

# Daas-Start-Kit 开发快速上手



成都双链科技有限责任公司

成都双链科技有限责任公司

## 修订历史

修订日期	变更类型	变更人	说明
2019.10.18	创建	李亚青	创建初始文档。
2019.10.29	更改	刘利	增加 Docker 开发配置
2020.01.02	变更	李亚青	1. 迁移文档至 WPS; 2. 调整注册链接; 3. 增加中台翻译调整接口说明;

# 目录

Daas-Start-Kit 开发快速上手.....	1
修订历史.....	2
目录.....	3
一、 简介.....	5
二、 快速开始.....	5
2.1 下载 Daas-Start-Kit 工具包.....	5
2.2 获取 daas 服务 token.....	5
2.3 定义业务模型.....	6
2.4 生成代码.....	6
2.5 备注.....	7
三、 本地编译运行.....	7
3.1 本地开发环境准备要求.....	7
3.2 配置数据库基础设施.....	8
3.3 初始化数据库并导入 Sample 数据.....	9
3.3.1 使用 MySQL WorkBench 数据库脚本信息.....	9
3.3.2 使用 Docker 导入数据库脚本信息.....	9
3.4 启动后台系统.....	10
四、 启动中台系统.....	10
4.1 环境要求.....	10
4.2 环境配置说明.....	11
4.3 构建中台系统前端.....	12
4.4 启动中台前端系统.....	12

4.5 调整中台显示文本的翻译.....	13
五、 我有疑问.....	13

双链科技

# 一、简介

Daas-Start-Kit 是成都双链科技（后续简称为“双链科技”）提供的一套自动代码生成服务。无需开发人员，只要产品经理&需求分析师就可以轻松完成业务系的基础代码开发并构建业务中台系统及中台系统用户界面。

## 二、快速开始

### 2.1 下载 Daas-Start-Kit 工具包

通过 Git 工具可以从 GitHub 上下载 Daas-Start-Kit 工具包：

```
git clone https://github.com/doublechaintech/daas-start-kit.git
```

### 2.2 获取 daas 服务 token

点击链接[注册 daas 自动代码生成服务](#)并获取 daas 代码生成服务的 token（因为当前阶段 daas 自动代码生成服务是通过命令行运行）。

完成账号注册后，下载 daas.tok 的 token 文件，并将文件保存到 daas-start-kit 的根目录中替换 daas.tok 文件。

daas 服务注册链接：

<https://daas.doublechaintech.com/daas/daasService/login/>

## 2.3 定义业务模型

进入 daas-start-kit 的根目录，您可以看到我们在系统中默认提供的几个业务模型样例：bank.xml 、 his.xml 以及 retailscm.xml 模型文件。

您可以参考我们的模型样例，创建自己的模型文件，并定义自己的业务模型数据。关于如何定义业务模型文件，请参考文档[《DaaS 概要与建模》](#)

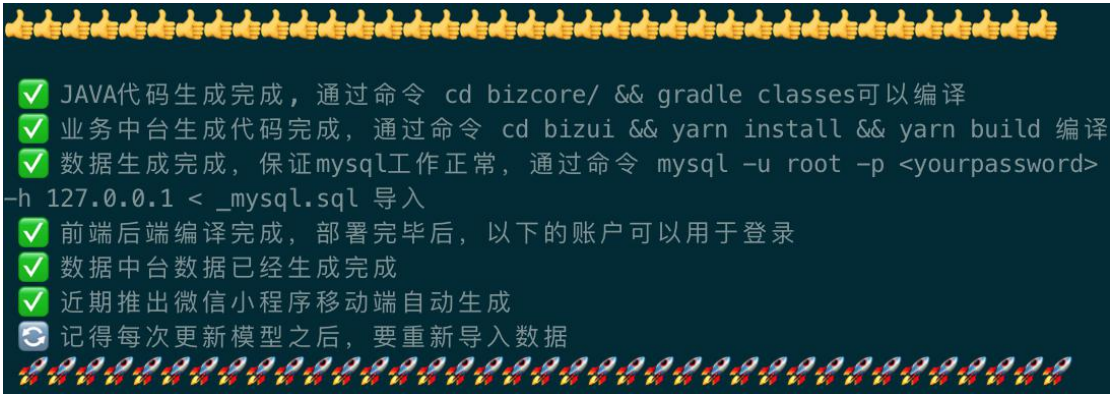
## 2.4 生成代码

在 daas-start-kit 的根目录下执行下面指令，生成代码：

```
./gencode bank.xml all ../output-project-name/.
```

命令	说明
./gencode	daas 代码生成指令  Windows 系统下，则是 ./gencode.bat  MacOS 以及 Linux 则是 ./gencode
bank.xml	指定你要进行代码生成的模型文件
all	代码生成指令的参数，用来控制是生成运算的结果范围
../output-project-name/.	生成后的代码所存放的路径

代码生成完成以后，您可以看到以下提示：



到目前位置，您已经完成了您项目的所有基础业务代码开发。

## 2.5 备注

./gencode.sh 是通过执行本地的一个 jar 包, 将模型文件提交到双链科技的代码生成服务系统中, 如果运行失败, 您需要安装 JDK 8。

JDK 8 下载地址:

[https://www.oracle.com/technetwork/java/javase/downloads/index.ht](https://www.oracle.com/technetwork/java/javase/downloads/index.html)

ml

## 三、本地编译运行

既然代码都已经生成了, 那么我们可以跑一把, 验证下我们的功能是否正常呢?

答案是: 可以的, 因为我们连演示的数据都帮您生成好了。

### 3.1 本地开发环境准备要求

工具名称	说明
Java JDK 8.0	必选

Redis >= 3.0	必选
gradle	必选
MySQL >= 8.0	必选
Docker	可选
VS Code	免费好用的代码编辑器

### 3.2 配置数据库基础设施

在生成代码的目录下，在 bicore 目录下找到 infra.properties 文件，修改数据库和 Redis 的配置信息。

文件路径如：

/output-project-name/bizcore/WEB-INF/bank\_custom\_src/META-INF/infra.properties

配置信息如下：

```
app.database.jdbcURL=jdbc:mysql://mysqlServerHost:3306/bank?characterEncoding=utf8&autoReconnect=true&useSSL=false&serverTimezone=Asia/Shanghai
app.database.driverClass=com.mysql.cj.jdbc.Driver
app.database.dataSourceClassName=com.mysql.cj.jdbc.MysqlDataSource
app.database.username=mysqlUserName
app.database.password=mysqlPassword
app.database.pool.initialSize=2
app.database.pool.maxActive=30
app.redis.host=redisServerHost
app.redis.port=6379
app.redis.timeout=2000
app.redis.password=redisPassword
app.redis.database=0
```



## 3.3 初始化数据库并导入 Sample 数据

在我们提供 MySQL 脚本中包含了：您当前开发项目的 MySQL 初始化脚本以及根据您建模过程中提供的样例值生成了样例数据信息。

您可以在 bizcore 中的下面路径找到 MySQL 的初始化脚本。

```
bizcore/WEB-INF/bank_core_src/META-INF/bank_mysql.sql
```

将该 SQL 脚本导入到您的数据库中即可。

注：您可以使用

### 3.3.1 使用 MySQL WorkBench 数据库脚本信息

使用 MySQL WorkBench 或者 其他数据库客户端用户，可以用文本编辑器打开 .sql 文件，复制粘贴到 MySQL WorkBench 中运行即可。

### 3.3.2 使用 Docker 导入数据库脚本信息

使用 Docker 的用户，我们提供了 Docker Compose 脚本，您可以通过 docker-compose 来运行 MySQL 并导入 sql 数据和运行 Redis 服务。

在 daas-start-kit 的根目录下，运行下面指令。

```
cd docker

#启动 mysql 和 redis

docker-compose -f app.yml up -d

#等数据库起来以后，可以导入数据

docker exec -i docker_daas-mysql_1 mysql -uroot -p0254891276
```

```
--default-character-set=UTF8

< ../bizcore/WEB-INF/bank_core_src/META-INF/bank_mysql.sql
```

### 3.4 启动后台系统

前提条件：已经完成数据库初始化并启动数据库运行和启动 Redis 运行

在 bizcore 目录中，使用下面的 gradle 命令启动后台系统。

```
gradle bootRun
```

## 四、启动中台系统

### 4.1 环境要求

工具名称	说明
Node.js > 10.0	必选
谷歌 Chrome 浏览器 > 79.0	必选
Npm > 6.8.0	必选
Yarn > 1.9	推荐使用
VS Code	免费好用的代码编辑器
编译计算机内存 > 4GB	必选

## 4.2 环境配置说明

由于生成的系统根据模型的复杂程度，可能会比较庞大，建议编译的计算机至少有 4~6GB 的内存，并设置额外的两个 node.js 参数

- ✓ `NODE_OPTIONS=--max-old-space-size=10230`，增加编译内容， 或者安装并且下载 `increase-memory-limit`
- ✓ `PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=1`，不下载 chromium 防止下载时间过长

以 MacOS 为例，请在 `~/.bash_profile` 中添加

```
export NODE_OPTIONS=--max-old-space-size=10230
export PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=1
```

如果不设置环境变量，可能会出现如下异常

```
==== JS stack trace =====
0: ExitFrame [pc: 0x1b1e8aadbe1d]
1: StubFrame [pc: 0x1b1e8d56aba2]
Security context: 0x2be87309e6e9 <JSObject>
2: getOptions(aka getOptions) [0x19d718b916d1]
[/home/philip/githome/retailscm-biz-suite/bizui/node_modules/acorn/dist/acorn.js:403] [bytecode=0x19e0b9bbeaa9
offset=85](this=0x1a01d14026f1 <undefined>,opts=0x0b7c125f33f9
<Object map = 0x27e6b3c63c59>)
3: new constructor(aka Parser) [0x41cc263b091] [/home/philip/g...
```

```
FATAL ERROR: Ineffective mark-compacts near heap limit Allocation
failed - JavaScript heap out of memory

1: 0x8fa0c0 node::Abort() [/usr/bin/node]
```

添加设置完环境变量后，运行如下指令，让环境变量立即生效

以 MacOS 为例，运行下面指令

```
source ~/.bash_profile
```

## 4.3 构建中台系统前端

在生成的代码的根目录下，执行下面指令

```
cd bizui
npm install
```

如果您安装了 yarn（推荐使用 Yarn 来替代 npm），在生成代码的根目录下执行下面指令。

```
cd bizui
yarn
```

## 4.4 启动中台前端系统

在 bizui 目录下面，执行下面指令就可以在浏览器中访问中台前端界面啦。

```
npm start
```

如果安装了 yarn，则可以使用。

```
yarn start
```

如果一切顺利，则可以在 chrome 中访问 <http://localhost:8000>

默认登录账号 SU000001/admin123

## 4.5 调整中台显示文本的翻译

在生成代码的目录下通过下面路径

```
/bizui/public/
```

找到下面两个文件

locale\_core.js

locale\_custom.js

参考 `locale_core.js` 中的 `coreLocaleMessage` 的键值对，在 `locale_custom.js` 中的 `customLocaleMessage` 添加需要替换翻译的键值对，就可以完成翻译文本的替换。

## 五、我有疑问

使用过程中，您有什么疑问，您有什么新的需求，欢迎到 github 上提交 Issue。

<https://github.com/doublechaintech/daas-start-kit/issues>

DaaS 的介绍文档可从这里获取：<https://kdocs.cn/l/sS9sjXdPo?f=502>

DaaS 的后端开发文档：<https://kdocs.cn/l/sUdwkkyZD?f=502>