

从 Latke 开始学习 Java web 开发 (1)

作者: [orz0](#)

原文链接: <https://ld246.com/article/1478676854371>

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PHP程序员是很无聊的, 所以看看Java.我对Java一无所知,以下内容一定有错漏的地方,请路过的高开示一二.多谢多谢.

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

从latke-demo的Starter.java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/"); // POM 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);

}
```

这段代码前两行Latkes的先略过不表.从这里开始看:

ClassLoader.getResource("")

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,返回类似这种路径 /path/to/WEB-INF/lib/cdi-api-1.0.jar!beans.xsd

内部实现没有找到.

classPath.replace目的是调试, 因为后和开发时生成的目录结构不一样, 所以算是一个补丁.

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, 这样做的原因还看前面的classloader部分.

接下来都是关于jetty的, 相关资料看这里: [embedding-jetty](#),暂时不会深入.

Latkes.getServerPort()

问题: 这个port是怎么来的呢?

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.那latke是如何加载的呢?回到代码的第二句.

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.不确定为什么.

latke到这里算是告一个小小的段落,接下来有一些servlet相关的东西.

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:

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

(这个编辑器不支持xml标签)

这一行定义了servlet的listener, 什么是listener呢?看这里: [filters & listeners](#)

event listener在servlet发生关键事件时给予应用监听此事件的能力. servlet事件包含两个级别:

- 应用级别(怎么形容?就是应用吧)
- 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`通知我们可以接收请求了.接下来latke底做了哪些事情?下回再说.

PS: jetty工作原理, [<http://www.ibm.com/developerworks/cn/java/j-lo-jetty/index.html>]([http://www.ibm.com/developerworks/cn/java/j-lo-jetty/index.html%5D\(http://www.ibm.com/developerworks/cn/java/j-lo-jetty/index.html](http://www.ibm.com/developerworks/cn/java/j-lo-jetty/index.html%5D(http://www.ibm.com/developerworks/cn/java/j-lo-jetty/index.html)))