SpringBoot

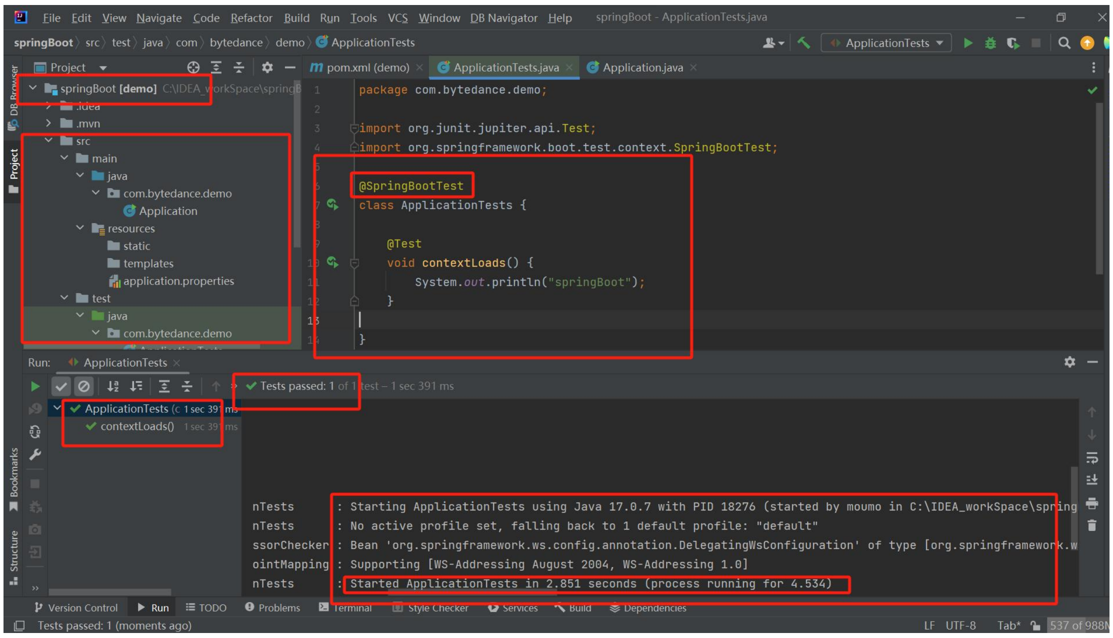
任务要求1

能够成功启动springboot空项目

1. 启动连接/生成初始项目包：https://start.spring.io/

2. SpringBoot 教程：https://www.cainiaojc.com/springboot/springboot-tutorial.html

3. 启动空 springboot 空项目：
   3.1. Output



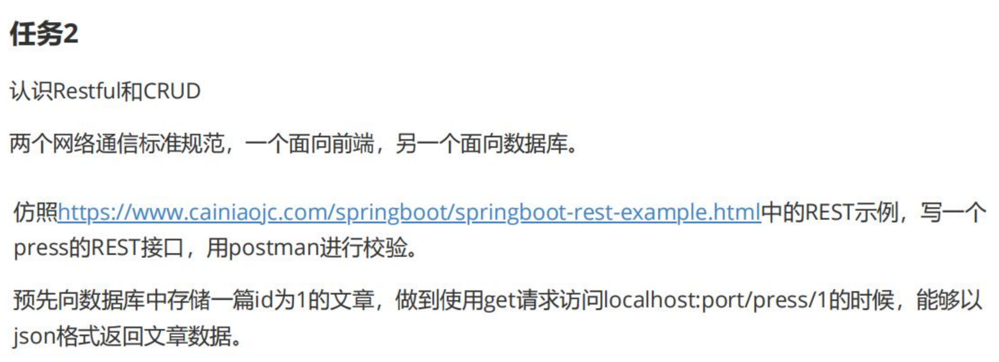
```java
package com.bytedance.demo;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class ApplicationTests {

    @Test
    void contextLoads() {
        System.out.println("springBoot");
    }

}
```

Tests passed: 1 of 1 test – 1 sec 391 ms

ApplicationTests (c 1 sec 391 ms
contextLoads() 1 sec 391 ms

```
nTests       : Starting ApplicationTests using Java 17.0.7 with PID 18276 (started by moumo in C:\IDEA_workSpace\spring
nTests       : No active profile set, falling back to 1 default profile: "default"
ssorChecker  : Bean 'org.springframework.ws.config.annotation.DelegatingWsConfiguration' of type [org.springframework.w
ointMapping  : Supporting [WS-Addressing August 2004, WS-Addressing 1.0]
nTests       : Started ApplicationTests in 2.851 seconds (process running for 4.534)
```

Tests passed: 1 (moments ago)



任务2

认识Restful和CRUD

两个网络通信标准规范，一个面向前端，另一个面向数据库。

仿照https://www.cainiaojc.com/springboot/springboot-rest-example.html中的REST示例，写一个press的REST接口，用postman进行校验。

预先向数据库中存储一篇id为1的文章，做到使用get请求访问localhost:port/press/1的时候，能够以json格式返回文章数据。

---

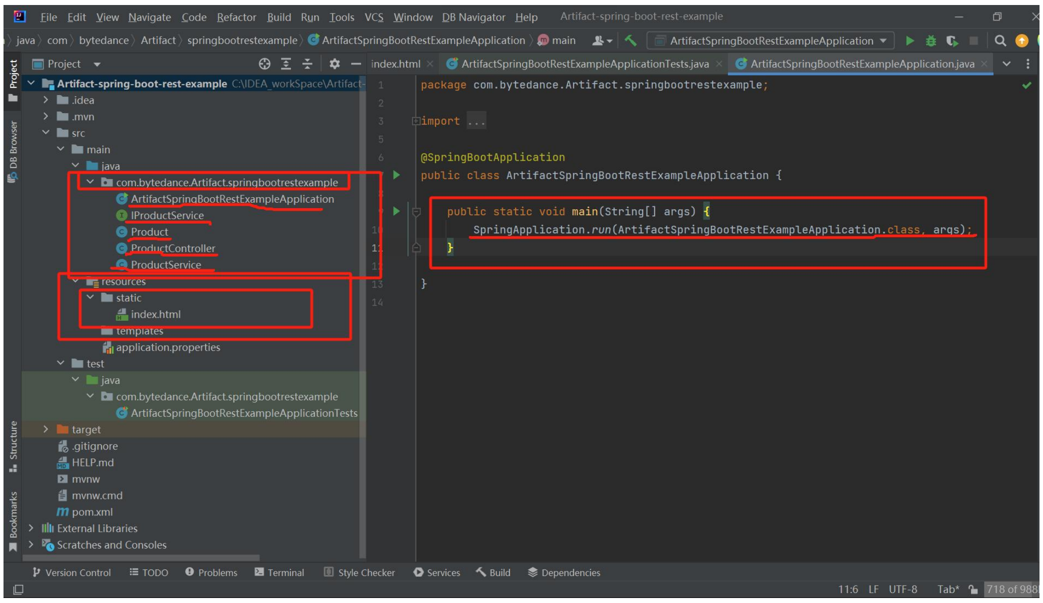
1. RESTful 架构详解/…什么是 RESTful? …/… ：
https://www.runoob.com/w3cnote/restful-architecture.html

2. crud/CRUD - (Create, Read, Update, Delete):
https://baike.baidu.com/item/crud/3606157?fr=aladdin

3. 复现“https://www.cainiaojc.com/springboot/springboot-rest-example.html”中的 REST 示例：

(Using using an in-memory list instead of using a database like MySQL)

3.1. In ‘C:\IDEA_workSpace\Artifact-spring-boot-rest-example’:

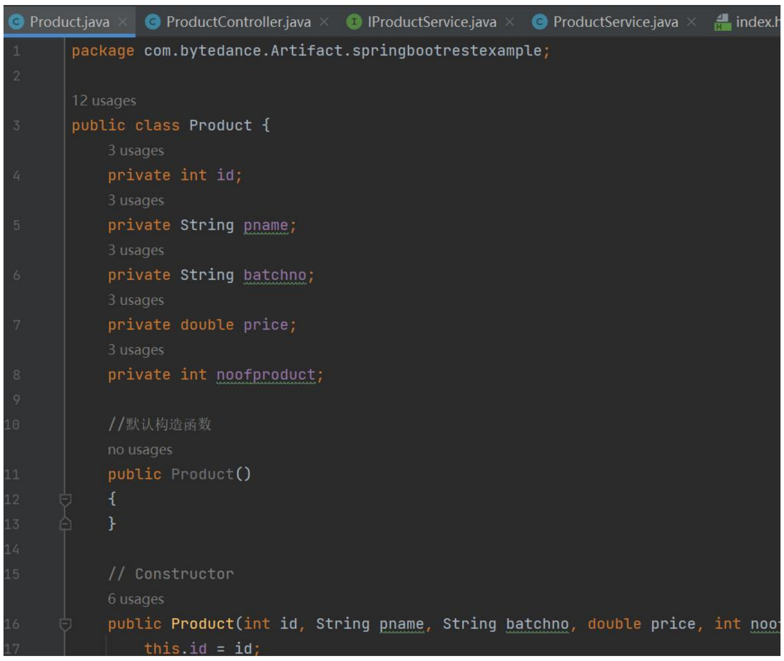


```
Artifact-spring-boot-rest-example  C:\IDEA_workSpace\Artifact-
  .idea
  .mvn
  src
    main
      java
        com.bytedance.Artifact.springbootrestexample
          ArtifactSpringBootRestExampleApplication
          IProductService
          Product
          ProductController
          ProductService
      resources
        static
          index.html
        templates
        application.properties
    test
      java
        com.bytedance.Artifact.springbootrestexample
          ArtifactSpringBootRestExampleApplicationTests
  target
  .gitignore
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
External Libraries
Scratches and Consoles
```

```java
package com.bytedance.Artifact.springbootrestexample;

import ...

@SpringBootApplication
public class ArtifactSpringBootRestExampleApplication {

    public static void main(String[] args) {
        SpringApplication.run(ArtifactSpringBootRestExampleApplication.class, args);
    }

}
```

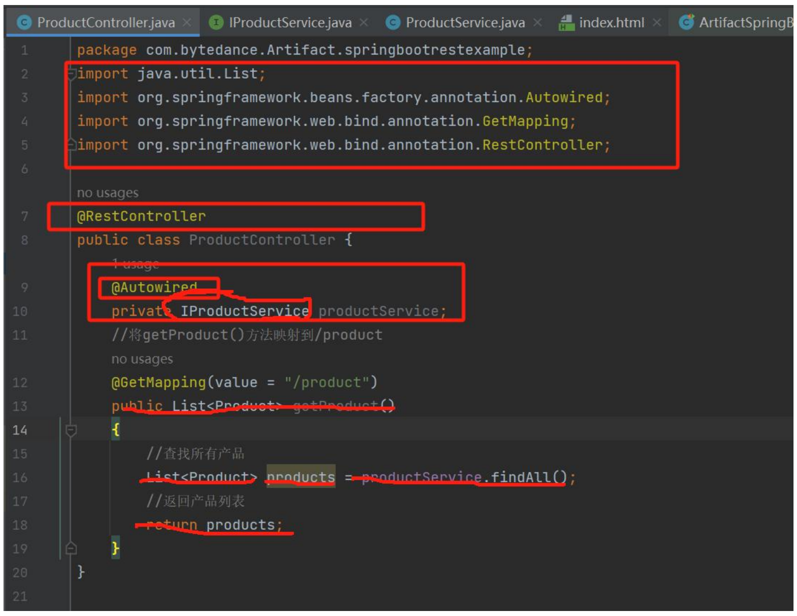
File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Artifact-spring-boot-rest-example
java 〉com 〉bytedance 〉Artifact 〉springbootrestexample 〉ArtifactSpringBootRestExampleApplication 〉main
ArtifactSpringBootRestExampleApplication
Project
index.html
ArtifactSpringBootRestExampleApplicationTests.java
ArtifactSpringBootRestExampleApplication.java
Version Control
TODO
Problems
Terminal
Style Checker
Services
Build
Dependencies
11:6 LF UTF-8 Tab* 718 of 988

3.2. Product.java:



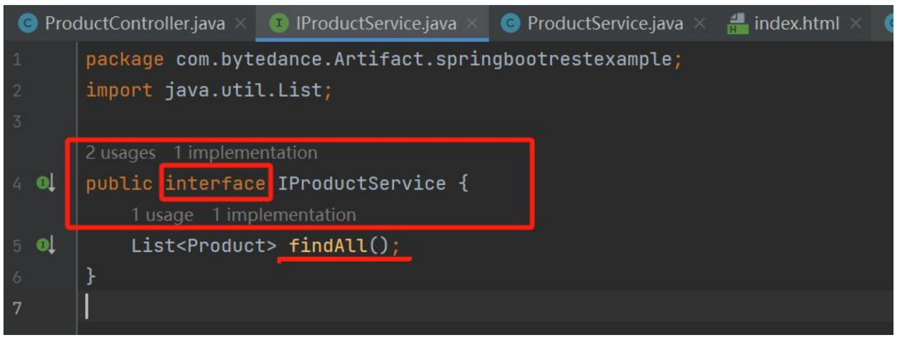
```java
package com.bytedance.Artifact.springbootrestexample;

12 usages
public class Product {
    3 usages
    private int id;
    3 usages
    private String pname;
    3 usages
    private String batchno;
    3 usages
    private double price;
    3 usages
    private int noofproduct;

    //默认构造函数
    no usages
    public Product()
    {
    }

    // Constructor
    6 usages
    public Product(int id, String pname, String batchno, double price, int noo
        this.id = id;
```

3.3. ProductController.java:



```java
package com.bytedance.Artifact.springbootrestexample;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

no usages
@RestController
public class ProductController {
    1 usage
    @Autowired
    private IProductService productService;
    //将getProduct()方法映射到/product
    no usages
    @GetMapping(value = "/product")
    public List<Product> getProduct()
    {
        //查找所有产品
        List<Product> products = productService.findAll();
        //返回产品列表
        return products;
    }
}
```

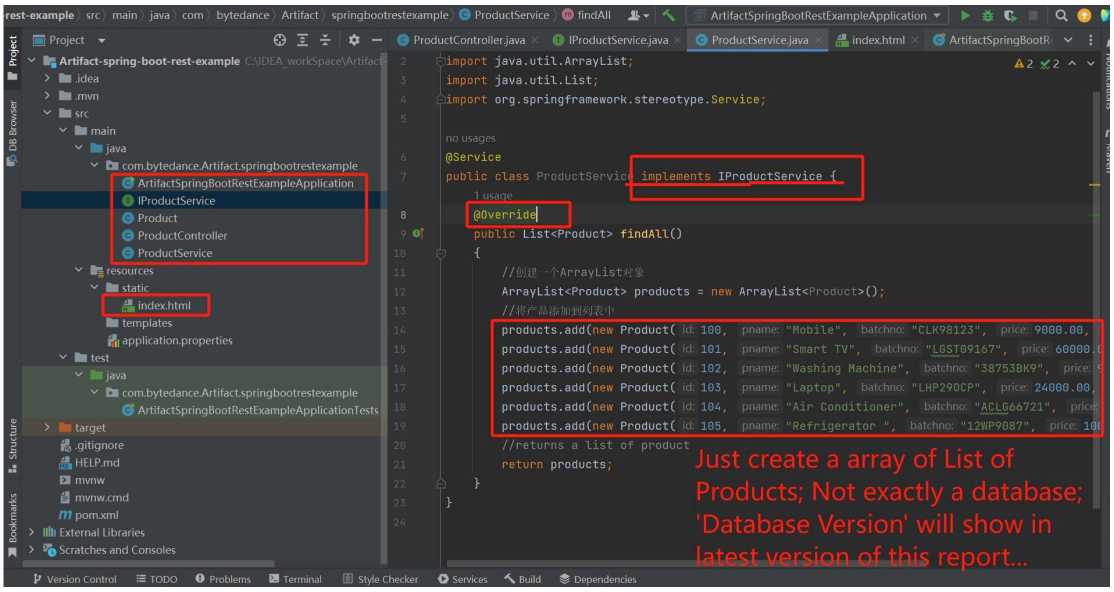
3.4. IProductService(Interface_Product_Service):



```java
package com.bytedance.Artifact.springbootrestexample;
import java.util.List;

2 usages   1 implementation
public interface IProductService {
    1 usage   1 implementation
    List<Product> findAll();
}
```

3.5. ProductService:

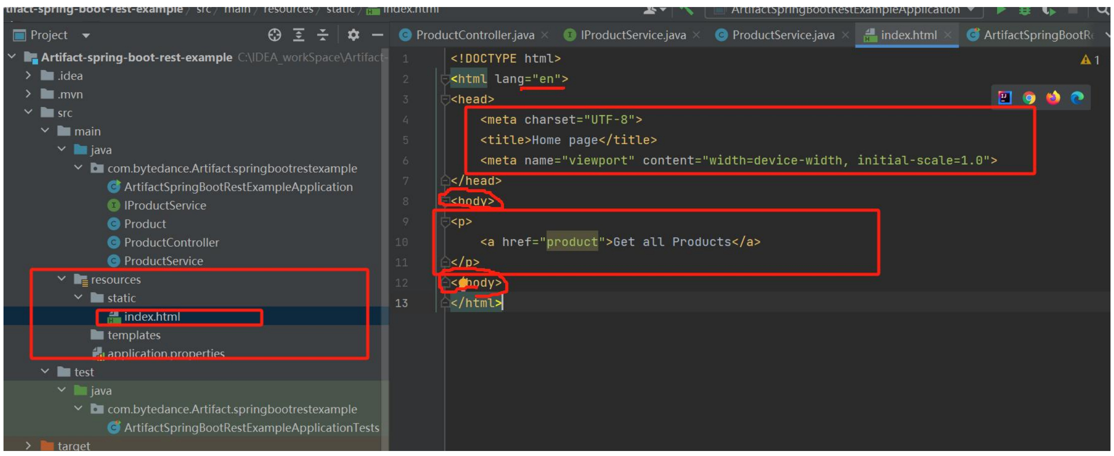
(Using using an in-memory list instead of using a database like MySQL)



```java
import java.util.ArrayList;
import java.util.List;
import org.springframework.stereotype.Service;

no usages
@Service
public class ProductService implements IProductService {
    1 usage
    @Override
    public List<Product> findAll()
    {
        //创建一个ArrayList对象
        ArrayList<Product> products = new ArrayList<Product>();
        //将产品添加到列表中
        products.add(new Product( id: 100,  pname: "Mobile",  batchno: "CLK98123",  price: 9000.00,
        products.add(new Product( id: 101,  pname: "Smart TV",  batchno: "LGST09167",  price: 60000.
        products.add(new Product( id: 102,  pname: "Washing Machine",  batchno: "38753BK9",  price:
        products.add(new Product( id: 103,  pname: "Laptop",  batchno: "LHP29OCP",  price: 24000.00,
        products.add(new Product( id: 104,  pname: "Air Conditioner",  batchno: "ACLG66721",  price:
        products.add(new Product( id: 105,  pname: "Refrigerator ",  batchno: "12WP9087",  price: 10
        //returns a list of product
        return products;
    }
}
```

Just create a array of List of Products; Not exactly a database; 'Database Version' will show in latest version of this report...

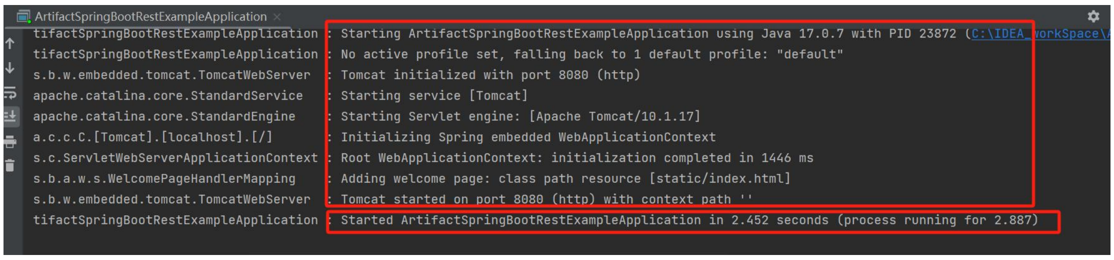
3.6. Index.html:



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Home page</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<p>
    <a href="product">Get all Products</a>
</p>
</body>
</html>
```

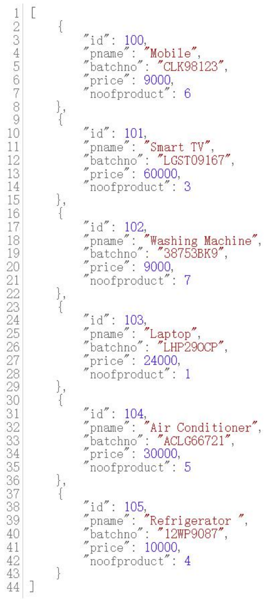
3.7. Output:

- Terminal



```
tifactSpringBootRestExampleApplication : Starting ArtifactSpringBootRestExampleApplication using Java 17.0.7 with PID 23872 (C:\IDEA_workSpace\
tifactSpringBootRestExampleApplication : No active profile set, falling back to 1 default profile: "default"
s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
apache.catalina.core.StandardService   : Starting service [Tomcat]
apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.17]
a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1446 ms
s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page: class path resource [static/index.html]
s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path ''
tifactSpringBootRestExampleApplication : Started ArtifactSpringBootRestExampleApplication in 2.452 seconds (process running for 2.887)
```

• 打开浏览器并调用 URL http://localhost:8080/index.html。它显示了 获取所有产品的链接，如下图所示:



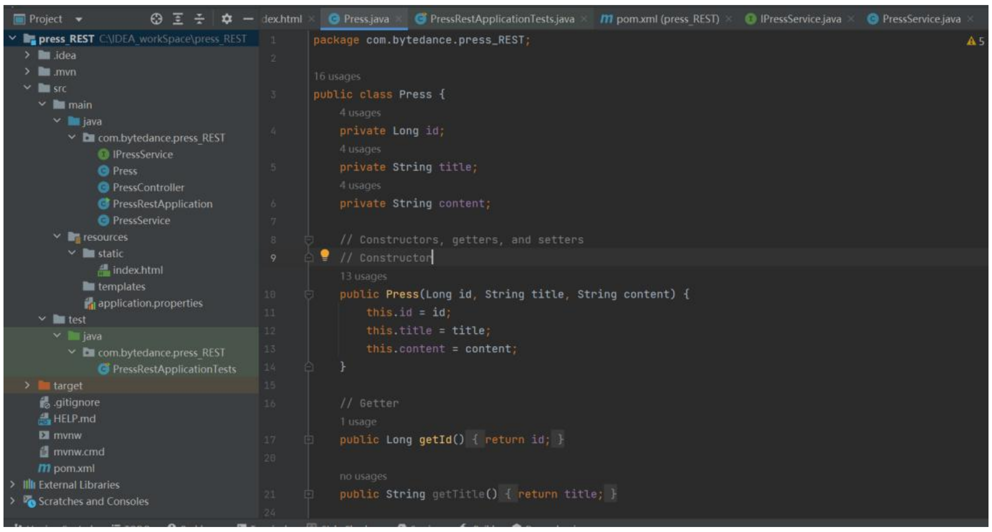
```
[
    {
        "id": 100,
        "pname": "Mobile",
        "batchno": "CLK98123",
        "price": 9000,
        "noofproduct": 6
    },
    {
        "id": 101,
        "pname": "Smart TV",
        "batchno": "LGST09167",
        "price": 60000,
        "noofproduct": 3
    },
    {
        "id": 102,
        "pname": "Washing Machine",
        "batchno": "38753BK9",
        "price": 9000,
        "noofproduct": 7
    },
    {
        "id": 103,
        "pname": "Laptop",
        "batchno": "LHP29OCP",
        "price": 24000,
        "noofproduct": 1
    },
    {
        "id": 104,
        "pname": "Air Conditioner",
        "batchno": "ACLG66721",
        "price": 30000,
        "noofproduct": 5
    },
    {
        "id": 105,
        "pname": "Refrigerator ",
        "batchno": "12WP9087",
        "price": 10000,
        "noofproduct": 4
    }
]
```

4. 仿照 https://www.cainiaojc.com/springboot/springboot-rest-example.html 中的 REST 示例，写一个 press 的 REST 接口，用 postman 进行校验(疑问：如何'用 postman 进行校验？'，是类似于 crul...命令吗？):

(Using using an in-memory list instead of using a database like MySQL)

"C:\IDEA_workSpace\press_REST":

4.1. JavaBean creation: (Press.java)

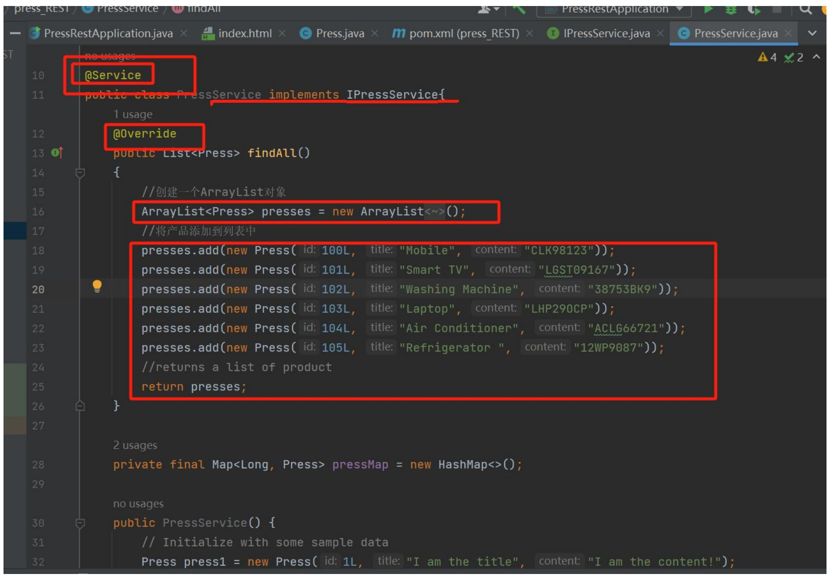


```
package com.bytedance.press_REST;

16 usages
public class Press {
    4 usages
    private Long id;
    4 usages
    private String title;
    4 usages
    private String content;

    // Constructors, getters, and setters
    // Constructor
    13 usages
    public Press(Long id, String title, String content) {
        this.id = id;
        this.title = title;
        this.content = content;
    }

    // Getter
    1 usage
    public Long getId() { return id; }

    no usages
    public String getTitle() { return title; }
```

...

• PressService.java:

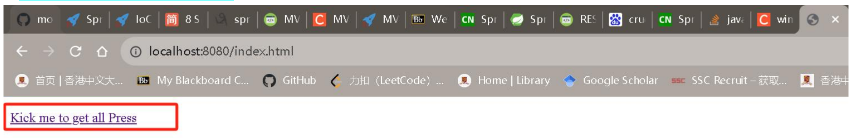
(Using using an in-memory list instead of using a database like MySQL)



```java
no usages
@Service
public class PressService implements IPressService{
    1 usage
    @Override
    public List<Press> findAll()
    {
        //创建一个ArrayList对象
        ArrayList<Press> presses = new ArrayList<~>();
        //将产品添加到列表中
        presses.add(new Press( id: 100L,  title: "Mobile",  content: "CLK98123"));
        presses.add(new Press( id: 101L,  title: "Smart TV",  content: "LGST09167"));
        presses.add(new Press( id: 102L,  title: "Washing Machine",  content: "38753BK9"));
        presses.add(new Press( id: 103L,  title: "Laptop",  content: "LHP29OCP"));
        presses.add(new Press( id: 104L,  title: "Air Conditioner",  content: "ACLG66721"));
        presses.add(new Press( id: 105L,  title: "Refrigerator ",  content: "12WP9087"));
        //returns a list of product
        return presses;
    }

    2 usages
    private final Map<Long, Press> pressMap = new HashMap<>();

    no usages
    public PressService() {
        // Initialize with some sample data
        Press press1 = new Press( id: 1L,  title: "I am the title",  content: "I am the content!");
```

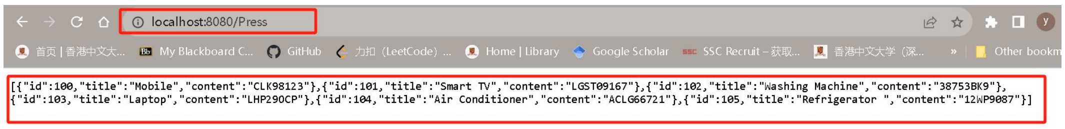
• 打开浏览器并调用 URL http://localhost:8080/index.html ('index.html' here is a folder under "C:\IDEA_workSpace\press-REST_databaseVersion\src\main\resources\static\index.html" !)。它显示了 获取所有产品的链接，如下图所示:

- In Google Chrome:



localhost:8080/index.html

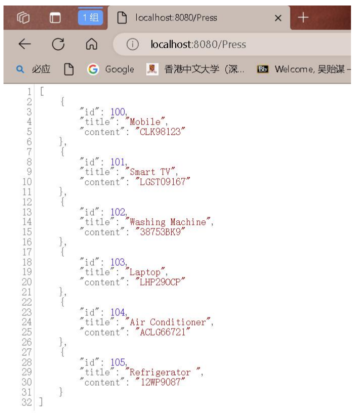
Kick me to get all Press



localhost:8080/Press

```
[{"id":100,"title":"Mobile","content":"CLK98123"},{"id":101,"title":"Smart TV","content":"LGST09167"},{"id":102,"title":"Washing Machine","content":"38753BK9"},
{"id":103,"title":"Laptop","content":"LHP29OCP"},{"id":104,"title":"Air Conditioner","content":"ACLG66721"},{"id":105,"title":"Refrigerator ","content":"12WP9087"}]
```

- In Microsoft Edge:



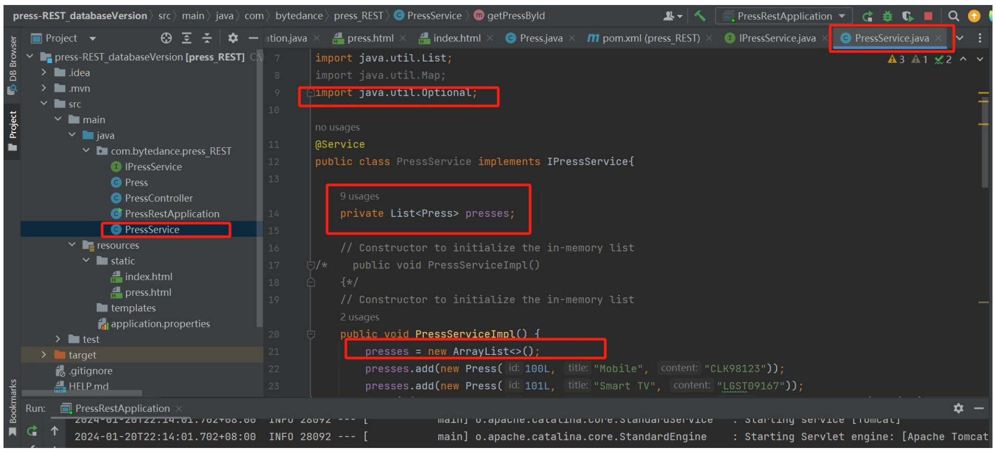
localhost:8080/Press

```
[
    {
        "id": 100,
        "title": "Mobile",
        "content": "CLK98123"
    },
    {
        "id": 101,
        "title": "Smart TV",
        "content": "LGST09167"
    },
    {
        "id": 102,
        "title": "Washing Machine",
        "content": "38753BK9"
    },
    {
        "id": 103,
        "title": "Laptop",
        "content": "LHP29OCP"
    },
    {
        "id": 104,
        "title": "Air Conditioner",
        "content": "ACLG66721"
    },
    {
        "id": 105,
        "title": "Refrigerator ",
        "content": "12WP9087"
    }
]
```

5. - 仿照 https://www.cainiaojc.com/springboot/springboot-rest-example.html 中的 REST 示例，写一个 press 的 REST 接口，用 postman 进行校验。
- 预先向数据库中存储一篇 id 为 1 的文章，做到使用 get 请求访问 localhost:port/press/1(localhost:8080/press/1 here, since)的时候，能够以 json 格式返回文章数据。

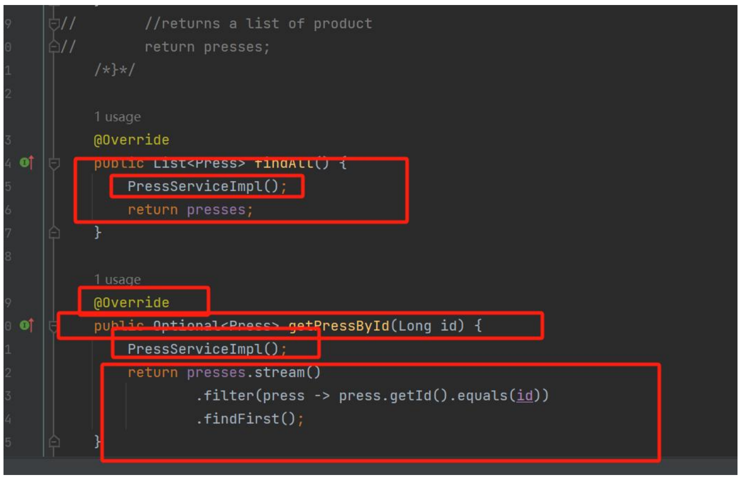
Modify parts:

5.1. PressService.java:



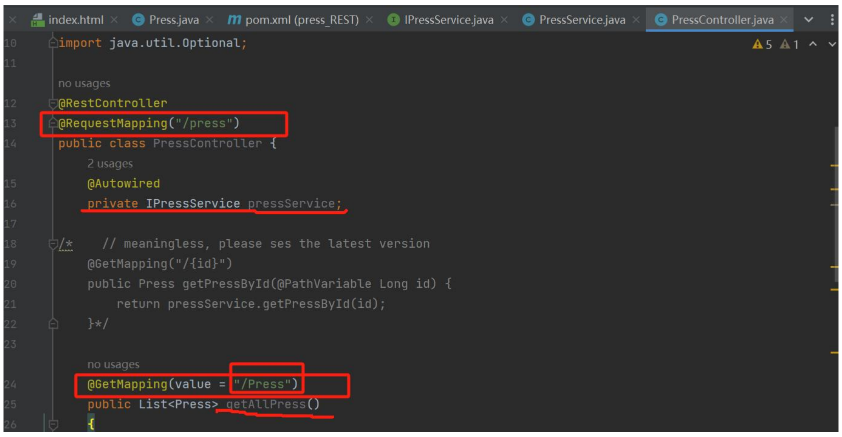
```java
import java.util.List;
import java.util.Map;
import java.util.Optional;

no usages
@Service
public class PressService implements IPressService{

    9 usages
    private List<Press> presses;

    // Constructor to initialize the in-memory list
/*    public void PressServiceImpl()
    {*/
    // Constructor to initialize the in-memory list
    2 usages
    public void PressServiceImpl() {
        presses = new ArrayList<>();
        presses.add(new Press( id: 100L,  title: "Mobile",  content: "CLK98123"));
        presses.add(new Press( id: 101L,  title: "Smart TV",  content: "LGST09167"));
```

```
2024-01-20T22:14:01.702+08:00  INFO 28092 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat
```

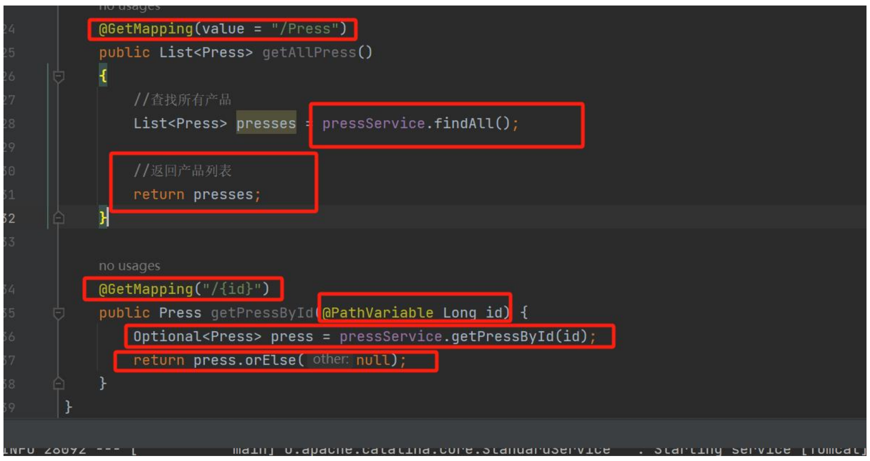


```java
//        //returns a list of product
//        return presses;
    /*}*/

    1 usage
    @Override
    public List<Press> findAll() {
        PressServiceImpl();
        return presses;
    }

    1 usage
    @Override
    public Optional<Press> getPressById(Long id) {
        PressServiceImpl();
        return presses.stream()
                .filter(press -> press.getId().equals(id))
                .findFirst();
    }
```

5.2. PressController.java:



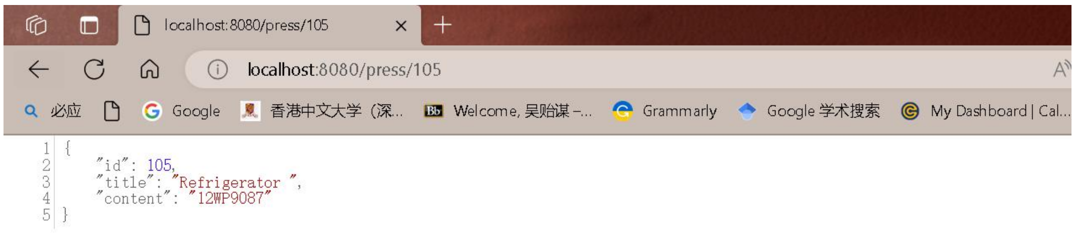
```java
import java.util.Optional;

no usages
@RestController
@RequestMapping("/press")
public class PressController {
    2 usages
    @Autowired
    private IPressService pressService;

/*    // meaningless, please ses the latest version
    @GetMapping("/{id}")
    public Press getPressById(@PathVariable Long id) {
        return pressService.getPressById(id);
    }*/

    no usages
    @GetMapping(value = "/Press")
    public List<Press> getAllPress()
    {
```



```
    @GetMapping(value = "/Press")
    public List<Press> getAllPress()
    {
        //查找所有产品
        List<Press> presses = pressService.findAll();

        //返回产品列表
        return presses;
    }

    no usages
    @GetMapping("/{id}")
    public Press getPressById(@PathVariable Long id) {
        Optional<Press> press = pressService.getPressById(id);
        return press.orElse( other: null);
    }
}
```

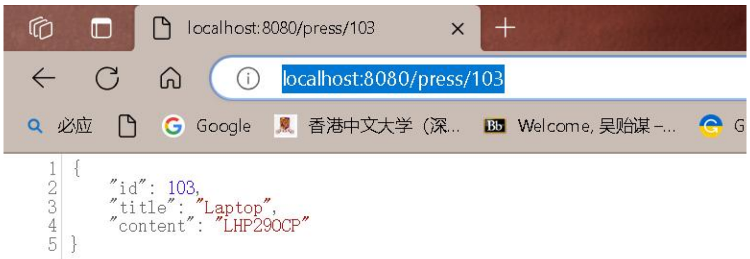
5.3. Output

"localhost:8080/press/105":



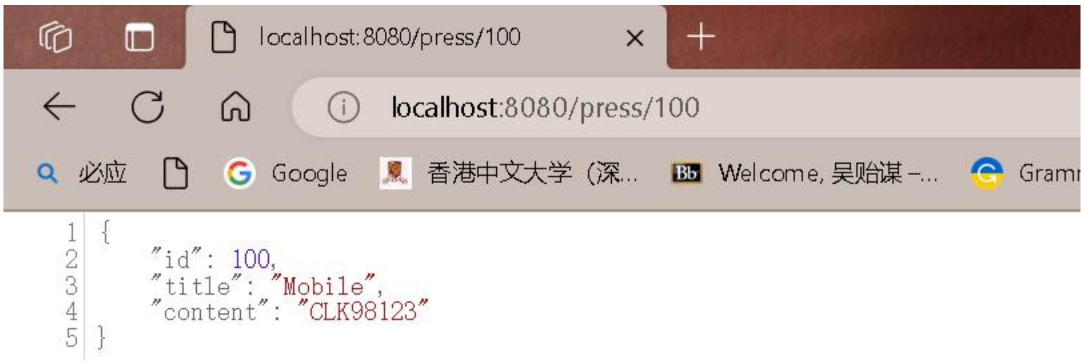
```
{
    "id": 105,
    "title": "Refrigerator ",
    "content": "12WP9087"
}
```

“localhost:8080/press/103”:



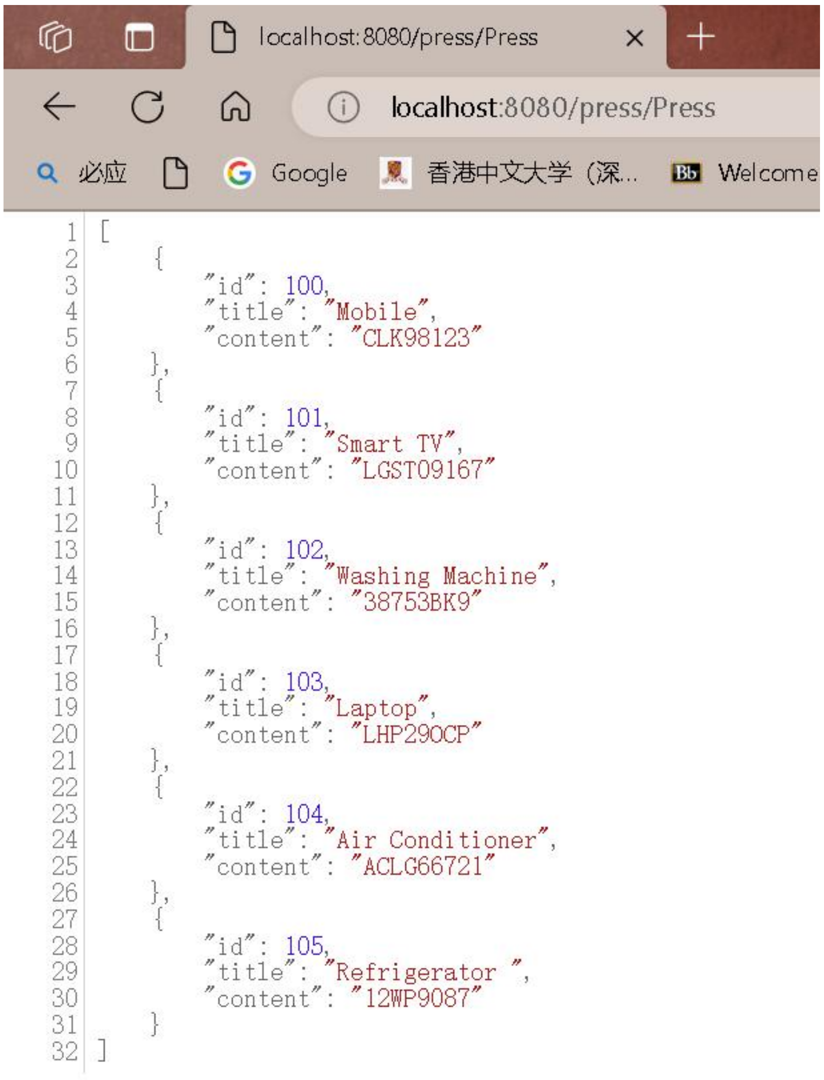
```
{
    "id": 103,
    "title": "Laptop",
    "content": "LHP290CP"
}
```

“localhost:8080/press/100”:



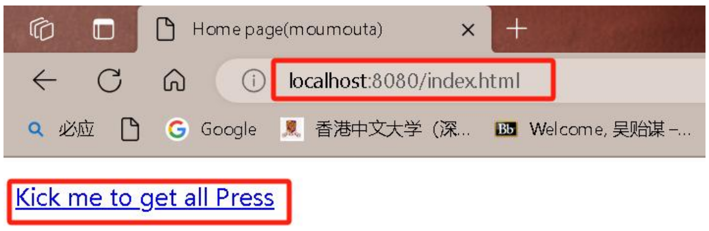
```
{
    "id": 100,
    "title": "Mobile",
    "content": "CLK98123"
}
```

"localhost:8080/press/Press":



```
[
    {
        "id": 100,
        "title": "Mobile",
        "content": "CLK98123"
    },
    {
        "id": 101,
        "title": "Smart TV",
        "content": "LGST09167"
    },
    {
        "id": 102,
        "title": "Washing Machine",
        "content": "38753BK9"
    },
    {
        "id": 103,
        "title": "Laptop",
        "content": "LHP29OCP"
    },
    {
        "id": 104,
        "title": "Air Conditioner",
        "content": "ACLG66721"
    },
    {
        "id": 105,
        "title": "Refrigerator ",
        "content": "12WP9087"
    }
]
```

"Home page(moumouta)" / "localhost:8080/index.html"

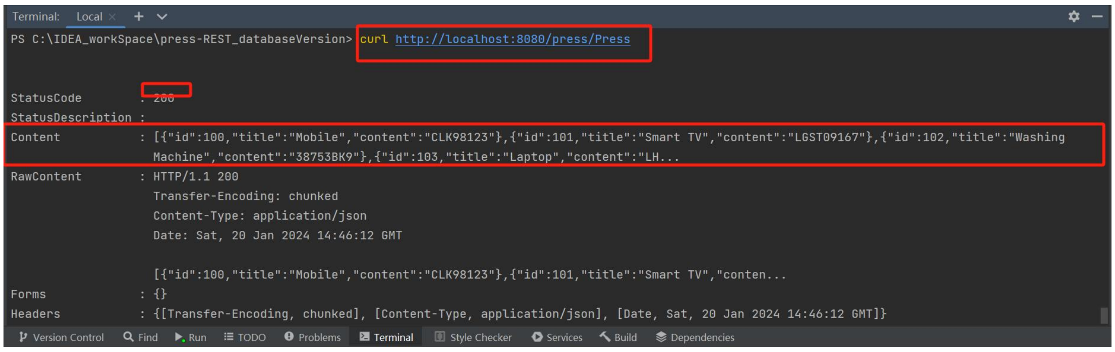


Kick me to get all Press

After Kick the link "Kick me to get all Press":

Same as "localhost:8080/press/Press" page.

- 用 postman 进行校验



```
PS C:\IDEA_workSpace\press-REST_databaseVersion> curl http://localhost:8080/press/Press


StatusCode        : 200
StatusDescription :
Content           : [{"id":100,"title":"Mobile","content":"CLK98123"},{"id":101,"title":"Smart TV","content":"LGST09167"},{"id":102,"title":"Washing
                    Machine","content":"38753BK9"},{"id":103,"title":"Laptop","content":"LH...
RawContent        : HTTP/1.1 200
                    Transfer-Encoding: chunked
                    Content-Type: application/json
                    Date: Sat, 20 Jan 2024 14:46:12 GMT

                    [{"id":100,"title":"Mobile","content":"CLK98123"},{"id":101,"title":"Smart TV","conten...
Forms             : {}
Headers           : {[Transfer-Encoding, chunked], [Content-Type, application/json], [Date, Sat, 20 Jan 2024 14:46:12 GMT]}
```

- 缺点/未完成的地方:

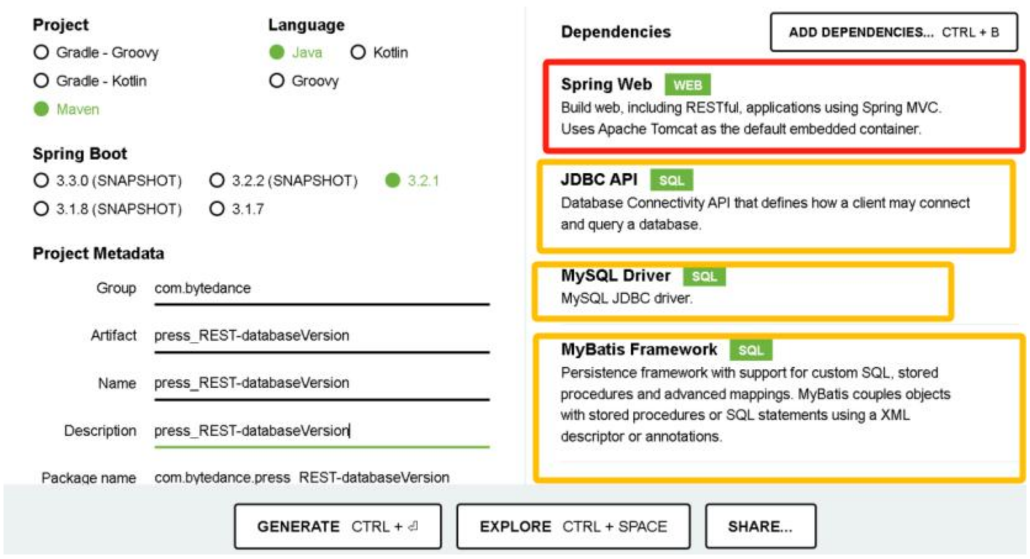
只能实现 in-memory list 而不是 using a database like MySQL。

在网上查了如何（远程）连接（绑定）SpringBoot 和 MySQL 数据库，但实现过程中出现了较大阻碍。

请老师帮忙指点下~(已解决)

- Database Version:

1. Spring initializr



Project
- Gradle - Groovy
- Gradle - Kotlin
- [✓ Maven]

Language
- [✓ Java]
- Kotlin
- Groovy

Spring Boot
- 3.3.0 (SNAPSHOT)
- 3.2.2 (SNAPSHOT)
- [✓ 3.2.1]
- 3.1.8 (SNAPSHOT)
- 3.1.7

Project Metadata

Group	com.bytedance
Artifact	press_REST-databaseVersion
Name	press_REST-databaseVersion
Description	press_REST-databaseVersion
Package name	com.bytedance.press_REST-databaseVersion

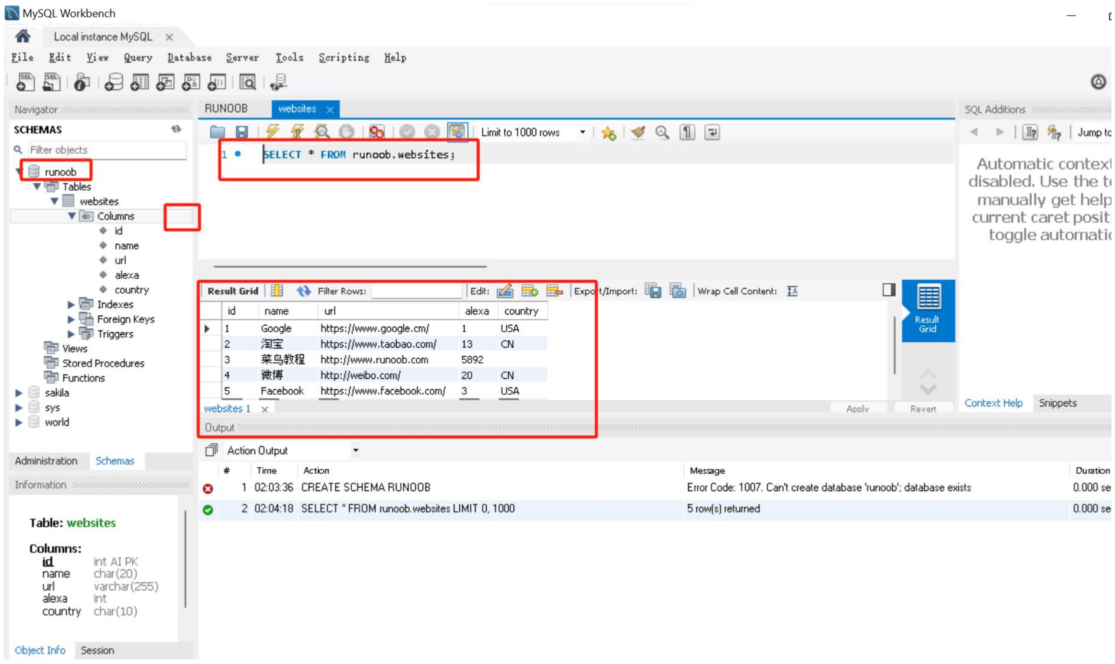
Dependencies

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

JDBC API SQL
Database Connectivity API that defines how a client may connect and query a database.

MySQL Driver SQL
MySQL JDBC driver.

MyBatis Framework SQL
Persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis couples objects with stored procedures or SQL statements using a XML descriptor or annotations.



```
SELECT * FROM runoob.websites;
```

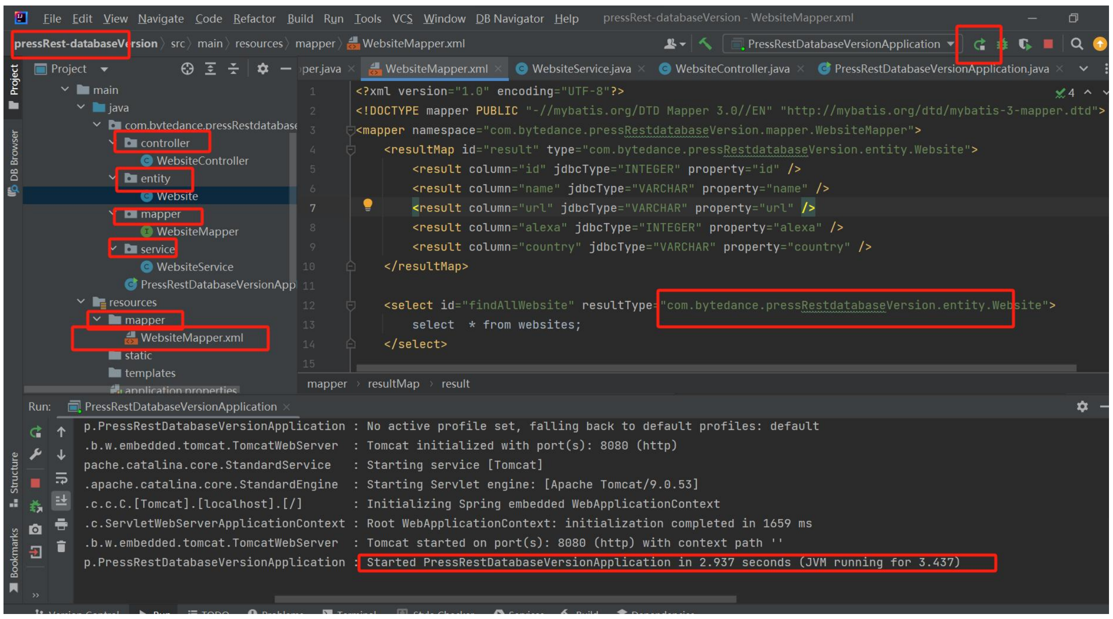
id	name	url	alexa	country
1	Google	https://www.google.cm/	1	USA
2	淘宝	https://www.taobao.com/	13	CN
3	菜鸟教程	http://www.runoob.com	5892	
4	微博	http://weibo.com/	20	CN
5	Facebook	https://www.facebook.com/	3	USA

#	Time	Action	Message	Duration
1	02:03:36	CREATE SCHEMA RUNOOB	Error Code: 1007. Can't create database 'runoob'; database exists	0.000 se
2	02:04:18	SELECT * FROM runoob.websites LIMIT 0, 1000	5 row(s) returned	0.000 se

Table: websites

Columns:

id	int AI PK
name	char(20)
url	varchar(255)
alexa	int
country	char(10)



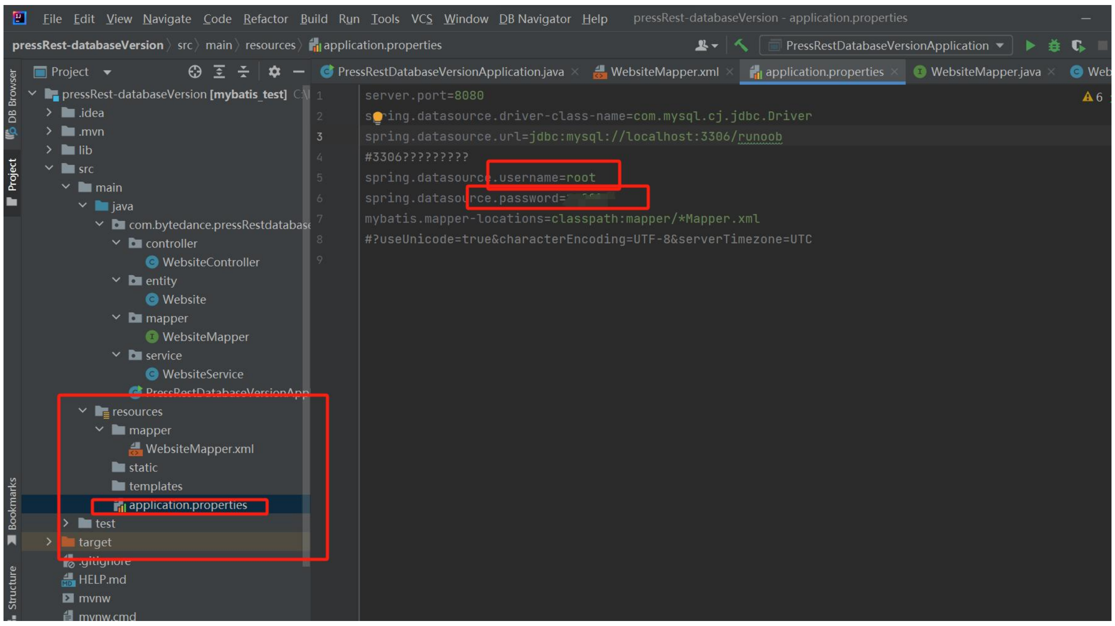
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.bytedance.pressRestdatabaseVersion.mapper.WebsiteMapper">
    <resultMap id="result" type="com.bytedance.pressRestdatabaseVersion.entity.Website">
        <result column="id" jdbcType="INTEGER" property="id" />
        <result column="name" jdbcType="VARCHAR" property="name" />
        <result column="url" jdbcType="VARCHAR" property="url" />
        <result column="alexa" jdbcType="INTEGER" property="alexa" />
        <result column="country" jdbcType="VARCHAR" property="country" />
    </resultMap>

    <select id="findAllWebsite" resultType="com.bytedance.pressRestdatabaseVersion.entity.Website">
        select  * from websites;
    </select>
```

```
p.PressRestDatabaseVersionApplication : No active profile set, falling back to default profiles: default
.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
pache.catalina.core.StandardService   : Starting service [Tomcat]
.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.53]
.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1659 ms
.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
p.PressRestDatabaseVersionApplication : Started PressRestDatabaseVersionApplication in 2.937 seconds (JVM running for 3.437)
```

Reference:

https://blog.csdn.net/YangMax1/article/details/120757964?spm=1001.2014.3001.5501

https://blog.csdn.net/YouthBlood9/article/details/120829154 “application.properties”：（!!!）



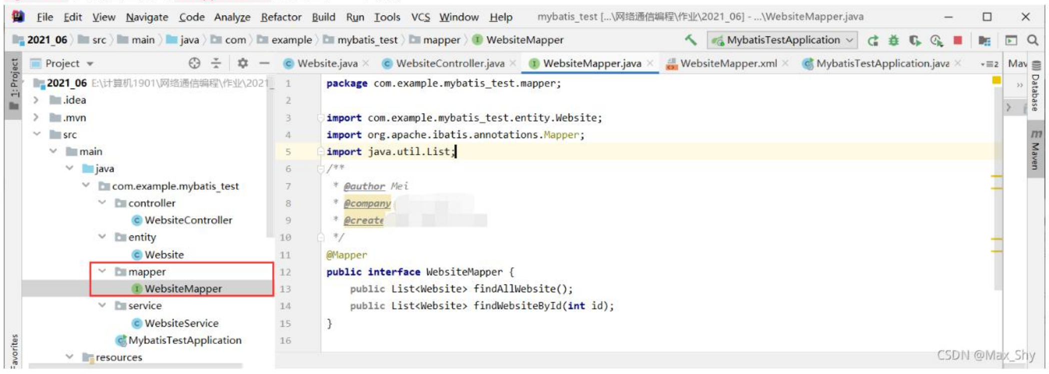
```
server.port=8080
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/runoob
#3306?????????
spring.datasource.username=root
spring.datasource.password=[redacted]
mybatis.mapper-locations=classpath:mapper/*Mapper.xml
#?useUnicode=true&characterEncoding=UTF-8&serverTimezone=UTC
```

mapper in ‘com.example’ and mapper in ‘resource.mapper’：

. . .

2. 创建 `mapper` 映射层：用于对数据库进行数据持久化操作，他的方法语句是直接针对数据库操作的，主要实现一些增删改查操作，在 `mybatis` 中方法主要与`*Mapper.xml` 内相互一一映射。

2.创建`mapper映射层`：用于对数据库进行数据持久化操作，他的方法语句是直接针对数据库操作的，主要实现一些增删改查操作，在`mybatis`中方法主要与`*Mapper.xml`内相互——映射。



```java
package com.example.mybatis_test.mapper;

import com.example.mybatis_test.entity.Website;
import org.apache.ibatis.annotations.Mapper;
import java.util.List;
/**
 * @author Mei
 * @company [redacted]
 * @create [redacted]
 */
@Mapper
public interface WebsiteMapper {
    public List<Website> findAllWebsite();
    public List<Website> findWebsiteById(int id);
}
```

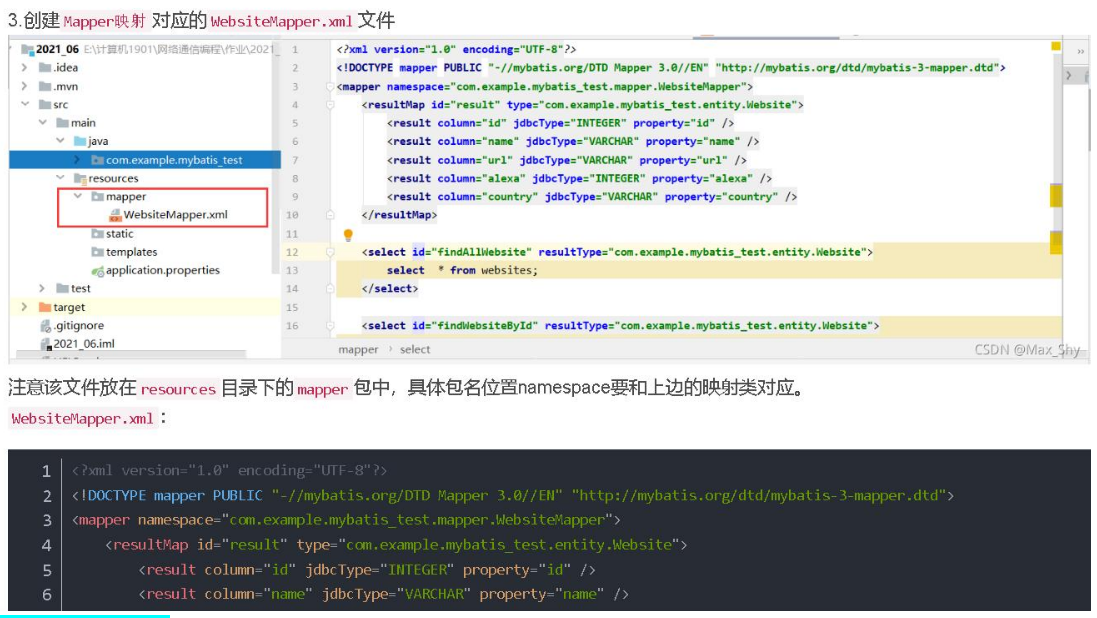
`WebsiteMapper` 接口：

3. 创建 Mapper 映射对应的 WebsiteMapper.xml 文件

…………

注意该文件放在 resources 目录下的 mapper 包中，具体包名位置 namespace 要和上边的映射类对应。

WebsiteMapper.xml：



3.创建 Mapper映射 对应的 WebsiteMapper.xml 文件

2021_06 E:\计算机1901\网络通信编程\作业\2021
.idea
.mvn
src
main
java
com.example.mybatis_test
resources
mapper
WebsiteMapper.xml
static
templates
application.properties
test
target
.gitignore
2021_06.iml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.mybatis_test.mapper.WebsiteMapper">
    <resultMap id="result" type="com.example.mybatis_test.entity.Website">
        <result column="id" jdbcType="INTEGER" property="id" />
        <result column="name" jdbcType="VARCHAR" property="name" />
        <result column="url" jdbcType="VARCHAR" property="url" />
        <result column="alexa" jdbcType="INTEGER" property="alexa" />
        <result column="country" jdbcType="VARCHAR" property="country" />
    </resultMap>

    <select id="findAllWebsite" resultType="com.example.mybatis_test.entity.Website">
        select  * from websites;
    </select>

    <select id="findWebsiteById" resultType="com.example.mybatis_test.entity.Website">
```

mapper › select

CSDN @Max_Shy

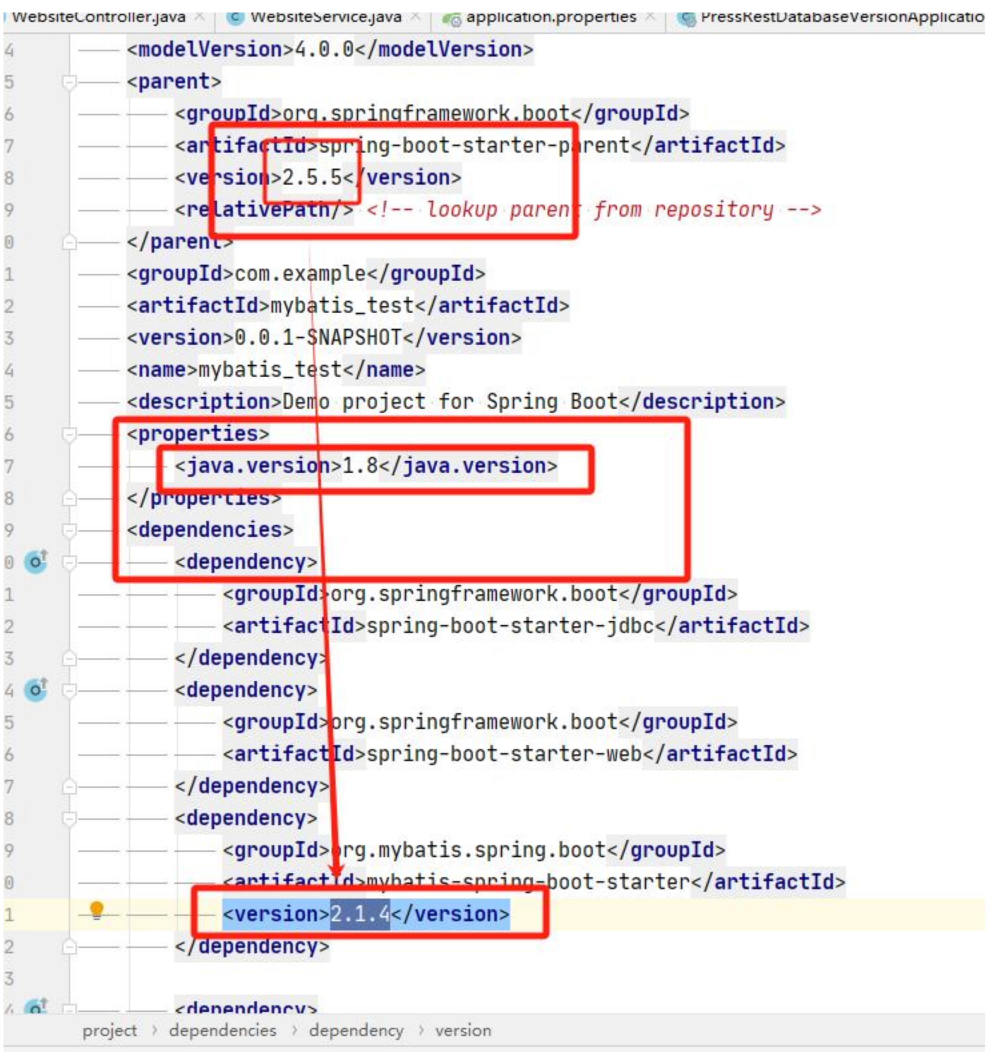
注意该文件放在 resources 目录下的 mapper 包中，具体包名位置namespace要和上边的映射类对应。

WebsiteMapper.xml：

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.mybatis_test.mapper.WebsiteMapper">
    <resultMap id="result" type="com.example.mybatis_test.entity.Website">
        <result column="id" jdbcType="INTEGER" property="id" />
        <result column="name" jdbcType="VARCHAR" property="name" />
```

Output:

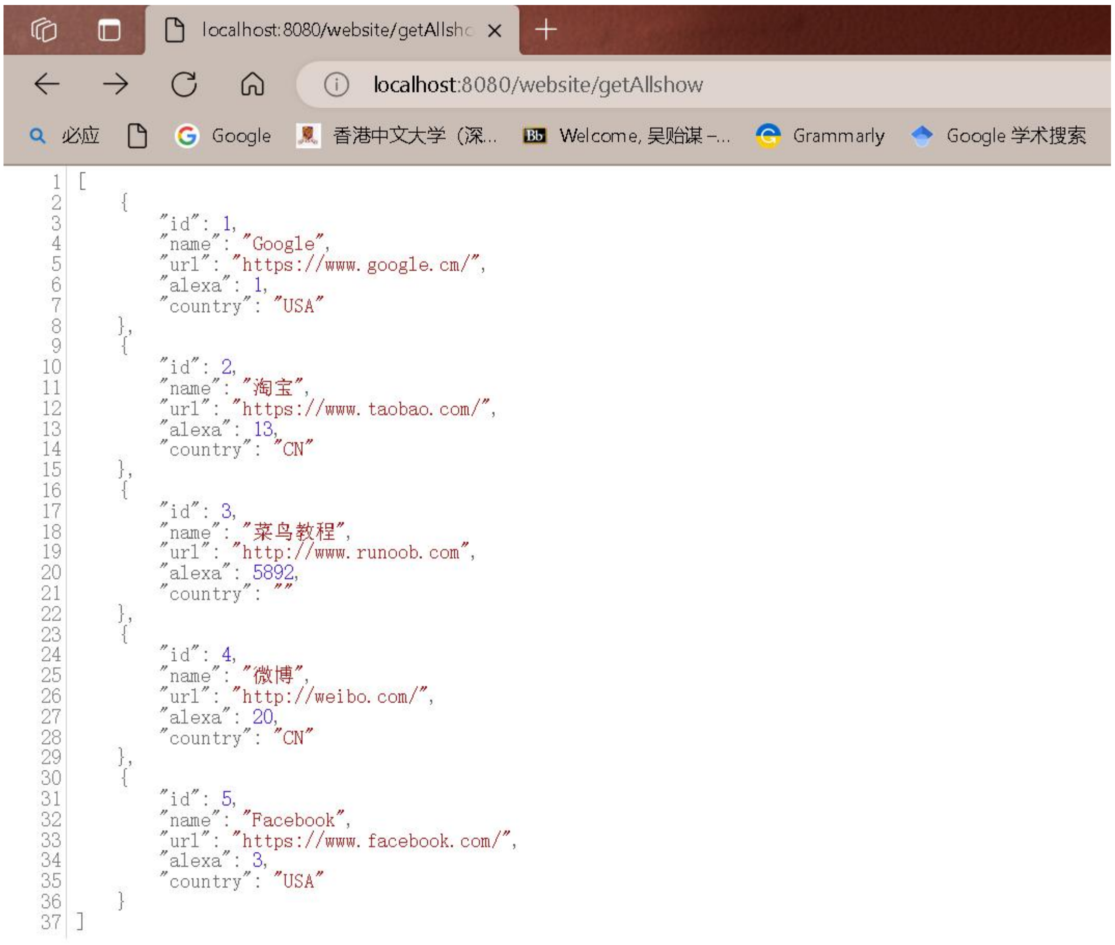
Mybytis, Java and springBoot’s Version!!! :


WebsiteController.java
WebsiteService.java
application.properties
PressRestDatabaseVersionApplicatio

```xml
<modelVersion>4.0.0</modelVersion>
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.5.5</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.example</groupId>
<artifactId>mybatis_test</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>mybatis_test</name>
<description>Demo project for Spring Boot</description>
<properties>
    <java.version>1.8</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-jdbc</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.mybatis.spring.boot</groupId>
        <artifactId>mybatis-spring-boot-starter</artifactId>
        <version>2.1.4</version>
    </dependency>

    <dependency>
```

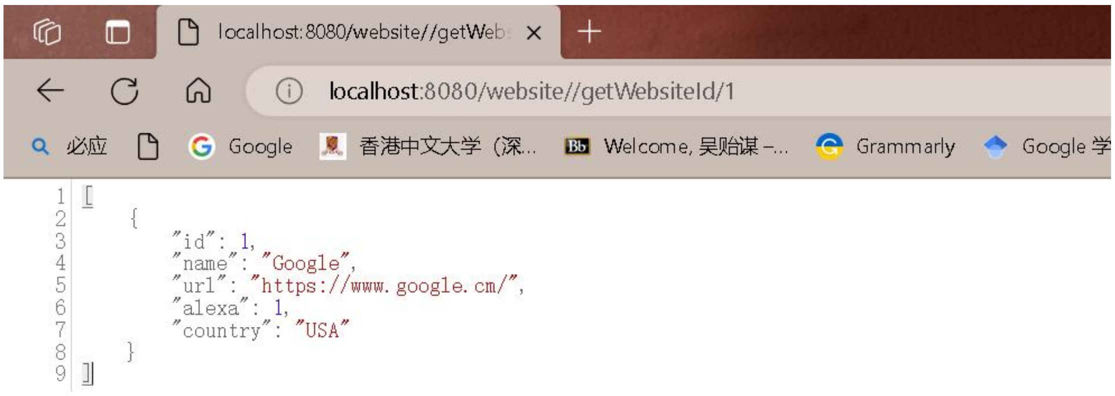
project › dependencies › dependency › version

localhost:8080/website/getAllshow



```
[
    {
        "id": 1,
        "name": "Google",
        "url": "https://www.google.cm/",
        "alexa": 1,
        "country": "USA"
    },
    {
        "id": 2,
        "name": "淘宝",
        "url": "https://www.taobao.com/",
        "alexa": 13,
        "country": "CN"
    },
    {
        "id": 3,
        "name": "菜鸟教程",
        "url": "http://www.runoob.com",
        "alexa": 5892,
        "country": ""
    },
    {
        "id": 4,
        "name": "微博",
        "url": "http://weibo.com/",
        "alexa": 20,
        "country": "CN"
    },
    {
        "id": 5,
        "name": "Facebook",
        "url": "https://www.facebook.com/",
        "alexa": 3,
        "country": "USA"
    }
]
```

localhost:8080/website//getWebsiteId/1


localhost:8080/website//getWeb
localhost:8080/website//getWebsiteId/1
必应
Google
香港中文大学（深...
Welcome, 吴贻谋 -...
Grammarly
Google 学

```
[
    {
        "id": 1,
        "name": "Google",
        "url": "https://www.google.cm/",
        "alexa": 1,
        "country": "USA"
    }
]
```