Tomcat主要目录

/bin– Tomcat： 脚本存放目录（如启动、关闭脚本）。\*.sh文件用于 Unix 系统；\*.bat文件用于 Windows 系统。

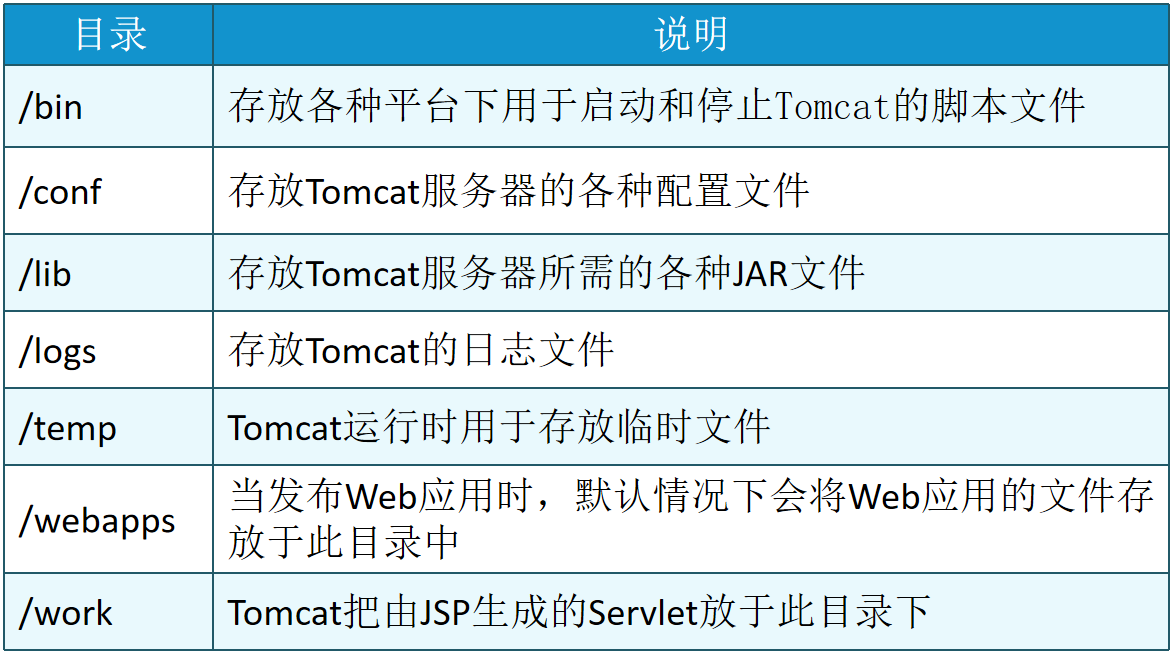
/conf– Tomcat： 配置文件目录。

/lib--Tomcat：tomcat服务器所需要的各个库文件

/logs– Tomcat： 默认日志目录。

/webapps– Tomcat：web应用程序存放的目录，web项目保存到此目录即可发布。

/work-- Tomcat：临时文件夹，生成所有的临时文件（\*.java、\*.class）



**一般 web 项目路径结构**

|-- webapp # 站点根目录

|-- META-INF # META-INF 目录

| `-- MANIFEST.MF # 配置清单文件

|-- WEB-INF # WEB-INF 目录

| |-- classes # class文件目录

| | |-- \*.class # 程序需要的 class 文件

| | `-- \*.xml # 程序需要的 xml 文件

| |-- lib # 库文件夹

| | `-- \*.jar # 程序需要的 jar 包

| `-- web.xml # Web应用程序的部署描述文件

|-- <userdir> # 自定义的目录

|-- <userfiles> # 自定义的资源文件

webapp：工程发布文件夹。其实每个 war 包都可以视为 webapp 的压缩包。

META-INF：META-INF 目录用于存放工程自身相关的一些信息，元文件信息，通常由开发工具，环境自动生成。

WEB-INF：Java web应用的安全目录。所谓安全就是客户端无法访问，只有服务端可以访问的目录。

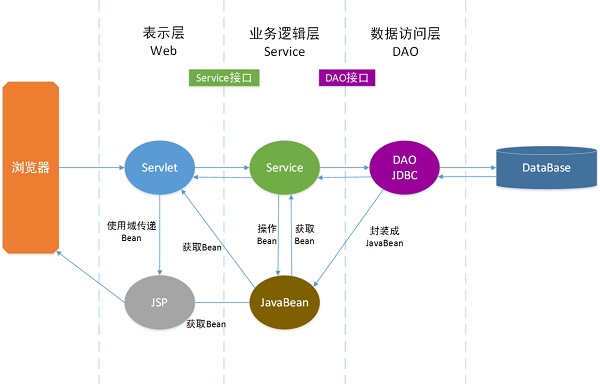
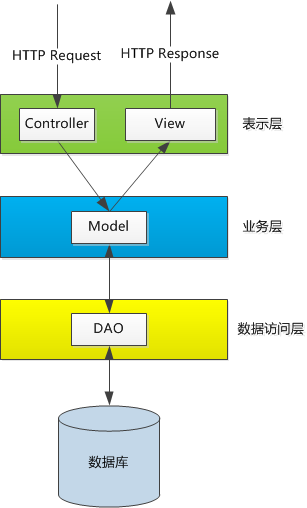
/WEB-INF/classes：存放程序所需要的所有 Java class 文件。

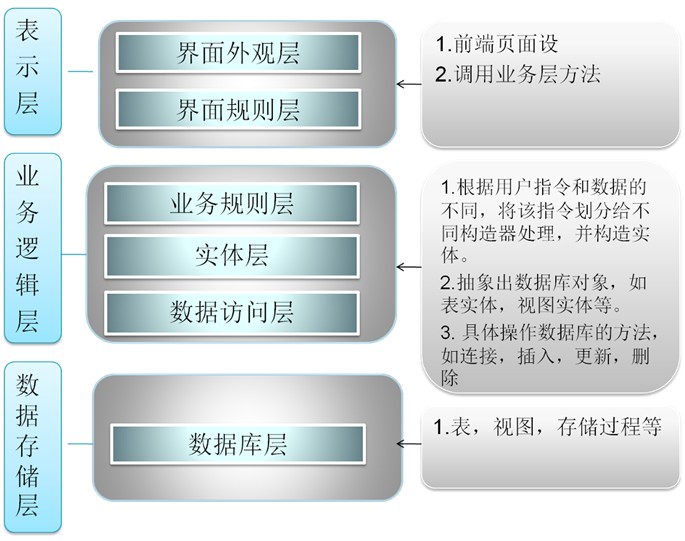
/WEB-INF/lib：存放程序所需要的所有 jar 文件。

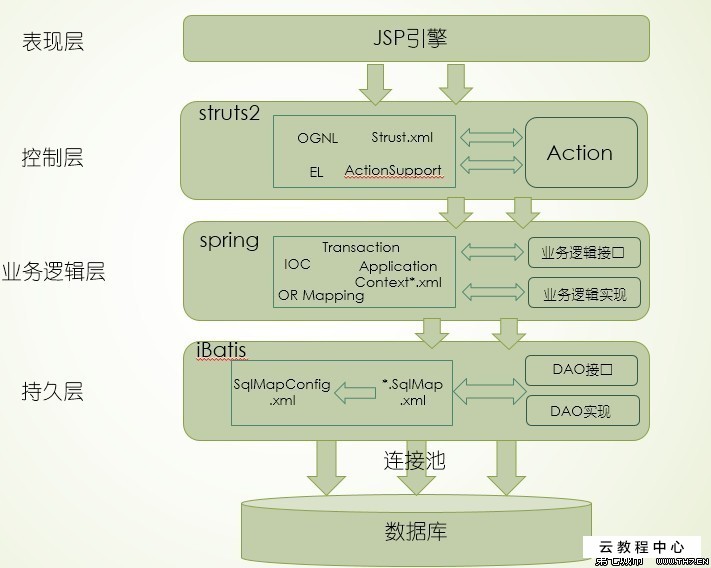
/WEB-INF/web.xml：web 应用的部署配置文件。它是工程中最重要的配置文件，它描述了 servlet 和组成应用的其它组件，以及应用初始化参数、安全管理约束等。



表现层、持久层、业务层、和域模型层。每一个层次都有其独特的职责，不能把各自的功能与其它层次相混合。每一个应用层都应该和其它层隔离开来，但允许使用接口在层间进行通信。







1. 持久层（Persistence layer–DAO–hibernate）

所谓持久，就是把数据保存到可以永久保持的存储设备当中。一般来说，持久更为直接的理解就是对数据库的各种操作，如增加，删除，修改，查询，更新等操作。

持久层，就是把持久的动作封装成一个独立的层，这是为了降低功能代码之间的关联。创建一个更清晰的抽象，提高代码的内聚力，降低代码的耦合度，从而增强代码的要劳动局生和可重用性。

持久层可以封装数据访问细节，为大部分业务逻辑提供面向对象的API。它是一种松散耦合，使持久化不依赖于义愤数据库和上层业务逻辑来实现。我们在设计代码的时候，应该避免在业务逻辑代码中混杂数据访问代码。

DAO，即持久层数据访问对象。利用来抽象和封装所有对数据源的访问，由DAO来管理各种数据源的连接以便于检索和存储数据。DAO一般由业务对象，数据访问对象，数据源和值对象组成。通过DAO实现了数据访问逻辑的抽象与实现细节的封装，只对外提供一个抽象化的数据访问API。我们通过调用API的方法就能实现对数据库的操作。而不必再去理解具体是怎么对数据库进行操作的。这样由于业务操作和数据访问实现分离，也使得开发人员的专业划分成为可能。

在DAO中业务对象仅仅包含与本领域相关的逻辑对象和算法就可以了。针对不同的数据库提供各自的数据访问代码，可以通过定义一个调用接口，针对该接口实现不同数据的数据访问。这样就可以避免对具体实现的依赖，在以后数据库变更时，不用改实现的代码。

Dao层是可以使用Hibernate连接数据库、操作数据库（简单的增删改查）。其和DAO设计模式还不完全是一个概念，我们此处的Dao层中不包含实体类的描述，这部分写在域模型层：

DAO设计模式一般分为几个类：

1.VO(Value Object)：一个用于存放网页的一行数据即一条记录的类，比如网页要显示一个用户的信息，则这个类就是用户的类。

2.DatabaseConnection：用于打开和关闭数据库。

3.DAO接口：用于声明对于数据库的操作。

4.DAOImpl：必须实现DAO接口，真实实现DAO接口的函数，但是不包括数据库的打开和关闭。

5.DAOProxy：也是实现DAO接口，但是只需要借助DAOImpl即可，但是包括数据库的打开和关闭。

6.DAOFactory：工厂类，含有getInstance()创建一个Proxy类。