

文档编号：EPPDEV-JEE-USAGE-01

---

---

# **EPPDEV-JEE 系统操作手册**

## **V1.6**

---

---

**郝金隆**

2019-09

## 目录

<b>1 文档概述</b>	<b>1</b>
1.1 文档说明	1
1.2 参考文件	1
1.3 保密及知识产权要求	1
<b>2 下载与安装</b>	<b>1</b>
2.1 环境准备	1
2.2 程序下载	1
2.2.1 直接下载打包号的程序文件	2
2.2.2 通过源代码进行安装使用	2
<b>3 使用前的基础配置</b>	<b>2</b>
3.1 数据库环境准备	2
3.2 修改 application.properties 文件	3
<b>4 代码生成</b>	<b>3</b>
4.1 启动代码生成程序	3
4.1.1 windows 下的启动	3
4.1.2 Linux/Mac 下的启动	3
4.2 启动后的基础配置	4
4.3 生成基础代码	5
4.4 库表信息配置	6
4.4.1 新增库表信息	6
4.4.2 自动加载原有库表	9
4.5 库表对应的代码生成	10
<b>5 附录</b>	<b>10</b>
5.1 生成的代码说明	10
5.2 历史版本修订记录	11
5.2.1 V1	11
5.2.2 V1.1	11
5.2.3 V1.2	12
5.2.4 V1.3	12
5.2.5 V1.4	12
5.2.6 V1.4.1	12
5.2.7 V1.5	12

5.2.8 V1.6 . . . . .	12
----------------------	----

# 1 文档概述

## 1.1 文档说明

本文是 EPPDEV-JEE 代码生成系统的使用说明，主要包括：

1. 平台的下载、安装与启动
2. 使用前的基础配置
3. 代码的生成
4. 生成的代码架构说明
5. 旧版本升级说明

## 1.2 参考文件

暂无

## 1.3 保密及知识产权要求

EPPDEV-JEE 代码生成系统是由郝金隆个人开发的开源软件，采用[Anti-996 License V1.0<sup>1</sup>](#)协议对外发布。本文知识产权归属郝金隆个人所有，但是同样采用相同协议对外发布。任何符合该协议的企业、团体和个人，均可在本文的基础上进行二次修改和对外分发。

# 2 下载与安装

## 2.1 环境准备

系统编译运行所必须的环境主要包括：

1. Java 1.8+
2. MySQL/MariaDB 5.5+
3. Maven 3.5+ <sup>2</sup>

## 2.2 程序下载

程序下载有以下两种方式：

---

<sup>1</sup>参见 <https://github.com/996icu/996.ICU/blob/master/LICENSE>

<sup>2</sup>Maven 仅用于编译安装使用

### 2.2.1 直接下载打包号的程序文件

本程序的正式发布地址为：<https://github.com/eppdev/eppdev-jee-pom/releases>，要下载的内容文件包括：

1. eppdev-jee-web-x.y.z.jar<sup>3</sup>：已打包号的程序文件
2. application.properties：主要是一些基础配置文件

上述两个文件下载以后，需要放到同一个文件夹下

### 2.2.2 通过源代码进行安装使用

源代码仓库位于：

- Github: <https://github.com/eppdev/eppdev-jee-pom>
- Gitee: <https://gitee.com/eppdev/eppdev-jee-pom>

任何人都可以从上述两个代码仓库下载并自行编译本系统，以 github 为例：

```
1 git clone https://github.com/eppdev/eppdev-jee-pom.git
2 cd eppdev-jee-pom
3 mvn -Dmaven.test.skip=true package
```

注意：编译安装需要 Java 和 Maven 环境

编译完成后，需要将以下两个文件复制到同一个文件夹下：

1. eppdev-jee-web/target/文件夹下的 eppdev-jee-web-x.y.z.jar 文件
2. eppdev-jee-web/src/main/resources/application.properties 文件

## 3 使用前的基础配置

### 3.1 数据库环境准备

已有数据库环境的可以跳过此章节。若无数据库环境，则需首先需要创建一个数据库 schema，并对其进行权限配置，示例如下：

<sup>3</sup>x.y.z 表示的程序的版本号，具体下载时可以下载最新的版本

```
1 create database test_db charset utf8;
2 grant all privileges on test_db.* to 'eppdev'@localhost identified by '
   eppdev';
3 flush privileges;
```

注：上述 sql 完成了 test\_db 数据库的创建，并将其所有权限赋予了密码为 eppdev 的 eppdev 用户具体的 schema 名称、用户名、密码可以根据实际需要进行修改

## 3.2 修改 application.properties 文件

根据需要修改 application.properties 文件，主要修改项目包括：

1. spring.datasource.url: 数据库连接地址
2. spring.datasource.username: 数据库用户名
3. spring.datasource.password: 数据库连接密码
4. eppdev.author.name: 在生成的代码注释中的作者名称，建议写成自己的名字
5. eppdev.project.path: 要生成的代码所在的目录，写成自己本地的目录绝对地址

针对个人开发，4、5 最后两项可以不在 application.properties 中配置，在平台启动后的界面配置即可。但若是多人协作开发，建议每个人在自己的本地单独配置不同的作者名、工程地址，以避免某一修改配置后对其他人的影响

## 4 代码生成

### 4.1 启动代码生成程序

#### 4.1.1 windows 下的启动

```
1 java -jar -Dfile.encoding=UTF-8 ./eppdev-jee-web-x.y.z.jar
```

注：此处需通过指定 file.encoding 以避免生成代码的乱码问题

#### 4.1.2 Linux/Mac 下的启动

Linux 和 Mac 下可以首先将下载的 jar 包转换为可执行程序，后续只需要直接执行即可

```
1  chmod a+x ./eppdev-jee-web-x.y.z.jar
2  ./eppdev-jee-web-x.y.z.jar
```

4.2 启动后的基础配置

系统启动后通过浏览器打开<http://localhost:8080/><sup>4</sup>，即可访问本系统，界面如下图所示：



图 1. 进入后的首页

点击顶部菜单中的初始化配置链接，即可进入初始化配置页面，进行初始化配置，如下图所示：



图 2. 初始化配置页面

主要需要修改的配置包括：

<sup>4</sup>若在 application.properties 中配置了不同的端口，则需要对应修改访问地址

- PROJECT\_NAME: 工程名称
- PROJECT\_PATH: 要生成的代码所在位置
- BASIC\_PACKAGE\_NAME: 要生成的代码的基础包名
- AUTHOR\_NAME: 生成的代码注释中的 author
- COMPANY\_NAME: 公司名称，用于展示在注释中
- LICENSE\_DESC: 软件许可说明，展示在注释中

更多的配置项说明，参见界面上的说明。

注：针对协作开发，建议直接在 application.properties 文件中修改配置 eppdev.author.name，eppdev.project.path，则 PROJECT\_PATH、AUTHOR\_NAME 在 web 页面不起作用，以区分不同的使用者

4.3 生成基础代码

点击 web 页面中的工程初始化菜单，即可进入工程初始化界面，如下图所示：



图 3. 工程初始化页面

在界面中可以初始化通用代码，具体初始化内容包括：



- 工程基础配置，包括 pom.xml, application.properties, Application.java, .gitignore 等文件
- 基础类，针对每个表要生成的代码的父类
- 工具类，主要是生成的代码所依赖的一些工具类
- 测试配置，主要是要生成的 Test 代码所依赖的一些工具类

在此页面可以选择一次性全部生成，也可以选择分类生成，还可以单个文件依次生成。

注意在生成相应的代码前，可以先预览以下具体要生成的内容，确认一些具体的代码路径，避免错误出现

## 4.4 库表信息配置

### 4.4.1 新增库表信息

#### 4.4.1.1 添加库表

点击最新库表信息菜单，选择新增表按钮，即可进入库表添加库表页面，如下图所示：



图 4. 新增库表页面

主要要填写的内容包括：

- 表名：即要生成的库表的名称<sup>5</sup>，根据规则会生成对应的 Java 对象名称
- 表说明：用于生成表注释
- 模块名：用于将生成的代码区分到不同的 java 包中
- 实体名：默认一般不填，根据规则<sup>6</sup>自动生成

根据提示填写相关内容以后，点击提交按钮即可完成库表的添加，添加库表的过程中，系统会自动创建 id、create\_date、create\_by、update\_date、update\_by、del\_flag、remarks 这六个技术字段

<sup>5</sup>建议全部用小写字母，下划线分割，如 test\_table

<sup>6</sup>系统会自动将表名转换成驼峰形的 Java 类名，如 test\_table 会自动对应为 TestTable

4.4.1.2 新增字段

创建完成库表成功后，页面会自动转到库表详情页面，如下图所示：



图 5. 表详情页面

在此页面中可以查看并添加字段，配置库表的索引，进行代码生成，查看修订记录。在字段列表中点击添加链接，即可进入新增字段页面，具体如下图所示：

🏠 首页

初始配置

工程初始化

最新库表信息列表

历史版本

EPPDEV-JEE

[返回](#)

添加字段

1. 基础属性

字段名:

请输入字段名

是否主键:

否

是否业务主键:

否

字段类型:

int

字段长度:

请输入整数

字段说明:

字段说明

实体名:

不填采用默认规则自动创建

属性类:

不填采用默认规则自动创建

列排序:

字段排序

备注:

图 6. 添加字段页面

在此页面可以配置内容包括：

- 1. 基础属性：包括字段名、字段类型、字段长度等信息
- 2. 查询条件：主要影响到未来是否可以按本字段的相关信息查询
- 3. GroupBy 结果：主要影响到未来生成的 listGroupBy 方法中是否有本字段对应的信息

相关信息填写完成后，提交保存即可。

4.4.1.3 配置索引

库表详情页面，索引列表中点击添加链接，即可配置库表的索引，如下图所示：

首页 初始配置 工程初始化 最新库表信息列表 历史版本 EPPDEV-JEE

返回

添加索引

索引字段:

字段选择:

id

create\_date

update\_date

create\_by

update\_by

remarks

del\_flag

名称字段: idx\_test\_table

重置

提交

图 7. 添加索引页面

在此页面中依次选择要创建的索引字段，选择提交后即可完成索引的配置

4.4.2 自动加载原有库表

4.4.2.1 加载物理表

系统还可以自动读取平台已有的库表信息，在工程初始化页面点击加载最新物理表链接，即可进入物理表加载页面，如下图所示：

首页 初始配置 工程初始化 最新库表信息列表 历史版本 EPPDEV-JEE

物理表信息

加载最新物理表

序号	表名	状态	操作
1	_eppdev_column	-	加载最新物理表
2	_eppdev_conf	-	加载最新物理表
3	_eppdev_index	-	加载最新物理表
4	_eppdev_table	-	加载最新物理表
5	_eppdev_table_log	-	加载最新物理表
6	_eppdev_user	-	加载最新物理表
7	_eppdev_version	-	加载最新物理表

图 8. 加载物理表页面

在此页面可选择加载所有的物理表，也可以选择单个表进行加载。

4.4.2.2 加载后的配置

为保证加载后库表对应的代码的可用性，一般需要对每个表的信息进行配置，主要配置内容是每个字段对应的信息，包括

- 基础信息：主要确认自动映射的结果是否正确
- 查询条件和结果：主要用于配置具体要生成的内容，包括本字段是否要生成相关查询条件和 GroupBy 结果

## 4.5 库表对应的代码生成

生成对应的代码，可以采用两种模式来生成：

1. 在库表详情页面，可以生成针对本表的所有代码，可以选择一次性生成所有代码，也可以选择仅生成单项代码
2. 在历史版本页面，可以选择生成所有表对应的代码

## 5 附录

### 5.1 生成的代码说明

具体会生成的内容包括：

- 每个表对应的生成文件（以 sys\_user 表为例）
  - \_SysUser.java：自动生成 Java 类，继承 BasicEntity 类，实现各个字段到 Java 属性的映射
  - SysUser.java：实际使用的 Java 实体类，继承自 \_SysUser，自定义属性可以在此添加
  - \_SysUserDao.java：自动生成的 Dao 文件，继承自 BasicDao，实现各基础调用方法
  - SysUserDao.java：实际使用的 Dao 文件，继承自 \_SysUserDao，自定义 DAO 方法可以在此定义
  - \_SysUserParam.java：自动生成的查询条件类，根据库表中各个字段的配置，自动生成
  - SysUserParam.java：实际使用的查询条件类，继承自 \_SysUserParam，自定义的条件可以在此添加
  - \_SysUserService.java：自动生成的 Service 类，继承自 BasicService，实现了 exists()、getDao() 方法
  - SysUserService.java：使用使用的 Service 类，需自定义 customInit 方法，有需要增加的方法可以在此处理
  - SysUserController.java：对应的 RestController，实现了基础的增删改查操作
  - sys\_user.sql：基础的库表定义文件
- 基础类类文件
  - BasicDao.java: 定义了基础的 Dao 方法
  - BasicEntity.java：定义了 OR 映射实体类的基础字段和方法
  - BasicService.java：定义了 Service 的基础方法
  - BasicParam.java: 定义了条件类的基础属性和方法，包括排序条件的构建等

- BasicController.java : 定义了 Controller 的基础方法，包括增删改查等操作
- 一些工具类
  - StringUtils.java
  - JsonUtils.java
  - PasswordUtils.java
  - RandomUtils.java
- 基础配置文件
  - pom.xml : maven 工程的基础配置
  - application.properties : springboot 配置文件
  - Application.java : springboot 的启动文件
  - .gitignore : git 配置文件
- 测试相关代码
  - ApplicationTest.java
  - applicataion\_test.properties

对于每个表，之所以在 Entity、Param、Dao、Service 和 Mapper 文件都要分两个，是为了区分自动生成以后要根据数据库配置自动更新的代码文件和可以手工修改的文件。有下划线的文件都是后续自动生成的时候要自动修改覆盖的文件，无下划线的是可以手工修改的，后续的自动代码生成不会覆盖。

## 5.2 历史版本修订记录

### 5.2.1 V1

2019 年 1 月第一次正式发布，主要完成内容：

- 在很久以前写的代码生成器基础上，采用 FreeMarker 进行了代码的改造
- 将可以个性化定制修改的代码和不能手工修改（重新生成后会覆盖）的代码的分离
- 将查询参数从映射的实体类中分离，实现查询参数的可定制化

### 5.2.2 V1.1

2019 年 2 月完成的小版本优化，主要完成内容包括：

- 针对单机版实现本地配置覆盖服务器通用配置，支持多人同时进行代码生成

### 5.2.3 V1.2

2019 年 3 月完成的 bugfix，主要修改内容：

- 针对插入或修改前的逻辑主键判断，只查询一条数据，避免因逻辑主键为空导致获取到大量的数据问题
- 增加 996ICU License

### 5.2.4 V1.3

2019 年 5 月完成的小版本升级，主要内容：

- 自动生成 Test 相关内容

### 5.2.5 V1.4

2019 年 5 月完成的小版本升级，主要内容：

- 增加单个物理库表更新加载功能

### 5.2.6 V1.4.1

2019 年 5 月完成的 bugfix，主要内容：

- 解决 \_minParam 参数中的 set 方法命名错误问题

### 5.2.7 V1.5

2019 年 8 月完成的小版本升级，主要内容：

- 增加了库表修改日志的记录
- 生成的代码中增加了修改日志的注释

### 5.2.8 V1.6

2019 年 9 月完成的小版本升级，主要内容：

- 增加自动生成 listGroupBy 方法的相关操作