



链滴

Sym 安装指南

作者: [88250](#)

原文链接: <https://ld246.com/article/1486188905847>

来源网站: [链滴](#)

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

本文档主要面向 Sym 社区版开发者，商业版采购后有单独的开发指南提供。

数据库

- 建议用 MySQL 8，最低支持 5.7
- 手动新建数据库 `b3log_symphony`，字符集用 `utf8mb4`，排序规则 `utf8mb4_general_ci`

数据库准备好后可选择如下方式部署。

源码编译

后端

1. 拉源码 `git clone https://github.com/88250/symphony.git`
2. 按需修改 `src/main/resources/local.properties` 中的数据库口令配置

```
jdbc.username=root  
jdbc.password=
```

3. 执行 `mvn package` 编译打包，完成后会生成 `target/symphony` 目录
4. 在 `symphony` 目录下执行：
 - Windows: `java -cp "lib/*;" org.b3log.symphony.Server`
 - Unix-like: `java -cp "lib/*:" org.b3log.symphony.Server`

也可以[下载](#)我们打好的包解压后在根目录通过以上命令启动；在 IDE 中的话可以直接运行 `Server.java`

前端

1. 下载 node
2. 根目录命令行运行

```
npm i
```

3. 开发时运行，以便调试

```
npm run dev
```

4. 部署前运行，以便进行代码合并和压缩

```
npm run build
```

Docker

1. 拉镜像 `docker pull b3log/symphony`
2. 启动容器

```
docker run --detach --name sym --network=host \  
--env RUNTIME_DB="MYSQL" \  

```

```
--env JDBC_USERNAME="root" \  
--env JDBC_PASSWORD="123456" \  
--env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \  
--env JDBC_URL="jdbc:mysql://127.0.0.1:3306/symphony?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC" \  
b3log/symphony --listen_port=8080 --server_scheme=http --server_host=localhost --serve_port=
```

启动参数说明:

- `--listen_port`: 进程监听端口
- `--server_scheme`: 最终访问协议, 如果反代服务启用了 HTTPS 这里也需要改为 `https`
- `--server_host`: 最终访问域名或公网 IP, 不要带端口号

完整启动参数的说明可以使用 `-h` 来查看。另外, `sym.props` 配置文件请用挂卷方式覆盖容器内的配置文件。

细节说明

1. 数据库表会在第一次启动时自动创建, 管理员账号 admin/admin
2. 端口、访问路径等配置请参考 [Latke 配置剖析](#)
3. 生产环境建议使用 NGINX 进行反向代理, 并需要配置好 WebSocket 代理
4. **千万不要**手动修改数据库, 否则可能会引发各种莫名其妙的问题

环境变量

配置文件 `latke.props`、`local.props` 以及 `sym.props` 均支持通过环境变量指定文件路径:

- `latke.props`: `LATKE_PROPS`
- `local.props`: `LATKE_LOCAL_PROPS`
- `symphony.props`: `SYM_PROPS`

常用配置

- 图片上传默认是上传服务器本地, 要使用 [七牛](#)可配置 `symphony.properties` 中的 `qiniu.*` 属性
- 搜索需要使用 [Algolia](#), 配置 `symphony.properties` 中的 `algolia.*` 属性, 也支持本地 Elasticsearch 但需要少量开发, 具体请参考相关代码
- 邮件发送可以配置 3 种渠道: 本地 JavaMail, [SendCloud](#), [阿里云邮件推送](#)。具体请参考配置 `symphony.properties` 中的 `mail.*` 属性
- 用户注册需要验证邮箱, 所以必须先配置好邮件; 也可以通过管理员在后台的用户管理中直接添加用户

完整的配置项说明请参考 [Sym 配置项说明](#) (请注意该文档中提到的**生产环境**配置必须修改, 否则可能存在安全隐患)

NGINX 示例

Sym 的回帖实时推送用了 WebSocket 技术，所以也需要对其配置反代。下面的示例仅供参考：

```
upstream backend_symphony {
    server localhost:8081;
}

map $http_upgrade $connection_upgrade {
    default upgrade;
    ""      close;
}

server {
    listen      80;
    server_name hacpai.com;

    location / {
        proxy_pass http://backend_symphony$request_uri;
        proxy_http_version 1.1;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
    }
}
```

技术资料

- [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](#)

FAQ

部署后界面样式乱了

F12 看下网络请求，然后参考[这里](#)修改配置。

Logo 修改

方法一： `src/main/webapp/js/symbol-defs.js` 中修改 `<symbol id="logo"...</symbol>` 为对应的 `sg`

方法二： `src/main/webapp/skins/classic/header.ftl` 及 `src/main/webapp/skins/mobile/header.ftl` 中修改 `<svg><use xlink:href="#logo"></use></svg>` 为对应的图片

更好的 Markdown 渲染

Sym 默认是使用内建的 [flexmark](#) 进行 md 渲染，可能对有的场景下的渲染效果不是很好。如果你想获得更好的 md 渲染效果请参考[这里](#)。

关于版本升级

1. 参考 [CHANGE_LOGS](#) 进行代码合并
2. 留意 sym.props 配置项的增减
3. 留意 repository.json 数据库表结构变更