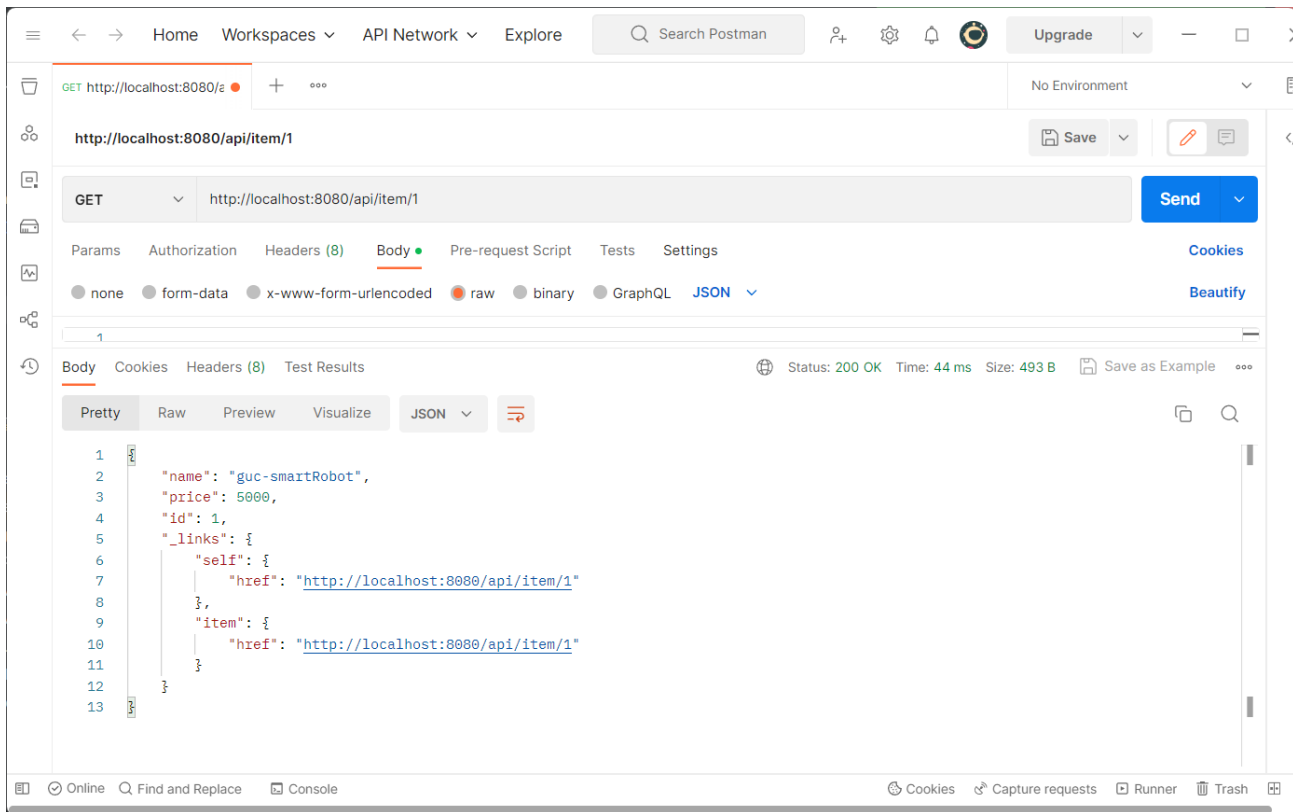
## RESTful TEST (2)



# Practice

>> 특정 ID에 대한 정보를  
데이터베이스에서  
가져와서 보여줍니다.

## RESTful TEST (3)



# MySQL Client

# IntelliJ에서 MySQL 연동 확인

```
MySQL 8.0 Command Line Cll x + v
Enter password: ****
Welcome to the MySQL monitor.  Comman
Your MySQL connection id is 496
Server version: 8.0.30 MySQL Communit

Copyright (c) 2000, 2022, Oracle and/

Oracle is a registered trademark of O
affiliates. Other names may be tradem
owners.

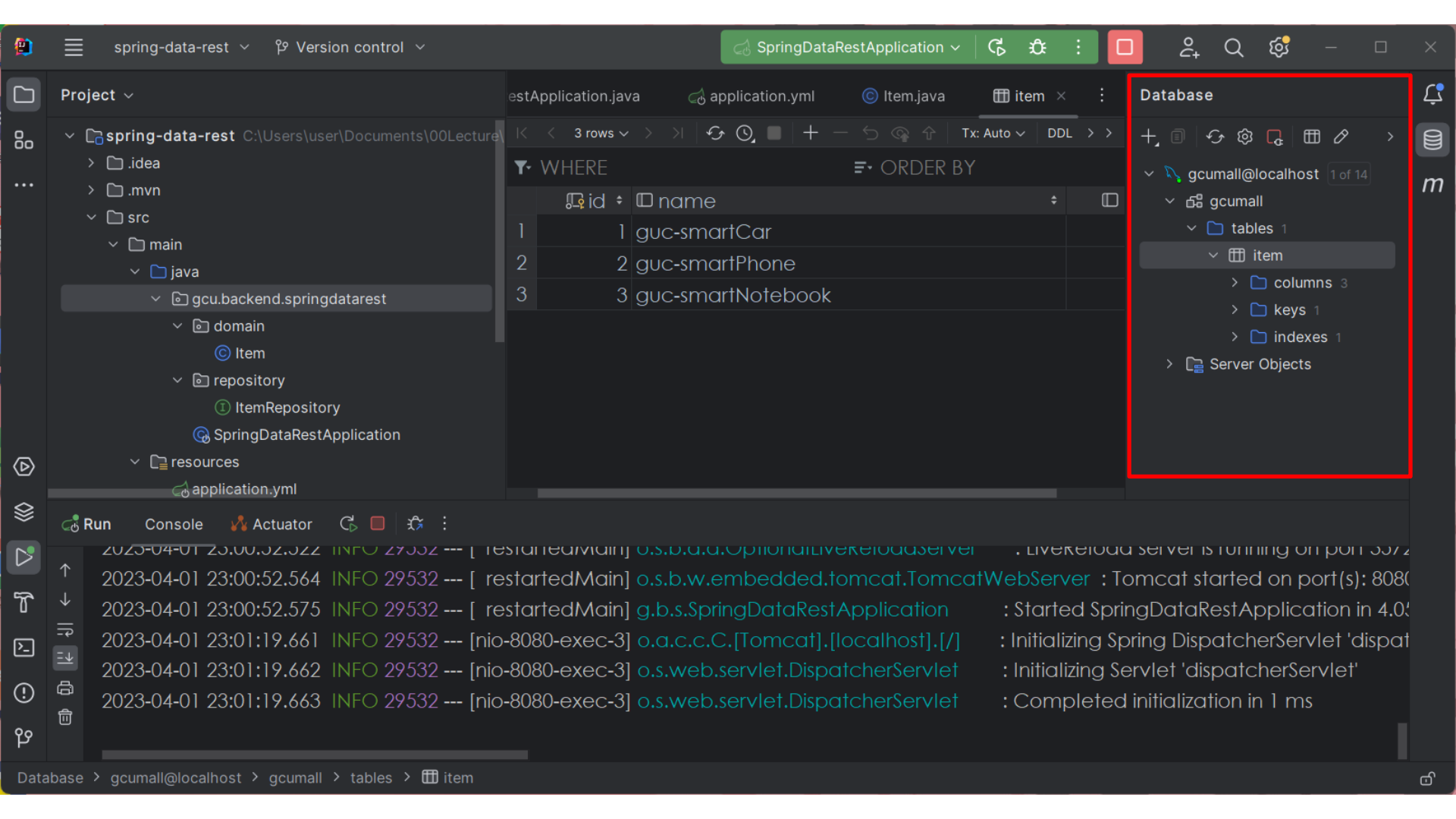
Type 'help;' or '\h' for help. Type '

mysql> use gcumall
Database changed
mysql> select * from item;
+-----+-----+-----+
| id | name          | price |
+-----+-----+-----+
| 1  | guc-smartNotebook | 3000  |
| 2  | guc-smartRobot   | 5000  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |
```

The screenshot shows the IntelliJ IDEA IDE with a Spring Data REST application. The code editor displays two SQL queries: `use gcumall;` and `select * from item;`. The Database tool window shows the 'gcumall@localhost' connection. The Services tool window shows the 'gcumall..item' table with two rows: (1, guc-smartNotebook, 3000) and (2, guc-smartRobot, 5000).

id	name	price
1	guc-smartNotebook	3000
2	guc-smartRobot	5000

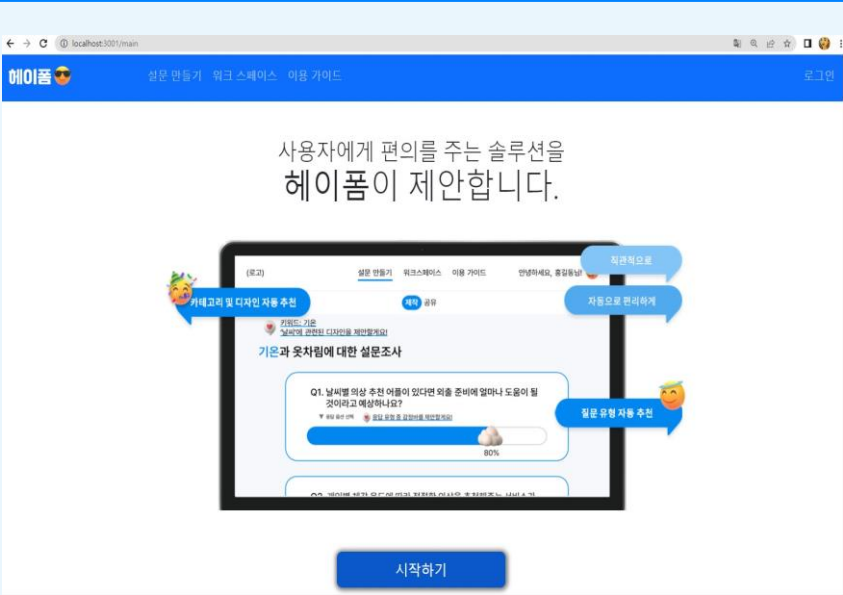


수업시간 내 진행하시기 바랍니다.

## 설문서비스 미니 프로토타입 제작

“요구분석엔지니어링을 기반으로 제작해봅니다”

- 프론트 엔드 페이지 제작 (2~3 page)
- 백 엔드 서버 연동 (RESTful 요청처리)
- 데이터베이스 연동 (DBMS 자유)
- 팀별 제출
- 평가(10점)



CONNECT.

SOLVE.

CREATE. 