REliable, INtelligent & Scalable Systems

互联网应用开发技术

Web Application Development

第13课
WEB后端框架-SSH框架整合

Episode Thirteen

SSH

陈昊鹏
chen-hp@sjtu.edu.cn

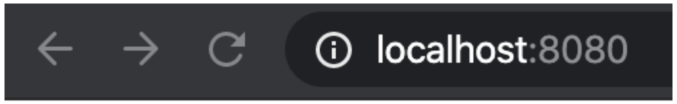
Web Application Development

SSH – Annotation

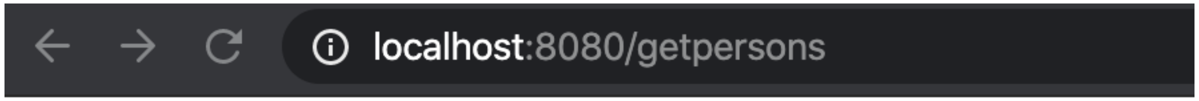
REliable, INtelligent & Scalable Systems

- SshApplication.java

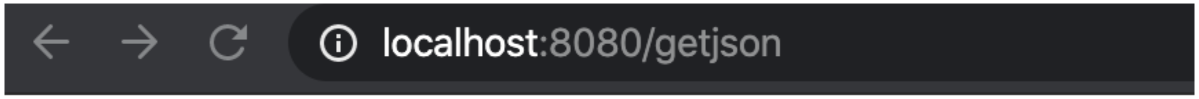
```java
@SpringBootApplication
public class SshApplication {
    public static void main(String[] args) {
        SpringApplication.run(SshApplication.class, args);
    }
}
```


localhost:8080

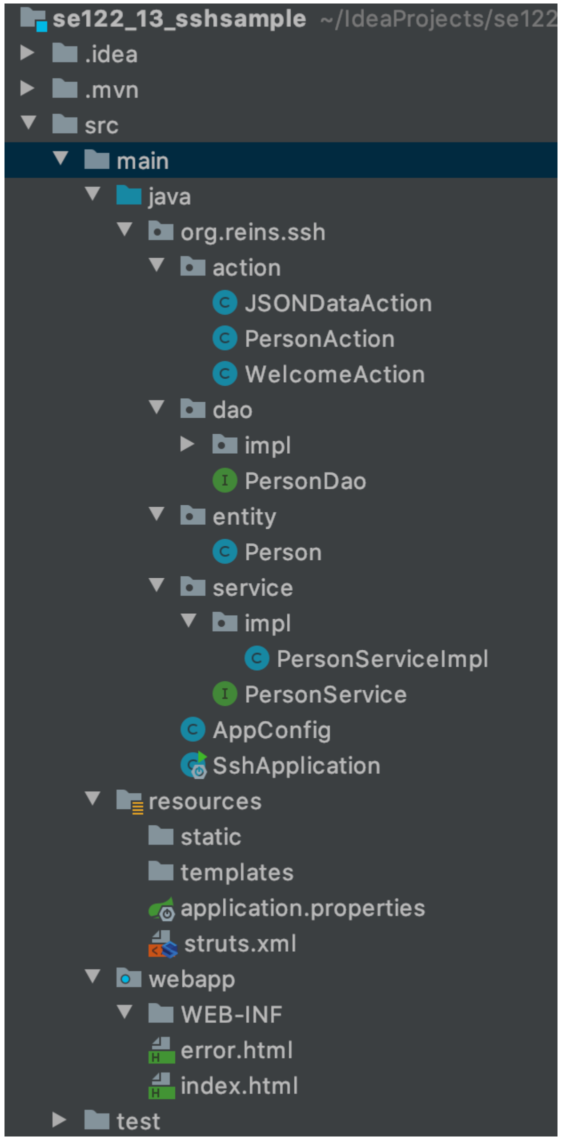
Get Persons Get JSON


localhost:8080/getpersons

```
[["54","Cao","Cao"],["50","Bei","Liu"],["27","Quan","Sun"]]
```


localhost:8080/getjson

```
[{"age":54,"firstname":"Cao","id":1,"lastname":"Cao"},
{"age":50,"firstname":"Bei","id":2,"lastname":"Liu"},
{"age":27,"firstname":"Quan","id":3,"lastname":"Sun"}]
```



```
se122_13_sshsample ~/IdeaProjects/se122
▶ .idea
▶ .mvn
▼ src
  ▼ main
    ▼ java
      ▼ org.reins.ssh
        ▼ action
            JSONDataAction
            PersonAction
            WelcomeAction
        ▼ dao
          ▶ impl
            PersonDao
        ▼ entity
            Person
        ▼ service
          ▼ impl
              PersonServiceImpl
            PersonService
          AppConfig
          SshApplication
    ▼ resources
        static
        templates
        application.properties
        struts.xml
    ▼ webapp
      ▼ WEB-INF
          error.html
          index.html
  ▶ test
```

SSH – Annotation

- struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">
<!-- 配置内容搜了些教程...最关键的操作是最后一步扫描action所在的web包 -->
<struts>
  <!-- 表示从cn包下开始扫描 -->
  <constant name="struts.convention.package.locators.basePackage" value="org.reins.ssh"></constant>
  <!-- 表示从cn包下开始加载注解 -->
  <constant name="struts.convention.package.locators" value="org.reins.ssh"></constant>
  <!-- 指定由spring负责action对象的创建 -->
  <constant name="struts.objectFactory" value="spring" />
  <!--
    SpringBoot集成struts2这个必须要配置
    表示不把SpringBoot内置Tomcat的类加载器排除在外
     默认是true 将类加载器排除了所以就算上面配置
    如果这个没有配置就算运行不报错但是Action还是访问不了
  -->
  <constant name="struts.convention.exclude.parentClassLoader" value="false"></constant>
  <!-- 这一点也是，好多网上教程，居然是配置的default -->
  <constant name="struts.convention.default.parent.package" value="convention-default" />
  <!-- 这一点网上好多教程都没有，既然是注解，这个扫描包的都不加，搞毛线啊 -->
  <constant name="struts.convention.action.packages" value="org.reins.ssh.action" />
</struts>
```

SSH – Annotation

REliable, INtelligent & Scalable Systems

- AppConfig.java

```java
@Configuration
@ComponentScan
public class AppConfig {
    @Bean
    public PersonDao personDao() {
        return new PersonDaoImpl();
    }
    @Bean
    public PersonService personService() {
        return new PersonServiceImpl();
    }
    @Bean
    public SessionFactory sessionFactory(){
        org.hibernate.cfg.Configuration config = new org.hibernate.cfg.Configuration();
        config.setProperty("hibernate.connection.driver_class", "com.mysql.cj.jdbc.Driver");
        config.setProperty("hibernate.connection.url", "jdbc:mysql://localhost:3306/ormsample");
        config.setProperty("hibernate.connection.username", "root");
        config.setProperty("hibernate.connection.password", "reins2011!");
        config.setProperty("hibernate.dialect", "org.hibernate.dialect.MySQL8Dialect");
        config.addAnnotatedClass(Person.class);
        return config.buildSessionFactory();
    }
```

SSH – Annotation

- AppConfig.java

```
@Configuration
@ComponentScan
public class AppConfig {

   @Bean
    public StrutsPrepareAndExecuteFilter strutsPrepareAndExecuteFilter(){
        return new StrutsPrepareAndExecuteFilter();
    }

}
```

SSH – Annotation

- action/WelcomeAction.java

```
@Controller
public class WelcomeAction  extends ActionSupport {

    @Action(value = "/", results = {
            @Result(name = "success", location = "/index.html")
    })
    public String welcome() {
        System.out.println("Welcome!");
        return SUCCESS;
    }
}
```

SSH – Annotation

- action/PersonAction.java

```
@Controller
public class PersonAction extends ActionSupport {
   @Autowired
   private PersonService personService;
   private List<Person> list;
   @Action(value = “getpersons”, results = {
         @Result(name = “error”, location = “/error.html”)
   })
   public String getPersons() {
      System.out.println(“This is an action”);
      try {
         list = personService.queryAll();
         Iterator<Person> it = list.iterator();
         ArrayList<JSONArray> personsJson = new ArrayList<JSONArray>();
         while (it.hasNext()) {
            Person person = (Person) it.next();
            ArrayList<String> arrayList = new ArrayList<String>();
            arrayList.add(Integer.toString(person.getAge()));
            arrayList.add(person.getFirstname());
            arrayList.add(person.getLastname());
            personsJson.add((JSONArray) JSONArray.toJSON(arrayList));
         }
```

SSH – Annotation

- action/PersonAction.java

```
            String personsString = JSON.toJSONString(personsJson, SerializerFeature.BrowserCompatible);
            HttpServletResponse response = ServletActionContext.getResponse();
            response.reset();
            response.setContentType("text/html;charset=utf-8");

            PrintWriter out = response.getWriter();
            out.write(personsString);
            out.flush();
            out.close();

            return null;
        } catch (Exception e) {
            return ERROR;
        }
    }

    public List<Person> getList() {
        return list;
    }
}
```

SSH – Annotation

REliable, INtelligent & Scalable Systems

- action/JSONDataAction.java

```
@Controller
@ParentPackage("json-default")
public class JSONDataAction extends ActionSupport {
  @Autowired
  private PersonService personService;
  private List<Person> list;
  @Action(value = "getjson", results = {
      @Result(type="json", name="jsonlist", params={"root", "list"}),
      @Result(name = "error", location = "/error.html")
  })
  public String execute() throws Exception {
    list = personService.queryAll();
    Iterator<Person> it = list.iterator();
    while (it.hasNext()) {
      Person person = (Person) it.next();
    }
    return "jsonlist";
  }
  public List<Person> getList() {
    return list;
  }
}
```

SSH – Annotation

REliable, INtelligent & Scalable Systems

- service/PersonService.java

```
public interface PersonService {
    List<Person> queryAll();
}
```

- service/impl/PersonServiceImpl.java

```
@Service
public class PersonServiceImpl implements PersonService {
    @Autowired
    PersonDao personDao;

    public List<Person> queryAll() {
        List<Person> list=personDao.queryAll();
        return list;
    }
}
```

SSH – Annotation

REliable, INtelligent & Scalable Systems

- dao/PersonDao.java

```
public interface PersonDao {
    List<Person> queryAll();
}
```

- dao/impl/PersonDaoImpl.java

```
@Transactional
@Repository
@Service
public class PersonDaoImpl implements PersonDao {
    @Autowired
    private SessionFactory sessionFactory;

    public List<Person> queryAll() {
        List<Person> list;
        list = sessionFactory.openSession().createQuery("from Person").list();
        for (Person person: list) {
            System.out.println(person.getAge() + "  " + person.getFirstname() + " " + person.getLastname());
        }
        return list;
    }
}
```

SSH – Annotation

REliable, INtelligent & Scalable Systems

- entity/Entity.java

```
@Entity
@Table(name = "PERSONS")
public class Person implements Serializable {
    private Long id;
    private int age;
    private String firstname;
    private String lastname;

    public Person() {
        // this form used by Hibernate
    }

    @Id
    @GeneratedValue(generator = "increment")
    @GenericGenerator(name = "increment", strategy = "increment")
    public Long getId() {
        return id;
    }
    private void setId(Long id) {
        this.id = id;
    }
```

SSH – Annotation

- entity/Entity.java

```
public int getAge() {
    return age;
}
private void setAge(int age) {
    this.age = age;
}

public String getFirstname() {
    return firstname;
}
private void setFirstname(String firstname) {
    this.firstname = firstname;
}

public String getLastname() {
    return lastname;
}
private void setLastname(String lastname) {
    this.lastname = lastname;
}
}
```

SSH – Annotation

- index.html

```
<HTML>
<BODY>

<a href="/getpersons">Get Persons</a>
<a href="/getjson">Get JSON</a>

</BODY>
</HTML>
```

SSH – Annotation

- error.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Sorry</title>
</head>
<body>
<h1>Sorry, something seems wrong!</h1>
</body>
</html>
```

Summary

- Spring
  - IoC: Design by Contract (Interface), Decoupling Interface and Impl
- Struts
  - MVC: Decoupling Model and View with Controller
- Hibernate
  - ORM: Decoupling Relational DBMS and Objects
- Points
  - Decoupling
  - Laying

SSH – Annotation

- pom.xml

```xml
<!-- https://mvnrepository.com/artifact/org.apache.struts/struts2-core -->
<dependency>
    <groupId>org.apache.struts</groupId>
    <artifactId>struts2-core</artifactId>
    <version>2.5.22</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.struts/struts2-convention-plugin -->
<dependency>
    <groupId>org.apache.struts</groupId>
    <artifactId>struts2-convention-plugin</artifactId>
    <version>2.5.22</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.struts/struts2-json-plugin -->
<dependency>
    <groupId>org.apache.struts</groupId>
    <artifactId>struts2-json-plugin</artifactId>
    <version>2.5.22</version>
</dependency>
```

References

- Configuring a DataSource Programmatically in Spring Boot
  - https://www.baeldung.com/spring-boot-configure-data-source-programmatic
- Hibernate Programmatic Configuration Example
  - https://www.codejava.net/frameworks/hibernate/hibernate-programmatic-configuration-example
- SpringBoot集成struts2,mybatis
  - https://blog.csdn.net/qq1105515654/article/details/73744349
- struts2基于action的注解操作详细总结笔记
  - https://blog.csdn.net/qq_41893274/article/details/85372124?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task
- struts2注解json 配置文件json
  - https://www.cnblogs.com/youhun/p/3586109.html
- Struts2的使用注解配置Action（零配置）
  - https://www.cnblogs.com/qlqwjy/p/8530035.html

- *Web开发技术*
- *Web Application Development*

Thank You!