Java Web 框架开发基础

作者：88250
原文链接：https://ld246.com/article/1423555085029
来源网站：链滴

请求处理

最常用也最简单的分发原理是根据请求 URI（request.getRequestURI()）来判断分发。

- 一个请求处理方法可以接受多种请求 URI 的请求
- 支持 Ant 路径与正则表达式
- 请求方法指定（GET/POST/etc）

```
@RequestProcessing(value = {"/\\d*", ""}, uriPatternsMode = URIPatternMode.REGEX, method = HTTPRequestMethod.GET)
```

在实现时，框架需要默认包含对上下文路径的处理，以便应用部署在非 ROOT 路径也能正常工作。

请求与响应

对于请求的处理是分为两部分的：操作（Action）、渲染（Render）。

操作部分相当于业务逻辑实现，例如入参校验，服务调用；渲染部分则是针对响应对象的，是需要进统一抽象的。

通过渲染器进行响应输出，渲染器主要逻辑包括：

- 设置响应头
- 生成响应内容
- 渲染，即 Response 写回

应该提供多种响应渲染器（例如 HTML、图片、JSON 等），方便应用直接使用。

另外，为了方便扩展渲染器，可以提供模版方法 beforeRender 和 afterRender。

模版处理

模版引擎有两个接口最为常用：

- 模版目录加载设置
- 获取模版

应用调用模版目录加载设置接口确定模版文件目录，如果有动态切换场景（例如换皮肤），则也调用接口。实现时这个接口必须使用 ServletContex 作为参数来定位当前运行的 Web 应用，因为用户在署 WAR 时可能会不展开，此时使用 Servlet API 是获取不到正确的 WAR 绝对路径的。

另外，框架可提供一个 boolean hasExpression(Template template, String expression) 接口，用判断指定的模版中是否定义了某个表达式。

这个接口可用于应用实现一些灵活的“按需加载”功能，没有表达式就执行业务逻辑实现（不加载/装数据），方便应用进行特殊逻辑处理或性能优化。

事件机制

事件机制本质上是观察者模式的实现，需要包含同步事件和异步事件。框架本身提供一些内置的事件

用于扩展解耦。

扩展点

扩展点可以使用模版方法实现或事件机制实现的。框架在请求-响应这个主流程不同阶段的处理中应提供一些扩展点，例如：

- 请求处理前
- 服务调用前、后
- 数据存取（Datastore，SQL）执行前、后
- 渲染前、渲染后
- 请求处理后
- 自动缓存清空前

在请求-响应处理顶层流程中，提供模版方法进行分发、渲染等逻辑的接口，这样方便与其他框架对
；
而在具体的渲染逻辑中则可以触发事件，方便框架用户进行事件监听并扩展。

依赖注入

单例生存周期的依赖注入必不可少，其他生存周期（请求、会话等）不常用，因为有状态的服务端设并不常用。

容器管理对象后，通过动态代理实现 AOP，为声明式事务管理、方法调用织入提供基础。

插件

插件执行是事件触发的，相同的事件类型一次可以触发多个插件执行，执行顺序按照插件注册顺序。

插件是需要可以完成完整业务功能的，包括了服务端逻辑执行、视图渲染、 插件数据存储等。

ORM

不要过度设计和封装（反面教材 Hibernate，正面教材 MyBatis），但需要提供：

- 连接池切换机制
- 连接获取接口
- 编程式/声明式事务，可配置传播属性

实现时进行简单的数据行-对象映射处理即可，CRUD 和简单 Query 可以支持大部分业务场景，较复的业务可以直接上 SQL。

工具

- 分页、ID 生成、序列化、反射、堆栈、秒表等
- 日志库封装

- 缓存、HTTP Client、邮件、任务队列、定时任务、图片处理等

构建

在构建阶段，框架需要做字节码增强，例如一些注解的功能实现。
除此以外，框架可以提供一些构建工具，例如静态资源压缩合并插件。

错误页面处理

Servlet 规范本身已经定义了统一的错误页面处理（web.xml ）机制，框架的请求分发逻辑中可以通过 request.getAttribute("javax.servlet.error.status_code"); 来判断是否是错误页面的请求。

容器兼容

虽然容器都是实现 Servlet 规范的，但是一些细节还是不相同，需要框架做统一处理：

- 有可能需要考虑容器的虚拟文件系统实现（例如 jboss-vfs）
- 有些容器自带了依赖注入 CDI 实现，可能会与框架实现冲突
- 静态资源转发使用容器的性能更好，但是不同容器的 Servlet 处理不同 （Default Servlet Name）

参考

- 为什么又要造一个叫 Latke 的轮子
- Servlet 规范、依赖注入规范等