<https://blog.csdn.net/u011687186/article/details/61194580>

/WEB-INF/web.xml

Web应用程序配置文件，描述了 servlet 和其他的应用组件配置及命名规则。

 /WEB-INF/classes/  
包含了站点所有用的 class 文件，包括 servlet class 和非servlet class，他们不能包含在 .jar文件中。  
 /WEB-INF/lib/  
存放web应用需要的各种JAR文件，放置仅在这个应用中要求使用的jar文件,如**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)**驱动jar文件。

 /WEB-INF/src/  
 源码目录，按照包名结构放置各个[**Java**](http://lib.csdn.net/base/javase)文件。

 /WEB-INF/database.properties  
 数据库配置文件

 /WEB-INF/tags/

存放了自定义标签文件，该目录并不一定为 tags，可以根据自己的喜好和习惯为自己的标签文件库命名，当使用自定义的标签文件库名称时，在使用标签文件时就必须声明正确的标签文件库路径。例如：当自定义标签文件库名称为 simpleTags 时，在使用 simpleTags 目录下的标签文件时，就必须在 jsp 文件头声明为：<%@ taglibprefix="tags" tagdir="/WEB-INF /simpleTags" % >。

/WEB-INF/jsp/

jsp 1.2 以下版本的文件存放位置。改目录没有特定的声明，同样，可以根据自己的喜好与习惯来命名。此目录主要存放的是 jsp 1.2 以下版本的文件，为区分 jsp 2.0 文件，通常使用 jsp 命名，当然你也可以命名为 jspOldEdition 。

/WEB-INF/jsp2/

与 jsp 文件目录相比，该目录下主要存放 Jsp 2.0 以下版本的文件，当然，它也是可以任意命名的，同样为区别 Jsp 1.2以下版本的文件目录，通常才命名为 jsp2。

META-INF

相当于一个信息包，目录中的文件和目录获得[**Java**](http://lib.csdn.net/base/java)2平台的认可与解释，用来配置应用程序、扩展程序、类加载器和服务  
 manifest.mf文件，在用jar打包时自动生成。

**以下转载至 http://blog.csdn.net/saygoodbyetoyou/article/details/9944773**

## 请求页面为何要放到META-INF目录下

**观点一：（较为赞同）**  
安全性不是真正的原因，因为jsp是要解析后才显示到浏览器的，即使用户知道你jsp的路径，也不可能通过浏览器看到jsp源码的，而如果是通过其它手段入侵服务器的话，放在WEB-INF又和放在别的地方有什么区别呢？？？  
真正的原因是，对于早期直接嵌入Java代码的jsp，是直接给出jsp路径给用户访问的，这种情况恰恰不能放在WEB-INF，而到了现在的mvc模式，jsp已经不是曾经那个包含完整逻辑的jsp，而仅仅充当view层的模板，必须要填入model数据后进行渲染，才能生成可读的页面，这样的模板是不能让用户直接访问的，直接访问的话要么是一堆各种空白，或者一堆占位符，甚至一堆NullPointException，所以需要放到WEB-INF里。补充一下，“安全性问题”指的是“不允许你访问，否则会对我不好”，而这里的情况其实是“不建议你访问，否则会对你不好”，应该属于健壮性的范畴。  
**观点二：**  
  放到WEB-INF下的Jsp文件，如果你直接去访问这个Jsp是不可以的，必须通过Action的跳转，这样就能够理解为什么安全性高了，对于用户是不可见的，对于一些对于用户，可以直接访问的，如，登录，注册页面，可以直接放在webroot下，这个也是根据你系统的需求了。  
**观点三：**  
通过设置过滤器，放在webroot下面的文件也可以实现不能直接访问。所以说放在哪里就看习惯是什么样了。  
一般项目都是要求隐藏性的，只让客户通过请求访问而不是直接访问jsp页面。若放在webroot下面，肯定要加一个过滤器阻止所有对\*.jsp的访问。只要比较的话：  
放在webroot下面:优点，程序结构清晰，便于编码和维护；缺点，要加过滤器。  
放在web-inf下面：优点，不用过滤器；缺点，打乱了程序结构，编码和维护麻烦  
**观点四：**  
为了减少风险，可以把这些页面文件移到WEB-INF 目录下。基于Servlet的声明，WEB-INF不作为Web应用的公共文档树的一部分。因此，WEB-INF 目录下的资源不是为客户直接服务的。我们仍然可以使用WEB-INF目录下的JSP页面来提供视图给客户，客户却不能直接请求访问JSP。JSP存放在 WEB-INF 目录下更为安全