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
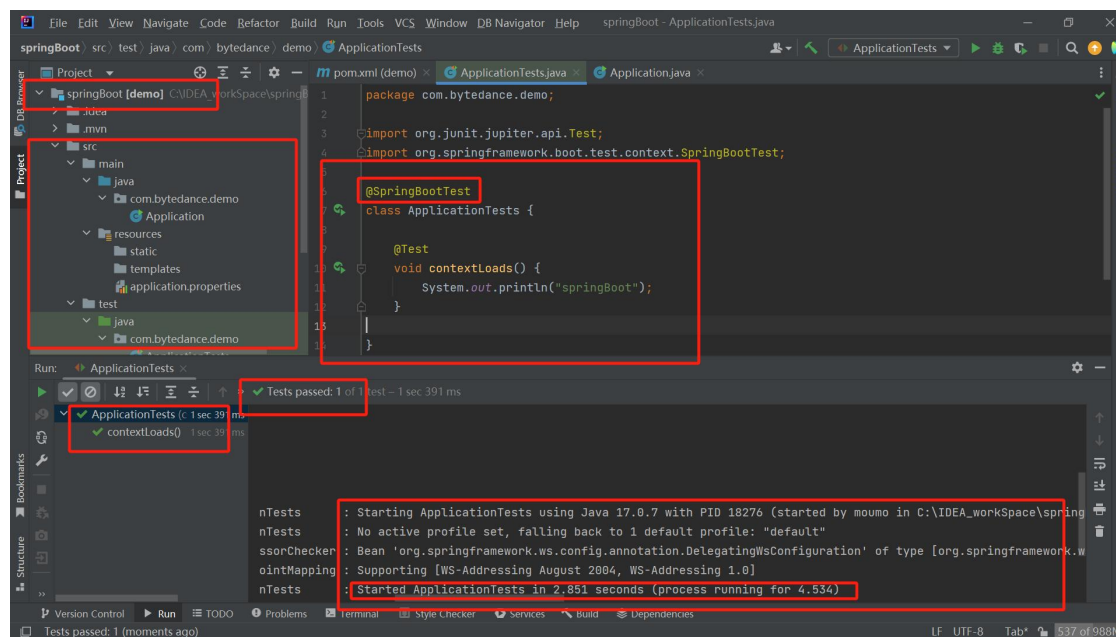
任务2

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。

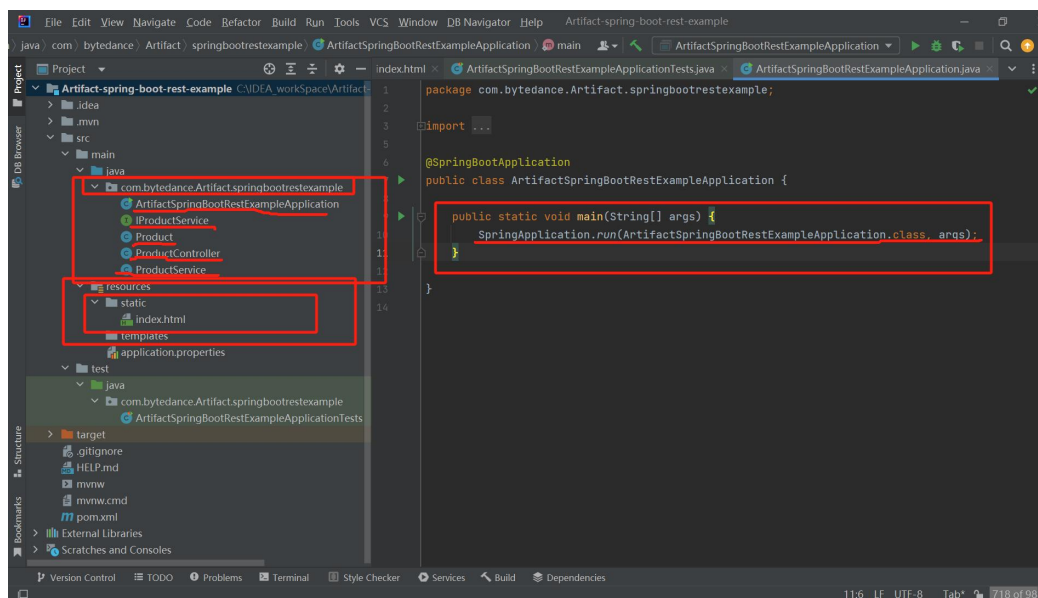1. RESTful 架构详解/...什么是 RESTful? .../... ：
https://www.runoob.com/w3cnote/restful-architecture.html

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':

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;
    }
}
```
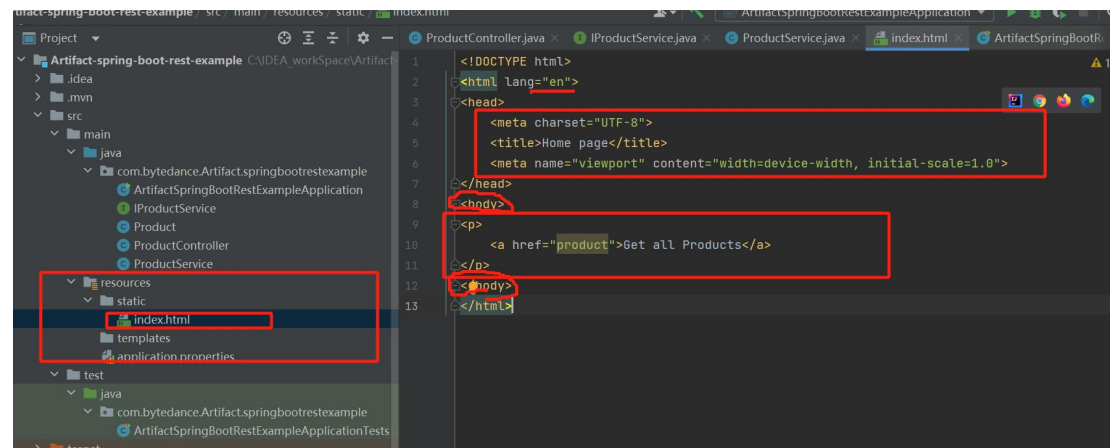
3.4. IProductService(Interface_Product_Service):

```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:

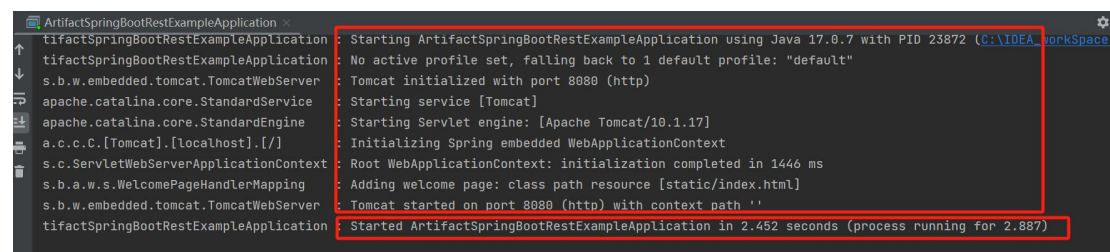<mark>(Using using an in-memory list instead of using a database like MySQL)</mark>
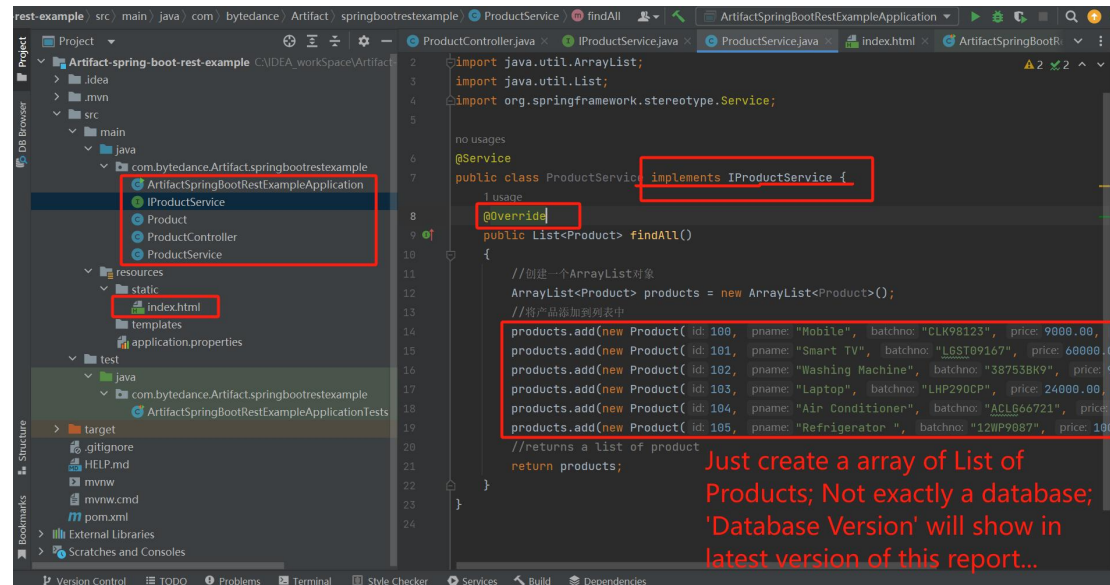
3.6. Index.html:

3.7. Output:

- Terminal

• 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:

```
1  [
2      {
3          "id": 100,
4          "pname": "Mobile",
5          "batchno": "CLK98123",
6          "price": 9000,
7          "noofproduct": 6
8      },
9      {
10         "id": 101,
11         "pname": "Smart TV",
12         "batchno": "LGST09167",
13         "price": 60000,
14         "noofproduct": 3
15     },
16     {
17         "id": 102,
18         "pname": "Washing Machine",
19         "batchno": "38753BK9",
20         "price": 9000,
21         "noofproduct": 7
22     },
23     {
24         "id": 103,
25         "pname": "Laptop",
26         "batchno": "LHP29OCP",
27         "price": 24000,
28         "noofproduct": 1
29     },
30     {
31         "id": 104,
32         "pname": "Air Conditioner",
33         "batchno": "ACLG66721",
34         "price": 30000,
35         "noofproduct": 5
36     },
37     {
38         "id": 105,
39         "pname": "Refrigerator",
40         "batchno": "12WP9087",
41         "price": 10000,
42         "noofproduct": 4
43     }
44 ]
```

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)

…
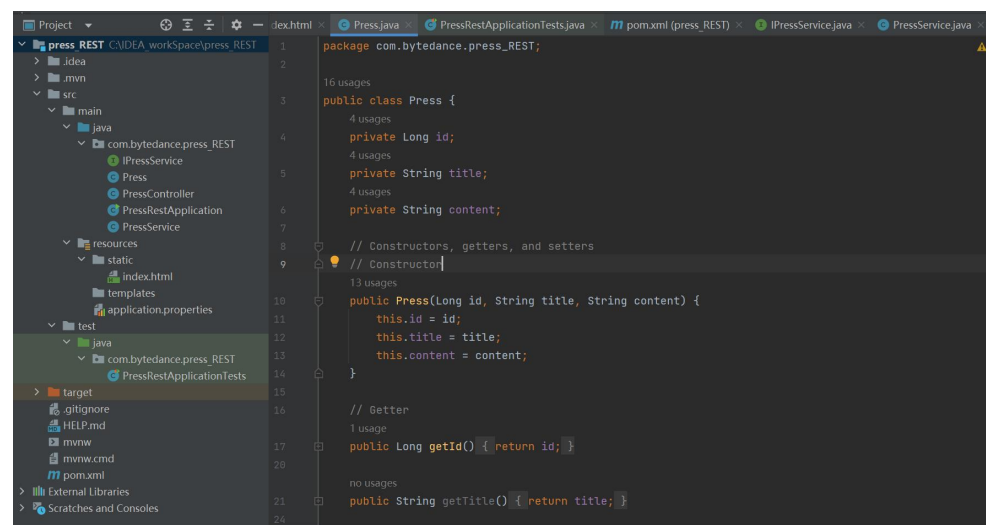
• PressService.java:

<mark>(Using using an in-memory list instead of using a database like MySQL)</mark>

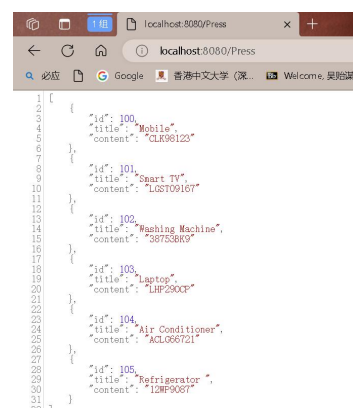• 打开浏览器并调用 URL <mark>http://localhost:8080/index.html</mark> <mark>('index.html' here is a folder under</mark>

<mark>"C:\IDEA_workSpace\press-REST_databaseVersion\src\main\resources\static\index.html" !</mark>)。
它**显示了** 获取所有产品的链接，如下图所示:

- <mark>In Google Chrome</mark>:
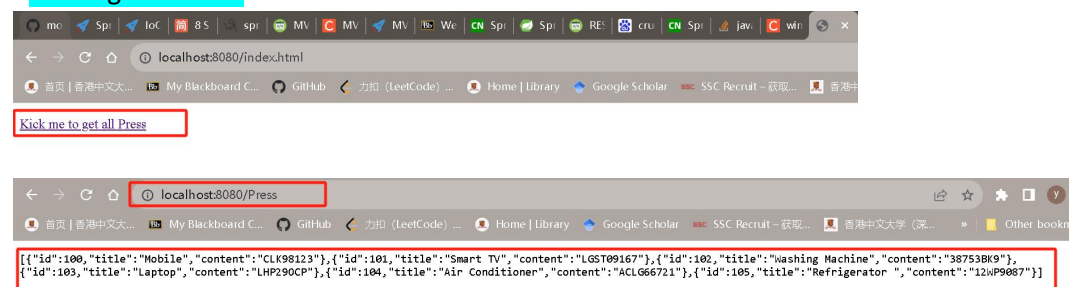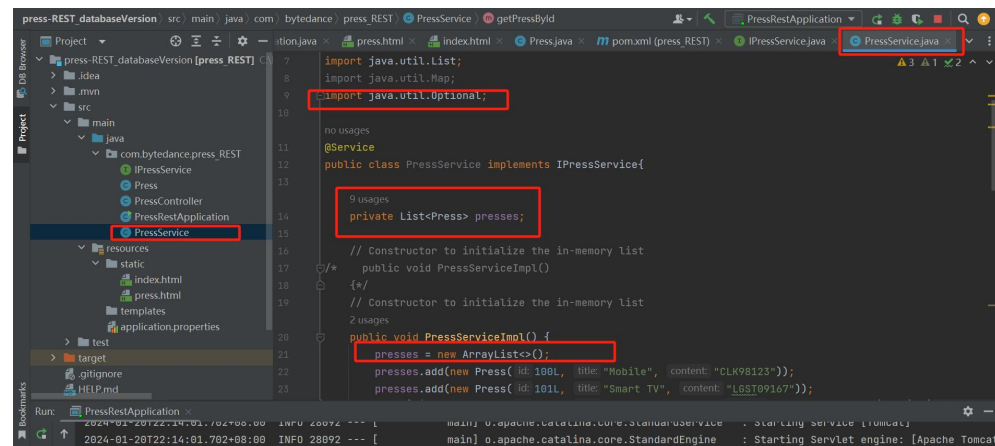

- <mark>In Microsoft Edge</mark>:

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:

5.2. PressController.java:

5.3. Output

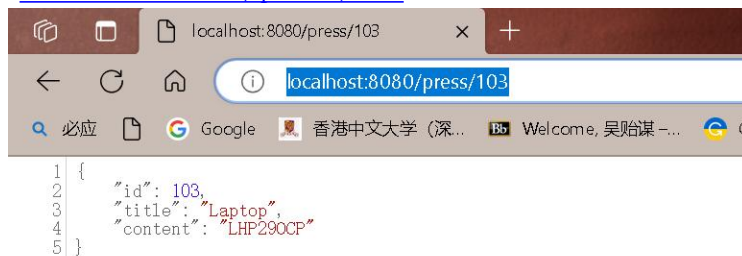"localhost:8080/press/105":


```
1  {
2      "id": 105,
3      "title": "Refrigerator ",
4      "content": "12WP9087"
5  }
```

"localhost:8080/press/103":


```
1  {
2      "id": 103,
3      "title": "Laptop",
4      "content": "LHP290CP"
5  }
```

"localhost:8080/press/100":


```
1  {
2      "id": 100,
3      "title": "Mobile",
4      "content": "CLK98123"
5  }
```

"localhost:8080/press/Press":


```
1  [
2      {
3          "id": 100,
4          "title": "Mobile",
5          "content": "CLK98123"
6      },
7      {
8          "id": 101,
9          "title": "Smart TV",
10         "content": "LGSTO9167"
11     },
12     {
13         "id": 102,
14         "title": "Washing Machine",
15         "content": "38753BK9"
16     },
17     {
18         "id": 103,
19         "title": "Laptop",
20         "content": "LHP29OCP"
21     },
22     {
23         "id": 104,
24         "title": "Air Conditioner",
25         "content": "ACLG66721"
26     },
27     {
28         "id": 105,
29         "title": "Refrigerator ",
30         "content": "12WP9087"
31     }
32  ]
```

"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

- <mark>缺点/未完成的地方</mark>：

只能实现<mark>in-memory list 而不是 using a database like MySQL</mark>。

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~