J2EE及其三大框架和MVC架构的介绍

1. **J2EE**

1.介绍

J2EE（Java 2 Platform, Enterprise Edition）是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。它是一套全然不同于传统应用开发的技术架构，包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。J2EE核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循J2EE架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，企业内部或外部难以互通的窘境。

2.结构

这种基于组件，具有平台无关性的J2EE结构使得J2EE 程序的编写十分简单，因为业务逻辑被封装成可复用的组件，并且J2EE 服务器以容器的形式为所有的组件类型提供后台服务。

容器和服务容器设置定制了J2EE服务器所提供的内在支持，包括安全，事务管理，JNDI(Java Naming and Directory Interface）寻址，远程连接等服务，以下列出最重要的几种服务：

（1）J2EE安全（Security）模型可以让你配置 web 组件或enterprise bean，这样只有被授权的用户才能访问系统资源. 每一客户属于一个特别的角色，而每个角色只允许激活特定的方法。你应在enterprise bean的布置描述中声明角色和可被激活的方法。由于这种声明性的方法，你不必编写加强安全性的规则。

（2）J2EE 事务管理（Transaction Management）模型让你指定组成一个事务中所有方法间的关系，这样一个事务中的所有方法被当成一个单一的单元. 当客户端激活一个enterprise bean中的方法，容器介入一管理事务。因有容器管理事务，在enterprise bean中不必对事务的边界进行编码。要求控制分布式事务的代码会非常复杂。你只需在布置描述文件中声明enterprise bean的事务属性，而不用编写并调试复杂的代码。容器将读此文件并为你处理此enterprise bean的事务。JNDI 寻址（JNDI Lookup）服务向企业内的多重名字和目录服务提供了一个统一的接口，这样应用程序组件可以访问名字和目录服务。

（3）J2EE远程连接（Remote Client Connectivity）模型管理客户端和enterprise bean间的低层交互。当一个enterprise bean创建后，一个客户端可以调用它的方法就象它和客户端位于同一虚拟机上一样。

（4）生存周期管理（Life Cycle Management）模型管理enterprise bean的创建和移除，一个enterprise bean在其生存周期中将会历经几种状态。容器创建enterprise bean，并在可用实例池与活动状态中移动他，而最终将其从容器中移除。即使可以调用enterprise bean的create及remove方法，容器也将会在后台执行这些任务。

（5）数据库连接池（Database Connection Pooling）模型是一个有价值的资源。获取数据库连接是一项耗时的工作，而且连接数非常有限。容器通过管理连接池来缓和这些问题。enterprise bean可从池中迅速获取连接。在bean释放连接之后可为其他bean使用。

3.J2EE的13种核心技术

（1）:JDBC(Java Database Connectivity)

JDBC API为访问不同数据库提供了统一的路径,像[ODBC](https://baike.baidu.com/item/ODBC" \t "https://baike.baidu.com/item/j2ee/_blank)一样,[JDBC](https://baike.baidu.com/item/JDBC" \t "https://baike.baidu.com/item/j2ee/_blank)开发者屏蔽了一些细节问题,另外,JDBC对数据库的访问也具有平台无关性.

（2）:JNDI(Java Name and Directory Interface)

JNDI API 被用于执行名字和目录服务.它提供了一致的模型来存取和操作企业级的资源DNS和LDAP,本地文件系统,或应用服务器中的对象.

（3）:EJB(Enterprise JavaBean)

J2EE技术之所以赢得广泛重视的原因之一就是EJB.它提供了一个框架来开发和实施分布式商务逻辑,由此很显著的简化了具有可伸缩性和高度复杂的企业级应用程序的开发.EJB规范定义了EJB组件在何时如何与它们的容器进行交互作用.容器负责提供公用的服务,例如目录服务,事务管理,安全性,资源缓冲池以及容错性.但这里值得注意的是,EJB并不是实现J2EE的唯一路径.正是由于J2EE的开放性,使得所有的厂商能够以一种和EJB平行的方式来达到同样的目地.

（4）:RMI(Remote Method Invoke)

远程方法请求,RMI协议调用远程对象上的方法.它使用了序列化的方式在客户端和服务器之间传递数据.RMI是一种被EJB使用的更底层的协议.

（5）:Java IDL/CORBA(通用对象请求代理架构是软件构建的一个标准 )

在Java IDL的支持下,开发人员可以将Java和CORBA集成在一起.他们可以创建Java对象并使之可在CORBA ORB中展开,或者他们还可以创建Java类并和其它ORB一起展开的CORBA对象客户.后一种方法提供了另外一种途径,通过它Java可以被用于将你的新的应用程序和旧的系统集合在一起.

（6）:JSP

JSP页面由[HTML](https://baike.baidu.com/item/HTML" \t "https://baike.baidu.com/item/j2ee/_blank)（[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/j2ee/_blank)下的一个应用）代码和嵌入其中的Java代码组成.服务器在

页面被客户端所请求以后对这些Java代码进行处理,然后将生成的HTML页面返回给客户端浏览器.

（7）:Java Servlet

Servlet 是一种小型的Java程序,它扩展了web服务器的功能.作为一种服务器的应用,当被请求时开始执行,这和CGI Perl脚本很相似.Servlet提供的功能大多和JSP类似,不过实现的方式不同.JSP通常是大多数的HTML代码中嵌入少量的Java代码,而servlet全部由java写成并且生成HTML.

（8）:XML

[XML](https://baike.baidu.com/item/XML" \t "https://baike.baidu.com/item/j2ee/_blank)（[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/j2ee/_blank)的子集）是一种可以用来定其它标记语言的语言.它被用来在不同的商务过程中共享数据.XML的发展和java是相互独立的,但是,它和java具有的相同目标是平台独立性.

（9）:JMS

MS是用于和面向对象消息的中间件相互通信的应用程序接口.它既支持点对点的域,又支持发布/订阅类型的域,并且提供了下列类型的支持:消息传递,事务型消息的传递,一致性消息和具有持久性的订阅者支持.JMS还提供了另一种方式来对新系统和旧后台系统相互集成.

（10）:JTA

[JTA](https://baike.baidu.com/item/JTA" \t "https://baike.baidu.com/item/j2ee/_blank)定义了一种标准API,应用程序由此可以访问各种事务监控.

（11）:JTS

JTS是CORBA OTS事务监控的基本实现.JTS规定了事务管理的实现方法.该事务管理器是在高层支持java Transaction API规范,并且在较低层次实现OMG OTS specification 和Java印象.JTS事务管理器为应用程序服务器,资源管理器,独立的应用以及同学资源管理器提供了事务服务.

（12）:JavaMail

JavaMail是用于存取邮件服务器的API,它提供了一套邮件服务器的抽象类.不仅支持SMTP服务器,也支持IMAP服务器.

（13）:JAF(JavaBeans Activation Framework)

[JavaMail](https://baike.baidu.com/item/JavaMail" \t "https://baike.baidu.com/item/j2ee/_blank)利用JAF来处理MIME编码的邮件附件.MIME的字节流可以被转换成java对象,大多数应用都可以不需要直接使用JAF。

**二、J2EE对应的三个框架**

SSH 在J2EE项目中表示了3种框架，即 Spring + Struts +Hibernate。 SSH不是一个框架，而是多个框架的集成，是目前较流行的一种Web应用程序开源集成框架，用于构建灵活、易于扩展的多层Web应用程序。Struts对Model，View和Controller都提供了对应的组件。Spring是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架，它由Rod Johnson创建。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。 Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，可以应用在任何使用JDBC的场合，可以在Servlet/JSP的Web应用中使用，也可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。

1. Spring
2. 简介

◆目的：解决企业应用开发的复杂性

◆功能：使用基本的JavaBean代替EJB，并提供了更多的企业应用功能

◆范围：任何Java应用

简单来说，Spring是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架。

1. 轻量

从大小与开销两方面而言Spring都是轻量的。完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布。并且Spring所需的处理开销也是微不足道的。此外，Spring是非侵入式的：典型地，Spring应用中的对象不依赖于Spring的特定类。

1. 控制反转

Spring通过一种称作控制反转（IoC）的技术促进了松耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为IoC与JNDI相反——不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

1. 面向切面

Spring提供了[面向切面编程](http://baike.baidu.com/view/1865230.htm" \t "http://blog.sina.com.cn/s/_blank)的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和事务（transaction）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

1. 容器

Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个bean如何被创建——基于一个可配置原型（prototype），你的bean可以创建一个单独的实例或者每次需要时都生成一个新的实例——以及它们是如何相互关联的。然而，Spring不应该被混同于传统的重量级的EJB容器，它们经常是庞大与笨重的，难以使用。　　（6）框架

Spring可以将简单的组件配置、组合成为复杂的应用。在Spring中，应用对象被声明式地组合，典型地是在一个XML文件里。Spring也提供了很多基础功能（事务管理、持久化框架集成等等），将应用逻辑的开发留给了你。所有Spring的这些特征使你能够编写更干净、更可管理、并且更易于测试的代码。它们也为Spring中的各种模块提供了基础支持。

1. 为什么要使用Spring

Spring的容器、轻量、面向切面、促进其它框架、面向切口这些特性使得开发人员使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。

1. Struts

Struts 是采用 Java Servlet/JSP 技术，开发 Web 应用程序的开放源码的框架，采用 Struts 能开发出基于 MVC(Model-View-Controller)设计模式的应用构架。Struts 有一组相互协作的类、Servlet 以及 Jsp Taglib 组成。

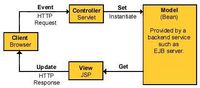
Struts 有如下的主要功能：

①包含一个 controller servlet，能将用户的请求发送到相应的 Action 对象。

②JSP 自由 tag 库，并且在 controller servlet 中提供关联支持，帮助开发员创建交互式表单应用。

③提供了一系列实用对象：XML 处理、通过 Java reflection APIs 自动处理 Java Beans 属性、国际化的提示和消息。

（1）Struts框架结构　如图：

[](http://baike.baidu.com/albums/16184/5909255/0/0.html#0$bf4875634fc012520d33fa3c)

Struts对Model，View和Controller都提供了对应的组件。在图中，ActionServlet，这个类是Struts的核心控制器，负责拦截来自用户的请求。Action，这个类通常由用户提供，该控制器负责接收来自ActionServlet的请求，并根据该请求调用模型的业务逻辑方法处理请求，并将处理结果返回给JSP页面显示。

1. Model部分

由ActionForm和JavaBean组成，其中ActionForm用于封装用户的请求参数，封装成ActionForm对象，该对象被ActionServlet转发给Action，Action根据ActionForm里面的请求参数处理用户的请求。JavaBean则封装了底层的业务逻辑，包括数据库访问等。

1. View部分

该部分采用[JSP](http://baike.baidu.com/view/3387.htm" \t "http://blog.sina.com.cn/s/_blank)实现。Struts提供了丰富的标签库，通过标签库可以减少[脚本](http://baike.baidu.com/view/54.htm" \t "http://blog.sina.com.cn/s/_blank)的使用，自定义的标签库可以实现与Model的有效交互，并增加了现实功能。对应上图的JSP部分。

1. Controller组件

Controller组件有两个部分组成——系统核心控制器，业务逻辑控制器。系统核心控制器，对应上图的ActionServlet。该控制器由Struts框架提供，继承HttpServlet类，因此可以配置成标注的Servlet。该控制器负责拦截所有的[HTTP请求](http://baike.baidu.com/view/641736.htm" \t "http://blog.sina.com.cn/s/_blank)，然后根据用户请求决定是否要转给业务逻辑控制器。业务逻辑控制器，负责处理用户请求，本身不具备处理能力，而是调用Model来完成处理。对应Action部分。

1. Hibernate

Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。 Hibernate可以应用在任何使用JDBC的场合，既可以在Java的客户端程序使用，也可以在Servlet/JSP的Web应用中使用，最具革命意义的是，Hibernate可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。

（1）Hibernate的核心接口一共有5个，分别为:Session、[SessionFactory](http://baike.baidu.com/view/1097304.htm" \t "http://blog.sina.com.cn/s/_blank)、Transaction、[Query](http://baike.baidu.com/view/209416.htm" \t "http://blog.sina.com.cn/s/_blank)和Configuration。这5个核心接口在任何开发中都会用到。通过这些接口，不仅可以对持久化对象进行存取，还能够进行事务控制。下面对这五个核心接口分别加以介绍。

①Session接口

Session接口负责执行被持久化对象的CRUD操作(CRUD的任务是完成与数据库的交流，包含了很多常见的SQL语句。)。但需要注意的是[Session对象](http://baike.baidu.com/view/2442531.htm" \t "http://blog.sina.com.cn/s/_blank)是非线程安全的。同时，Hibernate的session不同于JSP应用中的HttpSession。这里当使用session这个术语时，其实指的是Hibernate中的session，而以后会将HttpSesion对象称为用户session。

②SessionFactory接口

SessionFactory接口负责初始化Hibernate。它充当数据存储源的代理，并负责创建Session对象。这里用到了工厂模式。需要注意的是SessionFactory并不是轻量级的，因为一般情况下，一个项目通常只需要一个SessionFactory就够，当需要操作多个数据库时，可以为每个数据库指定一个SessionFactory。

③Configuration接口

Configuration接口负责配置并启动Hibernate，创建SessionFactory对象。在Hibernate的启动的过程中，Configuration类的实例首先定位映射文档位置、读取配置，然后创建SessionFactory对象。

④Transaction接口　Transaction接口负责事务相关的操作。它是可选的，开发人员也可以设计编写自己的底层事务处理代码。

⑤Query和Criteria接口

Query和Criteria接口负责执行各种数据库查询。它可以使用HQL语言或SQL语句两种表达方式。

1. 工作原理

①读取并解析配置文件

②读取并解析映射信息，创建SessionFactory

③打开Sesssion

④创建事务Transation

⑤持久化操作

⑥提交事务

⑦关闭Session

⑧关闭SesstionFactory

（3）为什么使用Hibernate？

①对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。

②Hibernate是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作，将软件开发人员从大量相同的数据持久层相关编程工作中解放出来，使开发更对象化了。

③移植性好，支持各种数据库，如果换个数据库只要在配置文件中变换配置就可以了，不用改变hibernate代码。

④支持透明持久化，因为hibernate操作的是纯粹的（pojo）java类，没有实现任何接口，没有侵入性。

**三、MVC**

1. 简介

[MVC](https://baike.baidu.com/item/MVC" \t "https://baike.baidu.com/item/MVC%E6%A1%86%E6%9E%B6/_blank)全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

（二）框架内容

MVC指MVC模式的某种框架，它强制性的使应用程序的输入、处理和[输出](https://baike.baidu.com/item/%E8%BE%93%E5%87%BA" \t "https://baike.baidu.com/item/MVC%E6%A1%86%E6%9E%B6/_blank)分开。使用MVC应用程序被分成三个核心部件：模型、[视图](https://baike.baidu.com/item/%E8%A7%86%E5%9B%BE" \t "https://baike.baidu.com/item/MVC%E6%A1%86%E6%9E%B6/_blank)、控制器。它们各自处理自己的任务。最典型的MVC就是JSP + [servlet](https://baike.baidu.com/item/servlet" \t "https://baike.baidu.com/item/MVC%E6%A1%86%E6%9E%B6/_blank) + [javabean](https://baike.baidu.com/item/javabean" \t "https://baike.baidu.com/item/MVC%E6%A1%86%E6%9E%B6/_blank)的模式。在该模型中，应用程序分为三个组成部分:

View:

用户视图模块，负责用户界面部分。主管应用程序与人之间的接口，一方面，它为用户提供了输入手段，并触发应用逻辑允许，另一方面，它又将逻辑允许的结果以某种形式显示给用户。

Controller:

流程控制模块，该部分是用户界面与Model的接口。一方面它解释来自于view的输入，将其解释成系统能够理解的对象，同时它也识别用户动作，将其解释为对Model特定方法的调用;另一方面，它也处理来自于Model的事件和Model逻辑执行的结果，调用适当的View为用户提供反馈。

Model:

事务逻辑模块，是整个模型的核心。它表示解决方案真正的逻辑。它采用面向对象的方法，将问题领域中的对象抽象为应用程序对象。在这些抽象的对象中封装了对象的属性和这些对象所隐含的逻辑。

## IMG_256

（三）[SSH框架与MVC架构之间的关系](http://www.cnblogs.com/itao/archive/2011/08/11/2135454.html)

MVC：是指Model—View—Controler，是程序的一种分层模式，是一种思想。

SSH：是指struts2,spring,hibernate，是三种被封装的框架，是框架模式的实现，SSH是一种经典的MVC模式。

MVC三层架构：模型层，控制层和视图层。模型层，用Hibernate框架让来JavaBean在数据库生成表及关联，通过对JavaBean的操作来 对数据库进行操作；控制层，用Struts框架来连接数据层和视图层的，接收、处理、发送数据并控制流程；视图层，用JSP模板把页面展现给用户以及提供 与用户的交互。而Spring框架粘合了Hibernate和Struts，透明的管理了整个架构，提供IOC容器使代码松耦合以及AOP框架的切面功能等。