链滴

# Solo 导入 Markdown 文章

作者：88250

原文链接：https://ld246.com/article/1547643961141

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

如果你之前使用过静态博客系统（比如 Hexo、Jekyll），Solo 可以很方便地将你已有的 Markdown 文件进行导入，并且在最大程度上保留以前的元信息，比如发布时间、标签、自定义链接等。

## 导入时机

因为导入操作不是必要且高频的操作，所以设计上为了保持简单，我们选择在启动时扫描固定路径 markdowns 文件夹，然后将该文件夹下所有 *.md 文件进行解析并作为文章导入。

文件夹路径获取：

```
final ServletContext servletContext = SoloServletListener.getServletContext();
final String markdownsPath = servletContext.getRealPath("markdowns");
```

导入主逻辑：

```
final Collection<File> mds = FileUtils.listFiles(new File(markdownsPath), new String[]{"md"}, tr
e);
if (null == mds || mds.isEmpty()) {
    return;
}

for (final File md : mds) {
    final String fileName = md.getName();
    if (StringUtils.equalsIgnoreCase(fileName, "README.md")) {
        continue;
    }

    try {
        final String fileContent = FileUtils.readFileToString(md, "UTF-8");
        final JSONObject article = parseArticle(fileName, fileContent);
        article.put(Article.ARTICLE_AUTHOR_ID, adminId);

        final JSONObject request = new JSONObject();
        request.put(Article.ARTICLE, article);

        final String id = articleMgmtService.addArticle(request);
        FileUtils.moveFile(md, new File(md.getPath() + "." + id));
        LOGGER.info("Imported article [" + article.optString(Article.ARTICLE_TITLE) + "]");
        succCnt++;
    } catch (final Exception e) {
        LOGGER.log(Level.ERROR, "Import file [" + fileName + "] failed", e);

        failCnt++;
        failSet.add(fileName);
    }
}
```

导入结束后原 md 文件将被重命名为 .md.{时间毫秒} 这样的格式，如不需要，可将这类后缀的文件

除。

## 文件解析

每个 md 文件都会按照 Hexo/Jekyll 定义的头部进行解析，以确定标题、标签等：

- Hexo 头
- Jekyll 头
- 支持头信息中使用 description、summary、abstract 作为文章摘要，如果没有的话将自动截取文部分
- 如果没有定义头信息，或者解析失败，则以文件名作为标题、 Note 作为标签、当前时间作为发时间进行导入，这也是导入普通 md 文件的规则

Hexo/Jekyll 的文件头是使用 Yaml 格式的，我们使用 org.yaml.snakeyaml 库进行解析：

```
fileContent = StringUtils.trim(fileContent);
String frontMatter = StringUtils.substringBefore(fileContent, "---");
if (StringUtils.isBlank(frontMatter)) {
    fileContent = StringUtils.substringAfter(fileContent, "---");
    frontMatter = StringUtils.substringBefore(fileContent, "---");
}

final JSONObject ret = new JSONObject();
final Yaml yaml = new Yaml();
Map elems;
try {
    elems = (Map) yaml.load(frontMatter);
} catch (final Exception e) {
    // treat it as plain markdown
    ret.put(Article.ARTICLE_TITLE, StringUtils.substringBeforeLast(fileName, "."));
    ret.put(Article.ARTICLE_CONTENT, fileContent);
    ret.put(Article.ARTICLE_ABSTRACT, Article.getAbstract(fileContent));
    ret.put(Article.ARTICLE_TAGS_REF, DEFAULT_TAG);
    ret.put(Article.ARTICLE_IS_PUBLISHED, true);
    ret.put(Article.ARTICLE_COMMENTABLE, true);
    ret.put(Article.ARTICLE_VIEW_PWD, "");

    return ret;
}
```

## 容错处理

解析时难免会遇到一些“不按套路出牌”的内容，为了尽量兼容各种情况，我们得做一些容错处理：

- 缺少标题则按文件名作为标题
- 兼容各种日期格式： yyyy/MM/dd HH:mm:ss、yyyy-MM-dd HH:mm:ss、dd/MM/yyyy HH:m:ss 等，解析失败则以当前时间作为文章创建时间
- 抽取摘要：依次获取 description、summary、abstract，如果都没有则按默认规则抽取
- 解析标签：依次获取 tags、category、categories、keyword、keywords，如果都没有则默认加 Note 作为标签

所以，理论上即使不是 Hexo/Jekyll 的 md 文件，只是普通 md 文件也是可以导入 Solo 的。