链滴

Solo 自定义链接路由

作者：88250

原文链接：https://ld246.com/article/1543582332567

来源网站：链滴

本文是《Solo 从设计到实现》的一个章节，该系列文章将介绍 Solo 这款 Java 博客系统是如何从无有的，希望大家能通过它对 Solo 从设计到实现有个直观地了解、能为想参与贡献的人介绍清楚项目也希望能为给~~重复发明~~重新定义博客系统的人做个参考
eart

自定义链接路由

“自定义链接”指的是发布文章或者页面时指定的 URL path 部分，自定义链接主要是为了让 URL “看”一些，对 SEO 也有一定帮助。

Solo 通过 Latke 框架提供的 Handler 扩展来实现这个特性，在 PermalinkHandler.java 中，先根据求的 URI 解析出 permalink，然后查库判断是否存在数据（可能是文章也可能是页面），如果不存在走常规的请求分发：

```
final ArticleRepository articleRepository = beanManager.getReference(ArticleRepository.class)

article = articleRepository.getByPermalink(permalink);
if (null == article) {
    LOGGER.log(Level.DEBUG, "Not found article with permalink [{0}]", permalink);
    context.handle();

    return;
}
```

如果存在数据记录，则通过方法 dispatchToArticleProcessor 直接分发给对应的控制器，数据记录到请求属性中传递：

```
request.setAttribute(Article.ARTICLE, article);
request.setAttribute(Keys.HttpRequest.REQUEST_URI, Latkes.getContextPath() + "/article");
request.setAttribute(Keys.HttpRequest.REQUEST_METHOD, HttpMethod.GET.name());
```

Latke 框架在进行请求路由解析时会先从请求属性中获取 URI 路径，所以这里的设置操作相当于覆盖原始的请求路径。然后在 ArticleProcessor#showArticle 方法中直接从请求属性中获取文章记录，后再处理渲染：

```
// See PermalinkHandler#dispatchToArticleProcessor()
final JSONObject article = (JSONObject) request.getAttribute(Article.ARTICLE);
if (null == article) {
    response.sendError(HttpServletResponse.SC_NOT_FOUND);

    return;
}

final String articleId = article.optString(Keys.OBJECT_ID);
LOGGER.log(Level.DEBUG, "Article [id={0}]", articleId);

final AbstractFreeMarkerRenderer renderer = new SkinRenderer(request);
context.setRenderer(renderer);
renderer.setTemplateName("article.ftl");

...
```

优化考量

在过滤器中，为了优化性能，做了一个链接格式的判断：

```
if (PermalinkQueryService.invalidPermalinkFormat(permalink)) {
    LOGGER.log(Level.DEBUG, "Skip permalink handling request [URI={0}]", permalink);
    context.handle();

    return;
}
```

根据链接的格式可以判断出“明显不是”自定义链接的请求，这样就不用查库了，减少查库次数以优性能。

另外，通过请求属性传递数据实体也是为了减少查库，既然在过滤器中查出来了，在控制器中就没必再查库了。