

写了个简单的 MVC 框架

作者: [Hby5909](#)

原文链接: <https://ld246.com/article/1518975446902>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

前言

前些日子写的轻量级MVC框架已经写得差不多了，基于 Servlet，同时我也封装了JDBC来提供对数据的访问。当初写这个框架是为了了解servlet是如何工作的，同时也作为自己的一个小项目，思路参考了[架构探险：从零开始写Java Web框架](#)》。虽然写这个框架只是作为学习之用，但后面还是会继续完和优化。附上[Github](#)地址

快速使用

Maven

```
<dependency>
  <groupId>group.jbee</groupId>
  <artifactId>jbee-framework</artifactId>
  <version>1.0.1</version>
</dependency>
```

还未上传Maven仓库，找不到依赖可clone本项目，执行mvn install打包到本地仓库。也可以直接下载jar包，[点此下载](#)。

其他依赖

MySQL驱动

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.33</version>
</dependency>
```

配置文件

在resource目录下新建jbee.properties配置文件，配置项如下

```
/**JDBC 驱动*/
jbee.framework.jdbc.driver=com.mysql.jdbc.Driver
/**JDBC URL*/
jbee.framework.jdbc.url=jdbc:mysql://localhost:3306/jbee
/**JDBC 用户名*/
jbee.framework.jdbc.username=root
/**JDBC 密码*/
jbee.framework.jdbc.password=123456
/**JDBC 连接池最大数量*/
jbee.framework.jdbc.max=5
/**JDBC 连接池最小数量*/
jbee.framework.jdbc.min=2
/**应用基础包名*/
jbee.framework.app.base_package=group.jbee.jbeedemo
/**JSP路径*/
jbee.framework.app.jsp_path=/view/
```

```
/**静态资源路径*/  
jbee.framework.app.static_path=/static/
```

当然，还有web.xml配置

```
<servlet-mapping>  
  <servlet-name>default</servlet-name>  
  <url-pattern>/*</url-pattern>  
</servlet-mapping>  
<servlet>  
  <servlet-name>default</servlet-name>  
  <servlet-class>group.jbee.framework.DispatcherServlet</servlet-class>  
  <load-on-startup>0</load-on-startup>  
</servlet>
```

以上就是所有必要的配置，下面看如何使用

IOC

IOC是Spring的核心，Jbee也实现了一个简单的IOC容器。

使用注解 @Inject修饰需要注入的类。

示例代码：

```
@Controller  
public class HelloAction {  
  
    @Inject  
    private UserService userService;  
  
    .....  
}  
  
@Service  
public class UserService {  
  
    @Inject  
    UserMapper userMapper;  
  
    public List<User> getUserInfo(User user){  
        return userMapper.getUserInfo(user);  
    }  
    public void createUser(User user){  
        userMapper.createUser(user);  
    }  
}
```

Controller

与Spring MVC差不多，使用 @Controller注解来修饰控制器类，使用 @Action注解来匹配URL

示例代码：

```

@Controller
public class HelloAction {

    @Inject
    private UserService userService;

    //返回json数据
    @Action("POST:/Action1")
    public Data action1(Param param){
        Map<String,Object> params = param.getMap();
        String name = (String) params.get("name");
        User user= new User();
        user.setName(name);
        Data data=new Data(userService.getUserInfo(user));
        return data;
    }
    //返回jsp页面
    @Action("GET:/Action2")
    public View action1(){
        View view = new View("hello.jsp");
        //使用EL表达式${data}取值
        view.setModel("data","HelloJbee!");
        return view;
    }
}

```

DAO

持久层与Mybatis类似，只需编写DAO接口以及SQL语句，目前只支持静态SQL，用占位符#{ }来设置数。接口用注解 @Repository声明，方法用 @Select、@Insert、@Delete、@Update来声明。

示例代码：

```

@Repository
public interface UserMapper {

    @Select("select * from users where name=#{name}")
    List<User> getUserInfo(User user);

    @Insert(value = "insert into users(name,password) values(#{name},#{password})")
    void createUser(User user);
}

```

最后

将项目打war包，放到Tomcat等servlet容器里就可以跑起来了。

寒假比较摸，AOP还没写-.-，会尽快写完吧，毕竟大三下又要很忙了。

框架实现原理后面我也会分享在博客里。