



链滴

# 肝了一个半月的 Java 项目快速开发脚手架 ： Chewing

作者：[jingqueyimu](#)

原文链接：<https://ld246.com/article/1612281252396>

来源网站：[链滴](#)

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

```
<h2 id="前言">前言</h2>
<p>闲来无事，整一个 Java 项目快速开发脚手架。 </p>
<h2 id="正文">正文</h2>
<h2 id="一-简介">一、简介</h2>
<p>Chewing 是一个简单的 Java 项目快速开发脚手架。既适合需要开发小型项目的小伙伴使用，也合刚入门的新手用来学习一些常用的技术。 </p>
<h2 id="二-源码">二、源码</h2>
<ul>
<li>GitHub: <a href="https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Fjingqueyimu%2Fchewing" target="_blank" rel="nofollow ugc">https://github.com/jingqueyimu/hewing</a>。 </li>
<li>Gitee: <a href="https://ld246.com/forward?goto=https%3A%2F%2Fgitee.com%2Fjingqueyimu%2Fchewing" target="_blank" rel="nofollow ugc">https://gitee.com/jingqueyimu/chewing</a>。 </li>
</ul>
<h2 id="三-核心技术">三、核心技术</h2>
<h3 id="1-后端">1、后端</h3>
<ul>
<li>SpringBoot: Web 应用框架。 </li>
<li>MyBatis: 持久层框架。 </li>
<li>MySQL: 关系型数据库。 </li>
<li>Redis: 缓存数据库。 </li>
<li>RabbitMQ: 消息队列中间件。 </li>
</ul>
<h3 id="2-前端">2、前端</h3>
<ul>
<li>Themeleaf: 模板引擎。 </li>
<li>Bootstrap: UI 框架。 </li>
</ul>
<h2 id="四-环境部署">四、环境部署</h2>
<h3 id="1-准备工作">1、准备工作</h3>
<ul>
<li>JDK 1.8</li>
<li>MySQL 5.7</li>
<li>Maven 3.x</li>
<li>Redis</li>
<li>RabbitMQ</li>
</ul>
<h3 id="2-必要配置">2、必要配置</h3>
<ul>
<li>application.properties 配置文件: 数据库、Redis、RabbitMQ、邮箱等配置。 </li>
<li>config/myconfig.properties 配置文件: 系统相关的自定义配置。 </li>
</ul>
<h2 id="五-项目介绍">五、项目介绍</h2>
<h3 id="1-文件结构">1、文件结构</h3>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">src/main/java
</span></span><span class="highlight-line"><span class="highlight-cl">    |— com.jingque
imu
</span></span><span class="highlight-line"><span class="highlight-cl">    |— annotatio
// 注解
</span></span><span class="highlight-line"><span class="highlight-cl">    |— aspect
// 切面
</span></span><span class="highlight-line"><span class="highlight-cl">    |— config
```

```

// 配置
</span></span><span class="highlight-line"><span class="highlight-cl"> |— constant
  // 常量
</span></span><span class="highlight-line"><span class="highlight-cl"> |— context
  // 上下文
</span></span><span class="highlight-line"><span class="highlight-cl"> |— controller
  // 控制层
</span></span><span class="highlight-line"><span class="highlight-cl"> |— exception
  // 异常
</span></span><span class="highlight-line"><span class="highlight-cl"> |— factory
  // 工厂
</span></span><span class="highlight-line"><span class="highlight-cl"> |— filter
/ 过滤器
</span></span><span class="highlight-line"><span class="highlight-cl"> |— handler
  // 处理器
</span></span><span class="highlight-line"><span class="highlight-cl"> |— init    //
初始化
</span></span><span class="highlight-line"><span class="highlight-cl"> |— intercept
r    // 拦截器
</span></span><span class="highlight-line"><span class="highlight-cl"> |— mapper
  // 持久层
</span></span><span class="highlight-line"><span class="highlight-cl"> |— model
  // 数据模型
</span></span><span class="highlight-line"><span class="highlight-cl"> |— mq      //
消息队列
</span></span><span class="highlight-line"><span class="highlight-cl"> |— schedule
  // 调度
</span></span><span class="highlight-line"><span class="highlight-cl"> |— service
  // 服务层
</span></span><span class="highlight-line"><span class="highlight-cl"> |— util    //
工具
</span></span><span class="highlight-line"><span class="highlight-cl"> MyAppcation
java    // 应用启动类
</span></span><span class="highlight-line"><span class="highlight-cl">src/main/resource

</span></span><span class="highlight-line"><span class="highlight-cl"> |— config    /
配置文件
</span></span><span class="highlight-line"><span class="highlight-cl"> |— mapper
// 映射文件
</span></span><span class="highlight-line"><span class="highlight-cl"> |— static    //
静态文件
</span></span><span class="highlight-line"><span class="highlight-cl"> |— templates
  // 页面文件
</span></span><span class="highlight-line"><span class="highlight-cl"> application.pro
erties    // 应用配置文件
</span></span><span class="highlight-line"><span class="highlight-cl"> quartz.properti
s    // 调度配置文件
</span></span></code></pre>
<h3 id="2-代码说明">2、代码说明</h3>
<h4 id="-1-路由">(1) 路由</h4>
<ul>
<li>/api/xxx: 需要用户登录。 </li>
<li>/console/xxx: 需要管理员登录, 登录、登出等部分接口除外。 </li>
</ul>

```

## <h4 id="-2-接口规范"> (2) 接口规范</h4>

<ul>

<li>页面请求: GET、URL 参数。 </li>

<li>接口请求: POST、JSON 参数。 </li>

</ul>

## <h4 id="-3-数据库初始化"> (3) 数据库初始化</h4>

<ul>

<li>初始化配置文件: config/dbinit.json。 </li>

<li>配置初始化标识及 SQL 语句。 </li>

</ul>

<p>初始化示例: </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> [
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     "initKey": "sit  
_config_20210110",
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     "sqls": [  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "INSERT I  
TO t_site_config (id, code, name, content, description, public_flag, gmt_create) VALUES(NULL,  
site_record_no', '网站备案号', '&lt;a href=\"https://beian.miit.gov.cn\" class=\"ml-2\" target=\"  
blank\"&gt;备案号&lt;/a&gt;', '网站备案号', true, NOW());"
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     ]
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }  
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ]
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ]
```

```
</span> </span> </code> </pre>
```

## <h4 id="-4-文件上传下载"> (4) 文件上传下载</h4>

<ul>

<li>上传单个文件: /file/upload。 </li>

<li>上传多个文件: /file/uploads。 </li>

<li>下载文件: /file/download。 </li>

</ul>

## <h4 id="-5-Excel-导入导出"> (5) Excel 导入导出</h4>

<ul>

<li>Excel 导入导出方法: ExcelUtil.importExcel()、ExcelUtil.exportExcel()。 </li>

<li>Excel 导入导出处理器接口: IExcelImportHandler、IExcelExportHandler。 </li>

</ul>

<p>导入示例: </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> String msg = null;
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> try {  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     String[] keys =  
ew String[] {"username", "realName", "mobile"};
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     msg = ExcelUtil.  
mportExcel(file.getInputStream(), keys, new IExcelImportHandler() {
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         @Override  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         public void h
```

```
ndle(JSONObject data) {  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             if (StringUt
```

```
ls.isBlank(data.getString("username"))) {  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 throw n
```

```
w RuntimeException("用户名不能为空");  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             }  
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         if (StringUt
```

```
if (StringUt
```



```

dd(user.getLastLoginTime() == null ? "" : DateUtil.format(user.getLastLoginTime(), "yyyy-MM-
d HH:mm:ss"));
</span></span><span class="highlight-line"><span class="highlight-cl">        rowDatas.
dd(Boolean.TRUE.equals(user.getVipFlag()) ? "是" : "否");
</span></span><span class="highlight-line"><span class="highlight-cl">        return ro
Datas;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    });
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span> catch (IOExcepti
n e) {
</span></span><span class="highlight-line"><span class="highlight-cl">    e.printStackTrace();
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span> finally {
</span></span><span class="highlight-line"><span class="highlight-cl">    ...
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span>
</span></span></code></pre>

```

#### - <li>获取单个字典: /common/dict。 </li> - <li>获取多个字典: /common/dicts。 </li> - <li>UserContext: 用户上下文信息。 </li> - <li>AdminContext: 管理员上下文信息。 </li> - <li>新增调度任务: 继承 BaseJob。 </li> - <li>调度接口 - <li>执行调度任务: /console/schedule\_job/run。 </li> - <li>修改调度状态: /console/schedule\_job/update\_status。 </li> - <li>修改调度频率: /console/schedule\_job/update\_cron。 </li> </li> </ul> - <li>@Lock: 分布式锁。 </li> - <li>@Perm: 标注需要管理员权限的接口。 </li> - <li>由于个人习惯, 项目中使用的是 properties 配置文件, 对于习惯使用 yml 配置文件的, 请手动改。 </li> - <li>如果要加载自定义的 yml 配置文件, 可以使用项目中提供的 YAML 属性源工厂类: YamlPropert SourceFactory。 </li> <p>代码示例: </p> ``` <pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">@Component </span></span><span class="highlight-line"><span class="highlight-cl">@ConfigurationPr perties(prefix="test") ``` 原文链接: [肝了一个半月的 Java 项目快速开发脚手架: Chewing](#)

```
</span></span><span class="highlight-line"><span class="highlight-cl">@PropertySource(
  alue="classpath:config/test.yml", encoding="UTF-8", factory=YamlPropertySourceFactory.clas
)
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">public class TestY
  lConfig {
</span></span><span class="highlight-line"><span class="highlight-cl">  ...
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
```

#### (11) 属性名后缀匹配查询条件

BaseService 中以 JSON 对象为参数的方法，可通过在属性名后面添加后缀，来匹配查询条件。

- xxx\_begin: 大于等于。
- xxx\_end: 小于等于。
- xxx\_in: IN 查询。
- xxx\_like: 模糊查询。
- xxx\_llike: 左模糊查询。
- xxx\_rlike: 右模糊查询。
- 其他: 等于。
- 属性值为空: 不参与查询。

代码示例:

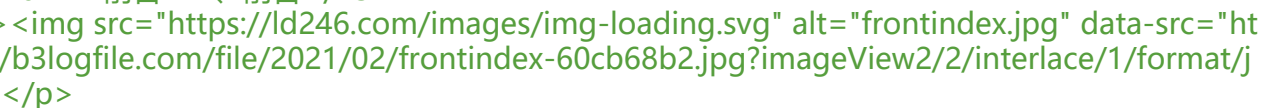
```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">@Test
</span></span><span class="highlight-line"><span class="highlight-cl">public void test() {
</span></span><span class="highlight-line"><span class="highlight-cl">  JSONObject pa
ams = new JSONObject();
</span></span><span class="highlight-line"><span class="highlight-cl">  params.put("us
rname_like", "test");
</span></span><span class="highlight-line"><span class="highlight-cl">  List<User>
user = userService.list(params);
</span></span><span class="highlight-line"><span class="highlight-cl">  System.out.print
n(user);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">@Test
</span></span><span class="highlight-line"><span class="highlight-cl">public void test2()
</span></span></code></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">  JSONObject pa
ams = new JSONObject();
</span></span><span class="highlight-line"><span class="highlight-cl">  params.put("us
rname_in", Arrays.asList("test"));
</span></span><span class="highlight-line"><span class="highlight-cl">  List<User>
user = userService.list(params);
</span></span><span class="highlight-line"><span class="highlight-cl">  System.out.print
n(user);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
```

#### (12) ...

## 六、演示图

### 1、前台

 frontindex.jpg

<p></p>

<p></p>

### <p></p> <p></p> <p></p> <p></p> <p></p> <p></p> <p></p> <p>目前，Chewing 还只是提供了一些较为常用的功能（实在是肝不动了）。但是，后续会不断完善、新增功能。有时间的话，也会整一个微服务版的。</p> <p>敬请期待~</p> <p align="center"> <br> 微信公众号：惊却一目<br> 个人博客：<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.jingqueyimu.com" target="\_blank" rel="nofollow ugc">惊却一目</a> </p>