

Flyway 数据管理工具

编辑

江易(胡一航)

发布于 :2022-11-21 10:15

浏览量 :85

分享至 [POPO 眼界大开](#)

*本内容仅代表个人观点，不代表网易，仅供内部分享传播，不允许以任何形式外泄，否则追究法律责任

本文仅面向以下用户开放，请注意内容保密范围

查看权限：Omega 工作室

简单介绍

Flyway 是一款开源的数据库版本管理工具。它可以很方便的在命令行中使用。在项目或产

品中，很难一开始就把业务理清楚，把数据库表设计好，因此数据表也会在迭代周期不断

迭代。Flyway 官方文档：[官方文档](#)

安装包和使用文档说明在附件中

为什么要使用 flyway

在多人开发的项目中，我们都习惯了使用 SVN 或者 Git 来对代码做版本控制，主要的目的

就是为了解决多人开发代码冲突和版本回退的问题。

其实，数据库的变更也需要版本控制，在日常开发中，我们经常会遇到下面的问题：

自己写的 SQL 忘了在所有环境执行。

别人写的 SQL 我们不能确定是否都在所有环境执行过了。

有人修改了已经执行过的 SQL，期望再次执行。

需要新增环境做数据迁移。

每次发版需要手动控制先发 DB 版本，再发布应用版本。

其它场景。

有了 flyway，这些问题都能得到很好的解决。

flyway 是如何工作的

flyway 工作流程如下：

项目启动，应用程序完成数据库连接池的建立后，Flyway 自动运行。

初次使用时，flyway 会创建一个 flyway_schema_history 表，用于记录 sql 执行记录。

Flyway 会扫描项目指定路径下(默认是 classpath:db/migration)的所有 sql 脚本，与

flyway_schema_history 表脚本记录进行比对。如果数据库记录执行过的脚本记录，与项目

中的 sql 脚本不一致，Flyway 会报错并停止项目执行。

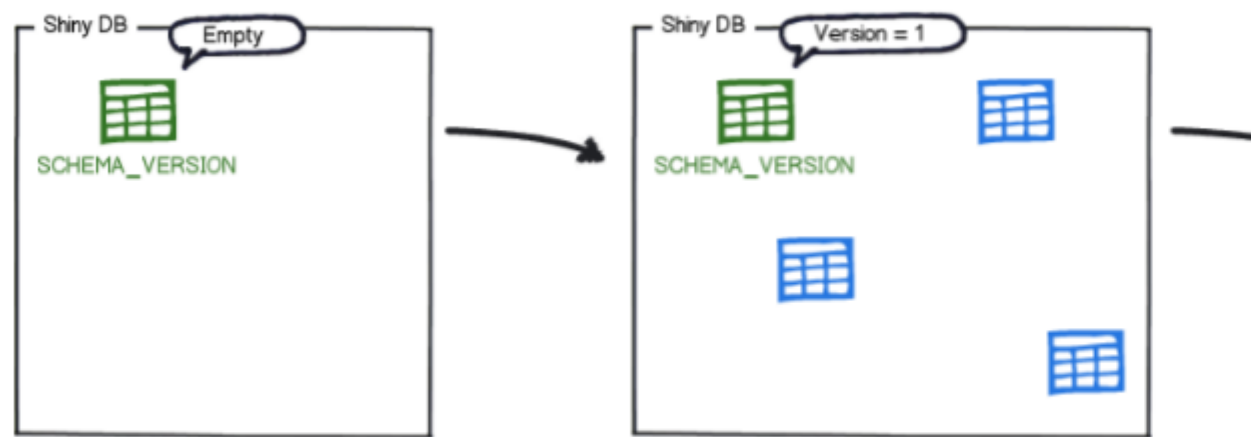
如果校验通过，则根据表中的 sql 记录最大版本号，忽略所有版本号不大于该版本的脚

本。再按照版本号从小到大，逐个执行其余脚本。

2) 版本信息记录

Flyway更新的时候会指定更新哪个数据库，然后去对应的目录寻找sql文件来进行更新。

Flyway会自动在数据库中新建一个名为"flyway_schema_history"的表（备注：Flyway V5.0之前的版本，下次会接着按照sql版本的顺序往下更新。



As each migration gets applied, the schema history table is updated accordingly:

flyway_schema_history

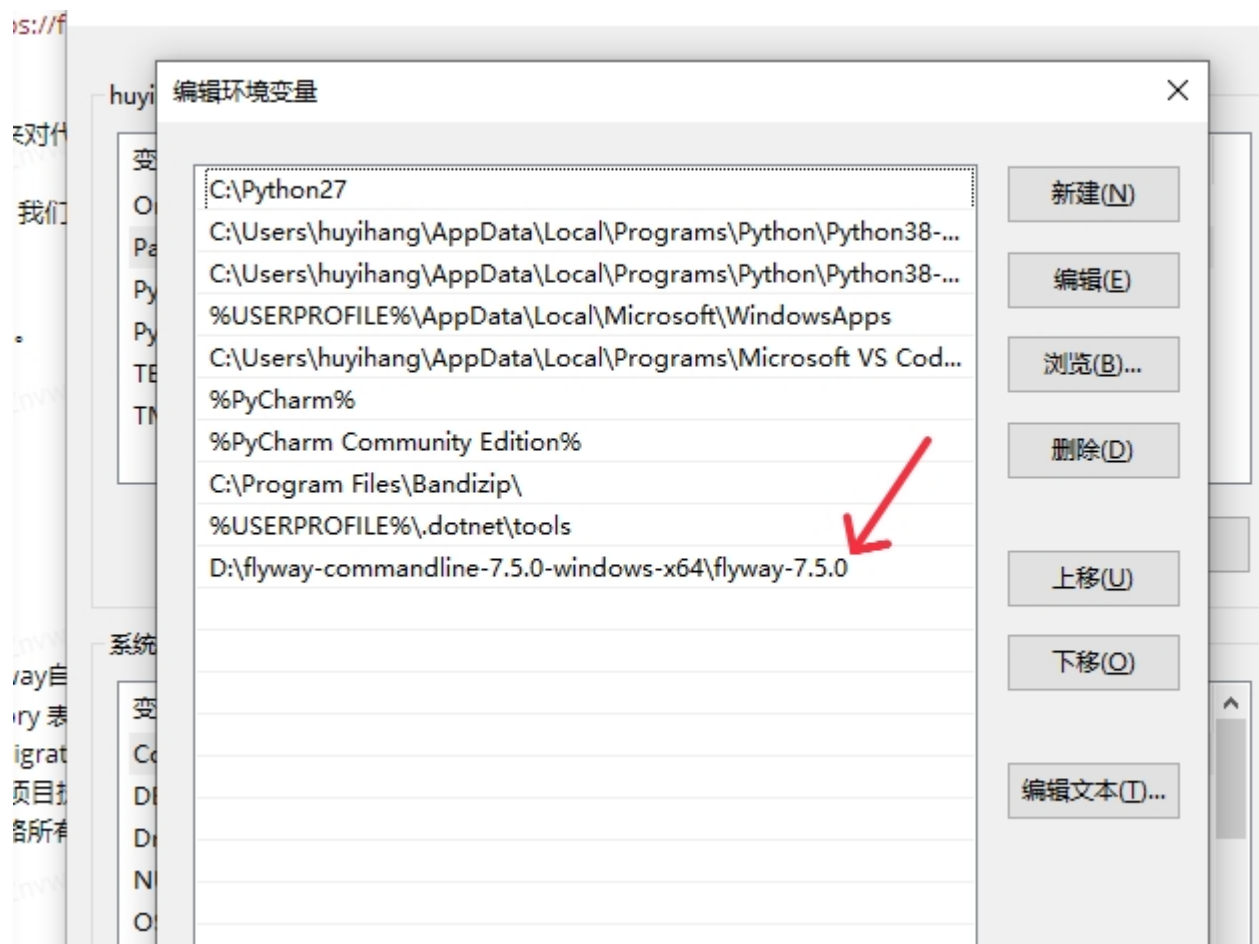
installed_rank	version	description	type	script	checksum	installed_by	
1	1	Initial Setup	SQL	V1__Initial_Setup.sql	1996767037	axel	2
2	2	First Changes	SQL	V2__First_Changes.sql	1279644856	axel	2

如何在本地使用 flyway

当然是先下载 flyway 安装包已放在附件中

设置 path

下载完之后第一步在系统环境变量中设置 path



配置默认配置文件

这一步可以配置也可以不配置 不配置的话可以在命令行中直接指定 但是我们这里为了更

加方便就配置一下默认配置 主要是数据库连接这些默认配置

配置文件所在位置

共享 查看

此电脑 > 新加卷 (D:) > flyway-commandline-7.5.0-windows-x64 > flyway-7.5.0 > conf

	名称	修改日期	类型	大小
	flyway.conf	2022/11/18 13:29	CONF 文件	22 KB

配置文件中主要配置

```
flyway.url=jdbc:mysql://localhost:10061?serverTimezone=GMT&character_set_server=utf
```

```
8mb4
```

```
flyway.user=xxxxx
```

```
flyway.password=xxxxxxxxx
```

下面是我的配置 我们项目编码格式是 utf8mb4

```
# Redshift*          : jdbc:redshift://<host>:<port>/<database>
flyway.url=jdbc:mysql://localhost:10061?serverTimezone=GMT&character_set_server=utf8mb4

# Fully qualified classname of the JDBC driver (autodetected by default based on flyway.url)
# flyway.driver=

# User to use to connect to the database. Flyway will prompt you to enter it if not specified
# connection is not using a password-less method of authentication.
flyway.user=root

# Password to use to connect to the database. Flyway will prompt you to enter it if not specified
# connection is not using a password-less method of authentication.
flyway.password=123456

# The maximum number of retries when attempting to connect to the database. After each failed
```

注意编码格式和时区

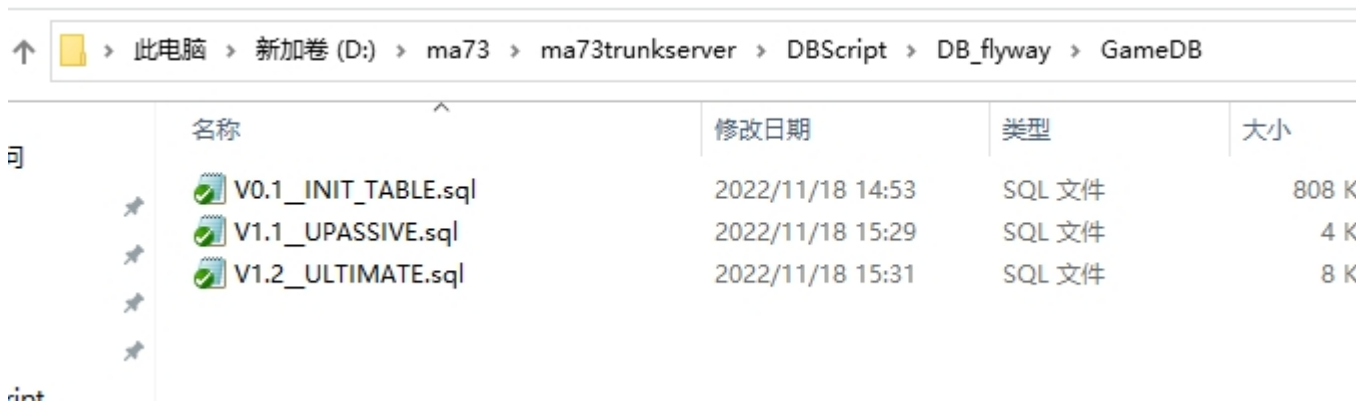
配置 SQL 文件目录

这一步 我将 sql 文件目录放在了 SVN 的 <https://svn->

[ma73.gz.netease.com/svn/code/trunk/ServerSB/DBScript/DB_flyway](https://svn-ma73.gz.netease.com/svn/code/trunk/ServerSB/DBScript/DB_flyway) 目录之下



根据名称在对应的目录下放相应的 SQL



SQL 文件里的语法就不叙述了

这边要注意的是 SQL 文件的命名

必须严格按照本规则命名，否则将会无法更新。

格式：

V[Version][Description].sql

以大写的 V 开头，小写的.sql 结尾；

Version: 数字和小数点组成的版本号，不能有字母！！！按大小顺序递增。例如：

1/2/3/3.1/3.2/3.2.1/3.2.2 等

Description: 对本 sql 文件的描述信息。

Version 和 **Description** 之间用两个下划线""隔开。 注意这里是两个——线

样例:

V1__initial.sql

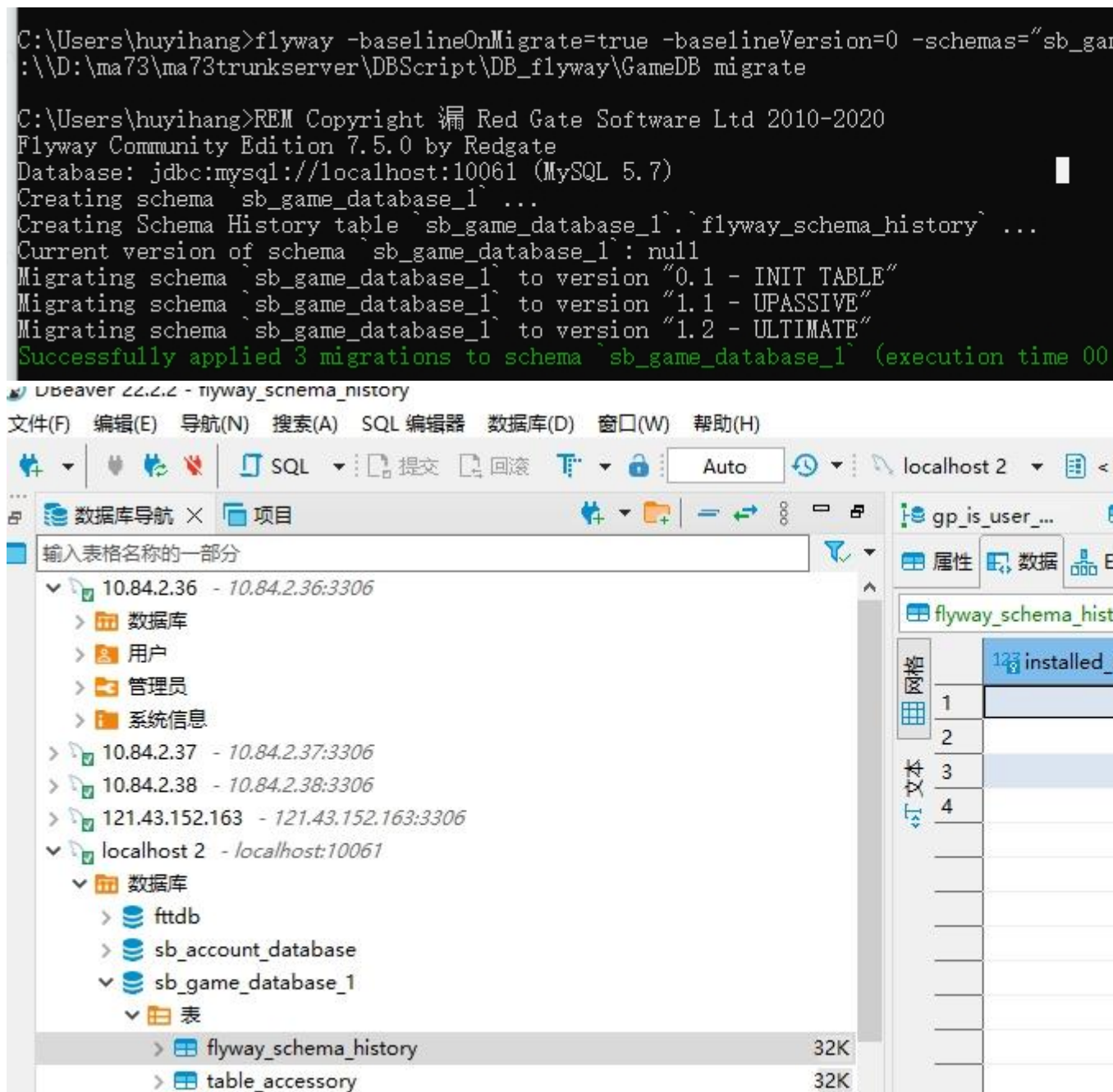
V2.1__first_changes.sql

V3.1.1__add_tables.sql

命令行使用

拿创建 sb_game_database_1 举例要使用的命令行是

```
flyway -baselineOnMigrate=true -baselineVersion=0 -schemas="sb_game_database_1" -  
locations=filesystem:\D:\ma73\ma73trunkserver\DBScript\DB_flyway\GameDB migrate
```



这样就能看到数据库脚本执行成功了 后面有新的数据库脚本 就提交在对应目录下 在执行

命令行就可以了

参数说明

1、-baselineOnMigrate=true -baselineVersion=0: 表示如果当前 DB 未使用 Flyway 的

话, 以当前 DB 版本为基线版本 (Version=0), 然后再继续更新 DB。适用于有产

品中途迁移使用 Flyway 的情况。

2、-schemas: 数据库名

3、-locations: sql 目录

4、migrate: 表示是对 DB 进行 migrate 更新

补充知识

flyway 执行 migrate 必须在空白的数据库上进行，否则报错。

对于已经有数据的数据库，必须先 baseline，然后才能 migrate。

clean 操作是删除数据库的所有内容，包括 baseline 之前的内容。

尽量不要修改已经执行过的 SQL，即便是 R 开头的可反复执行的 SQL，它们会不利于数据迁移。

当需要做数据迁移的时候，更换一个新的空白数据库，执行下 migrate 命令，所有的数据库更改都可以一步到位地迁移过去。