



链滴

从 Latke 开始学习 Java web 开发

作者: [orz0](#)

原文链接: <https://ld246.com/article/1478675409521>

来源网站: [链滴](#)

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

作为一个PHP程序员内心是空虚的,抛开了fastcgi我就无所适从了.所以我要学一些和以往不一样的东西. Java的web开发框架很多,为什么选择latke?没有理由,因为我没得对比嘛.为什么要阅读框架而不是基框架做一个应用?因为个人习惯.做php时也是一样,每用一个新框架都会读一遍它的源码(结构上),当然php的框架结构都很简单.

我对Java一无所知,以下一定会有错误疏漏,若有路过的大神肯开示一二,不胜感激!

这个学习笔记暂时会停留在servlet, latke上,暂时不会跨越jetty.

从Starter开始

```
Latkes.setScanPath("org.b3log.latke.demo.hello"); // For Latke IoC
Latkes.initRuntimeEnv();
```

```
String classesPath = ClassLoader.getSystemResource("").getPath(); // Real path including maven sub folder
```

```
String webappDirLocation = classesPath.replace("target/classes/", "src/main/webapp/"); // PM structure in dev env
```

```
final File file = new File(webappDirLocation);
if (!file.exists()) {
    webappDirLocation = "." // production environment
}
```

```
System.out.println(webappDirLocation);
```

```
final Server server = new Server(Integer.valueOf(Latkes.getServerPort()));
final WebApplicationContext root = new WebApplicationContext();
root.setParentLoaderPriority(true); // Use parent class loader
root.setContextPath("/");
root.setDescriptor(webappDirLocation + "/WEB-INF/web.xml");
root.setResourceBase(webappDirLocation);
server.setHandler(root);
```

```
try {
    server.start();
} catch (final Exception e) {
    e.printStackTrace();

    System.exit(-1);
}
```

前两行Latkes的先略过不表.

ClassLoader.getSystemResource("") 得到了什么?

行为上看: 按照classpath顺序查找参数中指定的资源,包括jar包里面.但是当参数为空串的时候,只会查目录而不包含jar包.

比如: `java -cp WEB-INF/lib/*:WEB-INF/classes Starter`

传递空串会返回/path/to/WEB-INF/classes路径

非空串则会检索jar包,返回这种路径 /path/to/WEB-INF/lib/cdi-api-1.0.jar!beans.xsd

但内部实现没有找到,如果有人知道请指教.

`classPath.replace`目的是调试, 因为打包后的和开发时生成的目录结构不一样, 应该算是一个补丁.

上面的代码在产品环境下按照注释:

```
java -cp WEB-INF/lib/*:WEB-INF/classes org.b3log.latke.demo.hello.Starter
```

应该是跑不通的,因为找不到web.xml,只需要 `-cp .:WEB-INF/lib/*:WEB-INF/classes` 这样, 原因还是前面的classloader.

接下来都是关于jetty的, 主要是加载web.xml, 相关资料看这里: <http://www.eclipse.org/jetty/documentation/current/embedding-jetty.html>, 暂时不会深入.

Latkes.getServerPort()

问题: 这个port是怎么来的呢?

我们回到最前面两句

```
Latkes.setScanPath("org.b3log.latke.demo.hello"); // For Latke IoC  
Latkes.initRuntimeEnv();
```

Latke类有一个静态初始化块, 里面加载了

- latke.properties
- local.properties
- remote.properties

三个文件, 第一个是latke框架本身的配置.端口在此配置.

`Latkes::initRuntimeEnv` 这个函数根据 `latke.properties` 初始化了运行环境和数据库配置, 看代码 `runtimeEnv` 的值总是 `Local`, 这样的话就会总是去初始化数据库了.

latke到这里算是一个阶段小阶段.

接下来看一点servlet相关的

Starter的代码中, jetty加载了web.xml, 到底jetty如何运作我们先不管,以免战线太长.总之先理解为jett按照web.xml的说明进行一些初始化工作吧.

但是servlet是绕不过去的,先看web.xml定义了什么:

```
Listener: org.b3log.latke.demo.hello.HelloServletListener
```

(编辑器不能插入xml标签)

里面设定了一个servlet的listener, 什么是listener呢?看此文档: <https://docs.oracle.com/cd/B1409919/web.1012/b14017/filters.htm>

如上文档可知, event listener在servlet(容器?)发生关键事件时给予应用监听此事件的能力. servlet事包含两个级别:

- 应用级别(如何形容?就是应用)
- 会话级别(同一个用户的一系列请求)

但是看latke的代码还有一个请求级别的ServletRequestListener, 暂时还不确定和session级别有多大区别之后验证.

现在我们知道jetty在初始化后会通过ServletContextListener通知我们应用可以接收请求了.接下来latke到底做了哪些事情?

明天继续.

PS: jetty工作原理,可以先大概了解下 <http://www.ibm.com/developerworks/cn/java/j-lo-jetty/index.html>