

Java Web 开发框架简介

专业班级： 网络工程 2 班

姓 名： 李毓琪

指导教师： 李辉

信息科学技术学院

摘 要

框架（Framework）是一个系统全部或者部分的可复用设计，是支持软件设计复用和实现复用的技术。随着 Internet 的发展，Web 应用变得日益复杂，框架技术已成为当前 Web 应用开发的一种主流技术。框架是一个提供了可重用的公共结构半成品，它为构建新的应用程序提供了极大的便利。

“框架”这个词最早出现在建筑领域，指在建造房屋前期构建的建筑骨架，对于应用程序来说，“框架”就是应用程序的骨架，开发者可以在这个骨架上搭建符合自己需求的应用系统；框架凝结着前人的经验和智慧，使用这些框架就等于站在了巨人的肩膀上。

Rickard Oberg（Web Work 的开发者和 JBoss 的创始人之一）说过：“框架的强大之处不是源自它能让你做什么，而是它不能让你做什么。”Rickard 强调框架另一个层面的含义：框架能使混乱的内容变得结构化。如果没有框架，一千个人将写出一千种 Servlet + JavaBean + JSP 的代码，而框架保证了程序结构风格的统一。

从企业的角度来说，框架也降低了人员培训和软件维护的成本。框架在结构统一和创造力之间维持着一个合适的平衡。

主流框架：Java Web 开发框架：Spring 框架，Spring MVC 框架，Struts 2 框架，MyBatis 框架，Hibernate 框架等。

ABSTRACT

Framework is the reusable design of all or part of a system, and is a technology that supports software design reuse and implementation. With the development of the Internet, Web applications have become increasingly complex, and framework technology has become a mainstream technology in current Web application development. A framework is a semi-finished product that provides a reusable public structure that greatly facilitates building new applications.

The word "framework" first appeared in the field of construction, referring to the building skeleton built in the early stage of building a house. For an application, a "framework" is the skeleton of the application, and developers can build an application system that meets their own needs on this skeleton; The frame condenses the experience and wisdom of the predecessors, and using these frames is equivalent to standing on the shoulders of giants.

Rickard Oberg (Web Work developer and one of the founders of JBoss) said: "The power of a framework is not what it allows you to do, but what it doesn't allow you to do." Rickard emphasizes another aspect of the framework Meaning: Frames make cluttered content structured. If there is no framework, a thousand people will write a thousand kinds of Servlet + JavaBean + JSP code, and the framework ensures the unity of the program structure style.

From an enterprise perspective, the framework also reduces the cost of personnel training and software maintenance. The framework maintains a suitable balance between structural unity and creativity.

Mainstream Java Web development frameworks: Spring framework, Spring MVC framework, Struts 2 framework, MyBatis framework, Hibernate framework, etc.

目 录

第一章 SPRING 框架	5
1.1 框架简介	5
1.2 IOC(INVERSION OF CONTROL).....	5
1.3 AOP 面向切面编程	5
1.4 SPRING 框架的优点	5
第二章 SPRING MVC 框架	6
2.1 框架简介	6
2.2 SPRING MVC 框架的优点	6
第三章 STRUTS 2 框架	7
3.1 框架简介	7
3.2 STRUTS 中的 MVC 结构.....	7
3.3 STRUTS 2 与 SPRING MVC 比较.....	7
第四章 MYBATIS 框架.....	8
4.1 框架简介	8
4.2 MYBATIS 框架的特点	8
第五章 HIBERNATE 框架	9
5.1 框架简介	9
5.2 HIBERNATE 与 MYBATIS 的比较.....	9
第六章 JAVASERVER FACES 框架	10
6.1 框架简介	10
6.2 JAVASERVER FACES 的 MVC 实现.....	10
6.3 JAVASERVER FACES 的优点	10
6.4 JAVASERVER FACES 的生命周期	10

第一章 Spring 框架

1.1 框架简介

Spring 框架是一个开源框架，是轻量级的 IoC 和 AOP 的容器框架，主要是用于管理 JavaBean 生命周期的轻量级容器，他不仅可以单独使用，也可以与 Struts 框架、iBatis 框架等 Java EE 技术结合使用。Spring 框架是由于软件开发的复杂性而创建的，它是一个开源框架。Spring 框架的用途不仅限于服务器端的开发，从简单性、可测试性和松耦合性角度而言，绝大部分 Java 应用都可以从 Spring 框架中受益。

1.2 IoC(Inversion of Control)

控制反转，对象创建的责任反转，BeanFacotory 是 spring 中 IoC 容器的核心接口，负责在应用程序中实例化、定位、配置对象以及建立这些对象之间的依赖关系。XmlBeanFacotory 实现了 BeanFactory 接口，通过获取 xml 配置文件数据组成应用对象和对象之间的依赖关系。Spring 框架中有三种注入方式，一种是 Set 注入，一种是接口注入，另一种是构造方法注入。

1.3 AOP 面向切面编程

AOP 是纵向的编程，所以如果业务 1 和业务 2 都需要一个共同的任务，最好编写一次代码，让两个业务一起使用这段代码，而不是在每个业务中添加相同的代码。在 Spring 框架中实现面向方面的编程有两种方式：一种是动态代理，另一种是 CGLIB。动态代理必须提供接口，而 CGLIB 实现具有继承性。

1.4 Spring 框架的优点

- 1) 控制反转将对象的创建交给了 spring, 简化了开发, 降低了代码之间的耦合性和侵入性。
- 2) 方便对程序进行声明式事物管理, 我们只需通过配置就可以完成对事物管理。
- 3) 方便集成各种优秀的框架, spring 不排斥各种优秀框架, 其内部提供了对各种优秀框架如(struts2, hibernate, mybatis, quartz, jpa)等的直接支。
- 4) 方便对程序进行测试, spring 对于 Junit4 的支持, 可通过注解方便测试程序。

- 5) 降低了 JavaEE API 的使用难度, JDBC, Javamail, 远程调用等, spring 对它们进行了封装, 使这些 API 的使用难度大大降低。

第二章 Spring MVC 框架

2.1 框架简介

Spring MVC 框架是 Spring Frame Work 集成到 Spring Web Flow 中的后续产品, 它实现了结构清晰的 MVC Model2。Spring 框架提供了用于构建 Web 应用程序的全功能 MVC 模块, 具有高度可配置性, 并支持多视图技术。定制和开发也非常灵活。

此外, Spring 框架集成 Spring MVC 框架是一种无缝集成, 是一种在互联网应用程序开发中越来越流行的高性能架构模式。使用 Spring 框架进行 Web 开发时, 可以选择 Spring MVC 框架或集成其他 MVC 开发框架, 如 Struts 1 (现在基本不常用)、Struts 2 (老项目中常用) 等。

2.2 Spring MVC 框架的优点

- 1) Spring MVC 框架提供了一整套完善的组件。

与以往的框架只提供接口, 让用户通过基于接口编写实现类来实现功能不同, Spring MVC 提供了一整套可以直接使用的完善的组件。这与之前的框架只提供设计图, 你必须根据设计图来构建工具不同, Spring MVC 提供了一个现成的工具箱, 其中包含你可能使用的各种工具。而且使用这些工具也很简单, 通过一些简单的注解就搞定了, 不需要编写繁琐冗长的代码。

- 2) Spring MVC 是以强大的 Spring 容器为基础的框架。

如果曾经使用过 Spring 容器, 那一定知道 Spring 容器的对象托管功能有多么的好用。它不需要你自己去创建和释放对象, Spring 容器能够帮你全搞定, 类似的, Spring MVC 工具箱, 可以放在一个 Spring 容器中, 不用担心管理里面的各种工具类。而这省心省力的优点, 是其他框架都没有的。

- 3) 框架的配置简单又不失灵活性。

Spring MVC 框架的常用配置项就已经涵盖了项目中 80% 的配置需求, 简单的项目甚至可以做到零配置, 拿过来就用。

- 4) 代码的可重用性很高。

翻译成大白话就是, 一个成熟的在线稳定的 Spring MVC 项目可以作为开发其他新项目的基础, 几乎不需要修改。这样能在开发新项目时省很多力气。

- 5) 可扩展性好。

如果 Spring MVC 框架提供的工具不能 100% 满足你的需求, 你需要定制开发, Spring MVC 框架也提供了相应的升级工具接口。

第三章 Struts 2 框架

3.1 框架简介

Struts 2 框架以 Web Work 的优秀设计思想为核心, 吸收了 Struts 框架的部分优点, 提供了一个基于 MVC 设计模式的更紧凑的 Web 应用程序框架, 本质上是一个 Servlet。在 MVC 设计模式中, Struts 2 框架作为控制器 (Controller) 来建立模型与视图的数据交互。

Struts 2 框架是 Struts 的下一代产品, 它是基于 Struts 1 和 Web Work 技术的综合创新。它采用拦截器的机制来处理用户的请求, 这也可以理解为 Web Work 的更新产品, 因为它允许业务逻辑控制器与 Servlet API 完全解耦。

3.2 Struts 中的 MVC 结构

在 Struts 中, 已经由一个名为 ActionServlet 的 Servlet 充当 控制器 (Controller) 的角色, 根据描述模型、视图、控制器对应关系的 struts-config.xml 的配置文件, 转发视图 (View) 的请求, 组装响应数据模型 (Model)。在 MVC 的 模型 (Model) 部分, 经常划分为两个主要子系统 (系统的内部数据状态与改变数据状态的逻辑动作), 这两个概念子系统分别具体对应 Struts 里的 ActionForm 与 Action 两个需要继承实现超类。在这里, Struts 可以与各种标准的数据访问技术结合在一起, 包括 Enterprise Java Beans (EJB), JDBC 与 JNDI。在 Struts 的视图 (View) 端, 除了使用标准的 JavaServer Pages (JSP) 以外, 还提供了大量的标签库使用, 同时也可以与其他表现层组件技术 (产品) 进行整合, 比如 Velocity Templates, XSLT 等。通过应用 Struts 的框架, 最终用户可以把大部分的关注点放在自己的业务逻辑 (Action) 与 映射关系的配置文件 (struts-config.xml) 中。

3.3 Struts 2 与 Spring MVC 比较

两者都有一个共同点。即都是数据 Java Web 层的开发框架, 都是 MVC 模式的经典产品, 都实现了页面分离控制的功能, 但是两者还是有区别的。在开发过程中, 人们更喜欢使用 Spring MVC 而不是 Struts。由于 Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色, 这种分离让它们更容易进行定制, 在扩展和灵活性上更胜一筹。

Struts 的优势在于静态注入, 插件机制和拦截器链, 但是 Struts 存在漏洞, 经常被作为攻击点攻击。相比更安全、更简单的 Spring MVC, 开发者已经逐渐放弃了它。

第四章 MyBatis 框架

4.1 框架简介

MyBatis 框架是一个优秀的数据库持久层框架, 可以建立实体类和 SQL 语句的映射关系, 是一种半自动化的 ORM 实现。MyBatis 的封装比 Hibernate 框架低, 性能更好, 而且容易学习, 所以被广泛使用。

MyBatis 框架最初是 iBatis, 它是 Apache 的一个开源项目。2010 年, 该项目被 Apache 软件基金会迁移到 Google Code, 并更名为“MyBatis”。2013 年 11 月它迁移到 Github。“iBatis”一词来源于“internet”和“abatis”的组合, 它是一个基于 Java 的持久层框架, 包括 SQL 映射和 DAO (Data Access Objects, 数据访问对象)。

4.2 MyBatis 框架的特点

与其他对象关系映射框架不同, MyBatis 没有将 Java 对象与数据库表关联起来, 而是将 Java 方法与 SQL 语句关联。MyBatis 允许用户充分利用数据库的各种功能, 例如存储过程、视图、各种复杂的查询以及某数据库的专有特性。如果要对遗留数据库、不规范的数据库进行操作, 或者要完全控制 SQL 的执行, MyBatis 是一个不错的选择。

与 JDBC 相比, MyBatis 简化了相关代码: SQL 语句在一行代码中就能执行。MyBatis 提供了一个映射引擎, 声明式的把 SQL 语句执行结果与对象树映射起来。通过使用一种内建的类 XML 表达式语言, 或者使用 Apache Velocity 集成的插件, SQL 语句可以被动态的生成。

MyBatis 与 Spring Framework 和 Google Guice 集成, 这使开发者免于依赖性问题。

MyBatis 支持声明式数据缓存 (declarative data caching)。当一条 SQL 语句被标记为“可缓存”后, 首次执行它时从数据库获取的所有数据会被存储在一段高速缓存中, 今后执行这条语句时就会从高速缓存中读取结果, 而不是再次命中数据库。MyBatis 提供了基于 Java HashMap 的默认缓存实现, 以及用于与 OSCache、Ehcache、Hazelcast 和 Memcached 连接的默认连接器。MyBatis 还提供 API 供其他缓存实现使用。

第五章 Hibernate 框架

5.1 框架简介

Hibernate 框架不仅是一个优秀的持久化框架，还是一个开源的对象-关系映射框架。它对 JDBC 进行了轻量级的对象封装，建立 POJO 与数据库表的映射关系，形成一个全自动的 ORM 框架。

Hibernate 框架可以自动生成并自动执行 SQL 语句，让 Java 程序员可以随心所欲地使用对象编程思维来操作数据库。Hibernate 框架还可以用于任何使用 JDBC 的情况。可以在 Java 的客户端程序使用；也可以应用在 Servlet/JSP 的 Web 应用中；最具革命意义的是，Hibernate 框架可以用于应用程序 EJB 的 Java EE 架构中。它取代了 CMP 来完成繁重的数据持久化工作。

Hibernate 框架已经成为当前主流的数据库持久化框架并被广泛使用。

5.2 Hibernate 与 MyBatis 的比较

- 1) Hibernate 与 Mybatis 都是流行的持久层开发框架，一句话概括：MyBatis 简单易上手。
- 2) hibernate 成熟，市场推广率高。
- 3) MyBatis 可以进行更为细致的 SQL 优化，可以减少查询字段。
- 4) MyBatis 容易掌握，而 Hibernate 门槛较高。
- 5) 更重要的是，mybatis 提供了对应各种用途、功能的插件，而 hibernate 在这一方面是远远比不上 mybatis 的。
- 6) Hibernate 的 DAO 层开发比 MyBatis 简单，Mybatis 需要维护 SQL 和结果映射。
- 7) Hibernate 对对象的维护和缓存要比 MyBatis 好，对增删改查的对象的维护要方便。
- 8) Hibernate 数据库移植性很好，MyBatis 的数据库移植性不好，不同的数据库需要写不同 SQL。
- 9) Hibernate 有更好的二级缓存机制，可以使用第三方缓存。MyBatis 本身提供的缓存机制不佳。
- 10) 但是 hibernat 缺点很明确，如果涉及到多张关联表的调用时：多表关联等比较复杂，使用的成本并不低；效率比较低，在大型项目中很少会使用到它，因为 sql 都是自动生成的，不太好进行人工的优化。

第六章 JavaServer Faces 框架

6.1 框架简介

JavaServer Faces 还引起了广大 Java/Web 开发人员的兴趣。“企业开发人员”和 Web 设计人员将发现 JSF 开发可以简单到只需将用户界面 (UI) 组件拖放到页面上,而“系统开发人员”将发现丰富而强健的 JSF API 为他们提供了无与伦比的功能和编程灵活性。JSF 还通过将良好构建的模型-视图-控制器 (MVC) 设计模式集成到它的体系结构中,确保了应用程序具有更高的可维护性。最后,由于 JSF 是通过 Java Community Process (JCP) 开发的一种 Java 标准,因此开发工具供应商完全能够为 JavaServer Faces 提供易于使用的、高效的可视化开发环境。

6.2 JavaServer Faces 的 MVC 实现

JSF 的主要优势之一就是它既是 Java Web 用户界面标准又是严格遵循模型-视图-控制器 (MVC) 设计模式的框架。用户界面代码 (视图) 与应用程序数据和逻辑 (模型) 的清晰分离使 JSF 应用程序更易于管理。为了准备提供页面对应用程序数据访问的 JSF 上下文和防止对页面未授权或不正确的访问,所有与应用程序的用户交互均由一个前端 “Faces” servlet (控制器) 来处理。

6.3 JavaServer Faces 的优点

它提供干净,清晰的 Web 应用程序的行为和呈现之间的分离。开发者可以分别编写业务逻辑和用户界面。JSF API 直接分层在 Servlet API 之上。这使得可以使用不同的应用程序使用案例,例如使用不同的表现层技术,直接从组件类创建自己的定制组件。在 JSF2.0 中包含 Facelets 技术,为其提供了巨大的优势。Facelets 现在是构建基于 JavaServer Faces 的 Web 应用程序的首选表现层技术。

6.4 JavaServer Faces 的生命周期

Faces Controller servlet 充当用户和 JSF 应用程序之间的纽带。它在明确限定的 JSF 生命周期 (规定了用户请求之间的整个事件流) 的范围内工作。例如,一收到访问 JSF 应用程序的初始 Web 请求,Faces Controller servlet 便通过首先准备 JSF 上下文 (存放所有应用程序数据的一个 Java 对象) 来处理请求。然后控制器把用户指引到所请求的页面。该页面通常使用简单的表达式

语言来处理来自 JSF 上下文的应用程序数据。一收到后续请求，控制器就更新所有模型数据（假设输入了新数据）。JSF 开发人员可以通过编程的方式在应用程序运行期间随时访问整个 JSF 生命周期，从而可以随时对应用程序的行为进行高度控制。

结 论

框架技术已成为当前 Web 应用开发的一种主流技术。框架对于当今的开发工作做不可或缺,提供了可重用的公共结构半成品,它为构建新的应用程序带来了极大的便利。学习框架的目的就是提高项目的编写效率,通过学习以及实验,可以有更多的时间去编写属于自己独特的东西,而不需要将大量的时间花费在底层代码的书写上。这是是一种很好的编程思想,既简化了编程,也保证了程序的正确性.所以,在项目中或者是团队开发中建立自己的框架就显得很有必要。