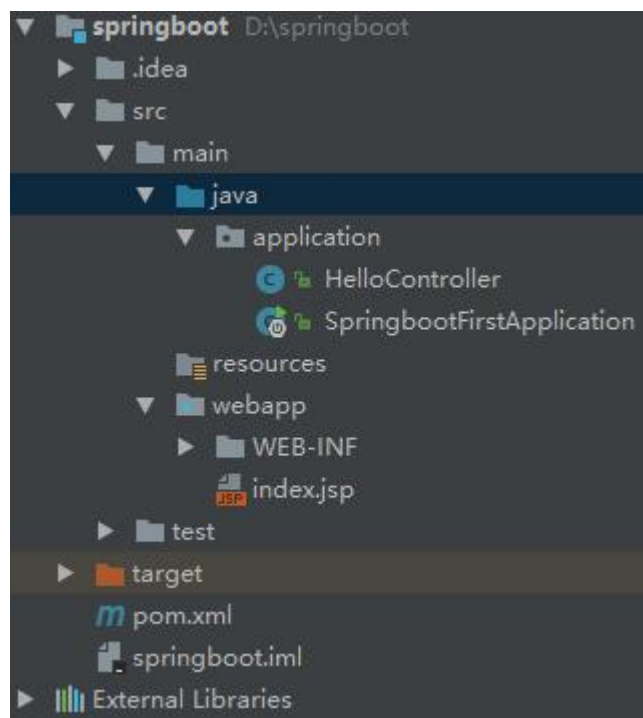


Spring-Boot 入门

第一个简单示例程序

```
/**
 * Created by 木子在南 on 2018/1/30.
 *记录下自己的学习之旅，对于我发的笔记有任何问题，欢迎联系我修正
 */
```

项目结构



Pom. XML

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>online.work</groupId>
  <artifactId>SpringDemo1</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
```

```

<name>SpringDemol Maven Webapp</name>
<url>http://maven.apache.org</url>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.5.2.RELEASE</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>

```

SpringBootFirstApplication.java

```
package online.work.application;
```

```

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

```

```

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;

import java.util.Arrays;

/**
 * Created by 84825 on 2018/1/28.
 */
@SpringBootApplication
public class SpringbootFirstApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringbootFirstApplication.class, args);
    }
    //查看 spring boot 注入了多少 bean
    // @Bean
    // public CommandLineRunner commandLineRunner(ApplicationContext ctx) {
    //     return args -> {
    //         //
    //         // System.out.println("Let's inspect the beans provided by Spring Boot:");
    //         //
    //         // String[] beanNames = ctx.getBeanDefinitionNames();
    //         // Arrays.sort(beanNames);
    //         // for (String beanName : beanNames) {
    //         //     System.out.println(beanName);
    //         // }
    //         //
    //         // };
    //     }
}

```

HelloController. java

```

package online.work.application;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

/**
 * Created by 84825 on 2018/1/28.
 */
@RestController
public class HelloController {
    @RequestMapping(value = "/", method = RequestMethod.GET)
}

```

```
public String index() {  
    return "my springboot";  
}  
}
```

Spring boot 项目属性配置(pom.xml 文件不变)

第一种，单属性注入

HelloController.java

```
package online.work.application;  
  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RestController;  
  
/**  
 * Created by 84825 on 2018/1/28.  
 */  
@RestController  
public class HelloController {  
    @Value("${cupsize}")  
    private String cupsize;  
    @Value("${age}")  
    private Integer age;  
    @RequestMapping(value = "/", method = RequestMethod.GET)  
    public String index() {  
        return cupsize+"-"+age;  
    }  
}
```

application.yml 文件

```
server:  
    port: 8081  
    #更改浏览器默认端口  
    context-path: /first  
    #相当于给路劲加了前缀,可以加可以不加,加上后访问地址为:  
    http://localhost:8081/first/
```

cupsize: B

age: 22

#注入属性的值，通过@value("\${cupsize}")注入其中

SpringbootFirstApplication. java

```
package online.work.application;
```

```
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;

import java.util.Arrays;

/**
 * Created by 84825 on 2018/1/28.
 */
@SpringBootApplication
public class SpringbootFirstApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringbootFirstApplication.class, args);
    }

    // @Bean
    // public CommandLineRunner commandLineRunner(ApplicationContext ctx) {
    //     return args -> {
    //         //
    //         // System.out.println("Let's inspect the beans provided by Spring Boot:");
    //         //
    //         // String[] beanNames = ctx.getBeanDefinitionNames();
    //         // Arrays.sort(beanNames);
    //         // for (String beanName : beanNames) {
    //         //     System.out.println(beanName);
    //         // }
    //         //
    //         // };
    //     }
}
```

第二种，多重属性注入(SpringbootFirstApplication.java 沿用上面的)(在配置中使用配置的属性)

application.xml

```
server:
port: 8081
#更改浏览器默认端口
context-path: /first
#相当于给路劲加了前缀,可以加可以不加,加上后访问地址为:
http://localhost:8081/first/
cupsize: B
age: 22
#注入属性的值,通过@value("${cupsize}")注入其中
content: "cupsize:${cupsize},age:${age}"
#在配置中使用配置的属性
```

HelloController.java

```
package online.work.application;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

/**
 * Created by 84825 on 2018/1/28.
 */
@RestController
public class HelloController {
    @Value("${cupsize}")
    private String cupsize;
    @Value("${age}")
    private Integer age;
    @Value("${content}")
    private String content;
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String index() {
        return content;
    }
}
```

第三种，对象属性的配置注入

application.yml

```
server:
port: 8081
#更改浏览器默认端口
context-path: /first
#相当于给路径加了前缀,可以加可以不加,加上后访问地址为:
http://localhost:8081/first/
girl:
cupsize: B
age: 22
```

Girl.java

```
package online.work.application;

import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.stereotype.Component;

/**
 * Created by 84825 on 2018/1/28.
 */
@Component
@ConfigurationProperties(prefix = "girl")//寻找前缀为 girl 的配置,然后为对象相应属性注入值
public class Girl {
    private String cupsize;
    private Integer age;

    public String getCupsize() {
        return cupsize;
    }

    public void setCupsize(String cupsize) {
        this.cupsize = cupsize;
    }

    public Integer getAge() {
        return age;
    }

    public void setAge(Integer age) {
```

```
this.age = age;
}
}
```

HelloController.java

```
package online.work.application;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
```

```
/**
```

```
 * Created by 84825 on 2018/1/28.
```

```
*/
```

```
@RestController
```

```
public class HelloController {
```

```
    @Autowired
```

```
    private Girl girl;
```

```
    @RequestMapping(value = "/", method = RequestMethod.GET)
```

```
    public String index() {
```

```
        return girl.getCupsize()+"-"+girl.getAge();
```

```
    }
```

```
}
```

第四种：多种环境下的配置文件切换

application.yml

```
spring:
```

```
profiles:
```

```
active: prod
```

```
#匹配配置含有 dev 的配置文件，可以自由更改
```

application-dev.yml

```
#开发环境
```

```
server:
```

```
port: 8081
```

```
#更改浏览器默认端口
```

```
context-path: /first
```

```
#相当于给路劲加了前缀,可以加可以不加,加上后访问地址为:
```

```
http://localhost:8081/first/
```

```
girl:
```


cupsize: G

age: 22

application-prod.yml

#生产环境

server:

port: 8081

#更改浏览器默认端口

context-path: /first

#相当于给路劲加了前缀,可以加可以不加,加上后访问地址为:

http://localhost:8081/first/

girl:

cupsize: F

age: 22

Ps:其他和上面一样,沿用即可

部分 Controller 控制器注解解析

@RestController

相当于@Controller+@ResponseBody 返回一个 json 字符串

@PathVariable

获取 URL 中的数据

```
package online.work.application;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
```

```
/**
```

```
* Created by 84825 on 2018/1/28.
```

```
*/
```

```
@RestController
```

```
public class HelloController {
```

```
//请求地址: http://localhost:8081/first/44
@RequestMapping(value =("/{id}", method = RequestMethod.GET)
public String index(@PathVariable("id") Integer id) {
    return "id:" + id;
}
}
```

@RequestParam

获取请求参数的值

```
package online.work.application;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.*;

/**
 * Created by 84825 on 2018/1/28.
 */
@RestController
public class HelloController {
    //请求地址: http://localhost:8081/first/findId?id=404
    @RequestMapping(value = "/findId", method = RequestMethod.GET)
    public String index(@RequestParam("id") Integer myId) {
        return "myId:" + myId;
    }
}
```

增加默认值

```
package online.work.application;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.*;

/**
 * Created by 84825 on 2018/1/28.
 */
@RestController
public class HelloController {
    //请求地址: http://localhost:8081/first/findId?id=404
    @RequestMapping(value = "/findId", method = RequestMethod.GET)
```

```

public String index(@RequestParam(value = "id", required = false, defaultValue =
"55") Integer myId) {
return "myId:" + myId;
}
}

```

@GetMapping

组合注解

数据库操作 CURD

pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>online.work</groupId>
<artifactId>SpringDemol</artifactId>
<packaging>war</packaging>
<version>1.0-SNAPSHOT</version>
<name>SpringDemol Maven Webapp</name>
<url>http://maven.apache.org</url>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.5.2.RELEASE</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>

```

```

</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
<!-- jpa 包 -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<!-- 数据库连接驱动 -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>

```

application.yml

```

spring:
profiles:
active: prod
#匹配配置含有 dev 的配置文件，可以自由更改
datasource:
driver-class-name: com.mysql.jdbc.Driver
url: jdbc:mysql://127.0.0.1:3306/dbgirl
username: root
password: root
jpa:
hibernate:
ddl-auto: update
show-sql: true

```

application-dev.xml

#开发环境

server:

port: 8081

#更改浏览器默认端口

context-path: /first

#相当于给路劲加了前缀,可以加可以不加,加上后访问地址为:

http://localhost:8081/first/

girl:

cupsize: G

age: 22

application-prod.yml

#生产环境

server:

port: 8081

#更改浏览器默认端口

context-path: /first

#相当于给路劲加了前缀,可以加可以不加,加上后访问地址为:

http://localhost:8081/first/

girl:

cupsize: F

age: 22

GirlRepository.java

```
package online.work.application;
```

```
import org.omg.PortableInterceptor.INACTIVE;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import java.util.List;
```

```
/**
```

```
 * Created by 84825 on 2018/1/30.
```

```
*/
```

```
public interface GirlRepository extends JpaRepository<GirlModel,Long> {
```

```
/**
```

```
 * 拓展根据年龄查询
```

```
 * @param age
```

```
 * @return
```

```
*/
```

```
public List<GirlModel> findByAge(Integer age);
```

```
}
GirlModel.java
package online.work.application;

import javax.persistence.*;

/**
 * Created by 84825 on 2018/1/29.
 */
@Entity
@Table(name = "girl_model")
public class GirlModel {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private Integer age;

    public GirlModel() {
    }
    public GirlModel(Long id) {
        this.id=id;
    }
    public GirlModel(String name, Integer age) {
        this.name = name;
        this.age = age;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

```

public Integer getAge() {
    return age;
}

public void setAge(Integer age) {
    this.age = age;
}

}

```

GirlController. java

```

package online.work.application;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

/**
 * Created by 84825 on 2018/1/30.
 */
@RestController
public class GirlController {
    @Autowired
    private GirlRepository girlRepository;

    /**
     * 查询所有
     * 访问地址: http://localhost:8081/first/findAllGirls
     * @return
     */
    @GetMapping(value = "findAllGirls")
    public List<GirlModel> girlList() {
        System.out.println(girlRepository.findAll());
        return girlRepository.findAll();
    }

    /**
     * 增加一个女生信息
     * 访问地址 http://localhost:8081/first/addGirl
     * @param name
     * @param age
     * @return
     */
}

```

```

@PostMapping(value = "addGirl")
public GirlModel addGirlModel(@RequestParam("name") String
name ,@RequestParam("age") Integer age){
    GirlModel girlModel=new GirlModel();
    girlModel.setAge(age);
    girlModel.setName(name);
    return girlRepository.save(girlModel);
}

/**
 * 查询一个女生
 * 查询地址: http://localhost:8081/first/findOneGirlModel/2
 * @param id
 * @return
 */
@GetMapping(value = "findOneGirlModel/{id}")
public GirlModel findOneGirlModel(@PathVariable Long id){
    return girlRepository.findOne(id);
}

/**
 * 更新数据库里面的一条数据
 * 更新地址: http://localhost:8081/first/girlModelUpdate/3
 * @param id
 * @param name
 * @param age
 * @return
 */
@PutMapping(value = "girlModelUpdate/{id}")
public GirlModel girlModelUpdate(@PathVariable Long id ,@RequestParam String
name,@RequestParam Integer age){
    GirlModel girlModel=new GirlModel();
    girlModel.setId(id);
    girlModel.setAge(age);
    girlModel.setName(name);
    return girlRepository.save(girlModel);
}

/**
 * 删除一条数据
 * 删除访问地址: http://localhost:8081/first/deleteGirlModel/3
 * @param id
 */
@DeleteMapping(value = "deleteGirlModel/{id}")

```



```

public void deleteGirlModel(@PathVariable Long id) {
    girlRepository.delete(id);
}

/**
 * 根据年龄查询列表
 * 访问地址: http://localhost:8081/first/findByAge/3
 * @param age
 * @return
 */
@GetMapping(value = "findByAge/{age}")
public List<GirlModel> findByAge(@PathVariable("age") Integer age) {
    return girlRepository.findByAge(age);
}
}

```

SpringbootFirstApplication. java

```

package online.work.application;

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;

import java.util.Arrays;

/**
 * Created by 84825 on 2018/1/28.
 */
@SpringBootApplication
public class SpringbootFirstApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringbootFirstApplication.class, args);
    }

    // @Bean
    // public CommandLineRunner commandLineRunner(ApplicationContext ctx) {
    //     return args -> {
    //         //
    //         // System.out.println("Let's inspect the beans provided by Spring Boot:");
    //         //
    //         String[] beanNames = ctx.getBeanDefinitionNames();
    //         Arrays.sort(beanNames);
    //     }
    // }
}

```

```
// for (String beanName : beanNames) {  
// System.out.println(beanName);  
// }  
//  
// };  
// }  
}
```

事务管理

GirlModelServer. java

```
package online.work.application;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import javax.transaction.Transactional;  
  
/**  
 * Created by 84825 on 2018/1/30.  
 */  
@Service  
public class GirlModelServer {  
    @Autowired  
    private GirlRepository girlRepository;  
    @Transactional  
    public void insertTwoGirlModel() {  
        GirlModel girlModel=new GirlModel();  
  
        girlModel.setAge(33);  
        girlModel.setName("李白");  
        girlRepository.save(girlModel);  
        GirlModel girlModel2=new GirlModel();  
  
        girlModel2.setAge(44);  
        girlModel2.setName("杜甫");  
        girlRepository.save(girlModel2);  
    }  
}
```

GirlController. java

```
package online.work.application;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

/**
 * Created by 84825 on 2018/1/30.
 */
@RestController
public class GirlController {
    @Autowired
    private GirlRepository girlRepository;
    @Autowired
    private GirlModelServer girlModelServer;

    /**
     * 保存两条数据，使用事物管理
     * 访问地址: http://localhost:8081/first/savaTwoGirlModel
     */
    @PostMapping(value = "savaTwoGirlModel")
    public void savaTwoGirlModel() {
        girlModelServer.insertTwoGirlModel();
    }
}
```