



链滴

Java Web 框架 Latke v2.4.39, 重写控制器层

作者: [88250](#)

原文链接: <https://ld246.com/article/1545364310381>

来源网站: [链滴](#)

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

简介

Latke ('lɑ:tkɪ, 土豆饼) 是一个简单易用的 Java Web 应用开发框架, 包含 MVC、IoC、事件通知 ORM、插件等组件。

在实体模型上使用 JSON 贯穿前后端, 使应用开发更加快捷。这是 Latke 不同于其他框架的地方, 非常适合小型应用的快速开发。

特性

- 注解式、函数式路由
- 依赖注入
- 多种数据库 ORM
- 多语言
- 内存/Redis 缓存
- 事件机制
- 插件机制

案例

- [Demo](#): 简单的 Latke 应用示例
- [Solo](#): 一款小而美的 Java 博客系统
- [Symphony](#): 一款用 Java 实现的现代化社区 (论坛/BBS/社交网络/博客) 平台

安装

```
<dependency>
  <groupId>org.b3log</groupId>
  <artifactId>latke-core</artifactId>
  <version>${latke.version}</version>
</dependency>
```

控制器层用法

注解声明式路由

```
@RequestMapping("/")
public void index(final RequestContext context) {
    context.setRenderer(new SimpleFMRenderer("index.ftl"));
    final Map<String, Object> dataModel = context.getRenderer().getRenderDataModel();
    dataModel.put("greeting", "Hello, Latke!");
}
```

函数式路由

```
DispatcherServlet.post("/register", registerProcessor::register);
DispatcherServlet.mapping();
```

路径变量和查询字符串

```
@RequestMapping("/var/{pathVar}")
public void paraPathVar(final RequestContext context) {
    final String paraVar = context.param("paraVar");
    final String pathVar = context.pathVar("pathVar");
    context.renderJSON(new JSONObject().put("paraVar", paraVar).put("pathVar", pathVar));
}
```

JSON 解析

```
final JSONObject requestJSON = context.requestJSON();
```

Servlet 封装

```
final String remoteAddr = context.remoteAddr();
final String requestURI = context.requestURI();
final Object att = context.attr("name");
final String method = context.method();
context.sendRedirect("https://b3log.org");
final HttpServletRequest request = context.getRequest();
final HttpServletResponse response = context.getResponse();
```

服务层用法

依赖注入、事务

```
@Service
public class UserService {

    private static final Logger LOGGER = Logger.getLogger(UserService.class);

    @Inject
    private UserRepository userRepository;

    @Transactional
    public void saveUser(final String name, final int age) {
        final JSONObject user = new JSONObject();
        user.put("name", name);
        user.put("age", age);

        String userId;

        try {
            userId = userRepository.add(user);
        } catch (final RepositoryException e) {
            LOGGER.log(Level.ERROR, "Saves user failed", e);

            // 抛出异常后框架将回滚事务
            throw new IllegalStateException("Saves user failed");
        }

        LOGGER.log(Level.INFO, "Saves a user successfully [userId={0}]", userId);
    }
}
```

```
}
```

持久层用法

构造 ORM

```
@Repository
public class UserRepository extends AbstractRepository {

    public UserRepository() {
        super("user");
    }
}
```

单表 CRUD

```
public interface Repository {
    String add(final JSONObject jsonObject) throws RepositoryException;
    void update(final String id, final JSONObject jsonObject) throws RepositoryException;
    void remove(final String id) throws RepositoryException;
    void remove(final Query query) throws RepositoryException;
    JSONObject get(final String id) throws RepositoryException;
    long count(final Query query) throws RepositoryException;
}
```

条件查询

```
public JSONObject getByName(final String name) throws RepositoryException {
    final List<JSONObject> records = getList(new Query().
        setFilter(new PropertyFilter("name", FilterOperator.EQUAL, name)));
    if (records.isEmpty()) {
        return null;
    }

    return records.get(0);
}
```

分页查询

```
new Query().setCurrentPageNum(1).setPageSize(50)
```

按字段排序

```
new Query().addSort("name", SortDirection.DESCENDING);
```

仅获取需要字段

```
new Query().addProjection("name", String.class);
```

原生 SQL

```
final List<JSONObject> records = select("SELECT * FROM `user` WHERE `name` = ?", name);
```

文档

- [《提问的智慧》精读注解版](#)
- [为什么又要造一个叫 Latke 的轮子](#)
- [Latke 快速上手指南](#)
- [Latke 配置剖析](#)
- [Latke 贡献指南](#)

社区

- [讨论区](#)
- [报告问题](#)

鸣谢

Latke 的诞生离不开以下开源项目：

- [FreeMarker](#)：使用广泛的 Java 模版引擎
- [Commons Lang](#)：Java 语言相关工具库
- [Commons IO](#)：Java IO 相关工具库
- [Commons Codec](#)：Java 编解码库
- [Javassist](#)：Java 字节码处理工具库
- [SLF4j](#)：Java 日志门户
- [Gin](#)：又快又好用的 golang HTTP web 框架