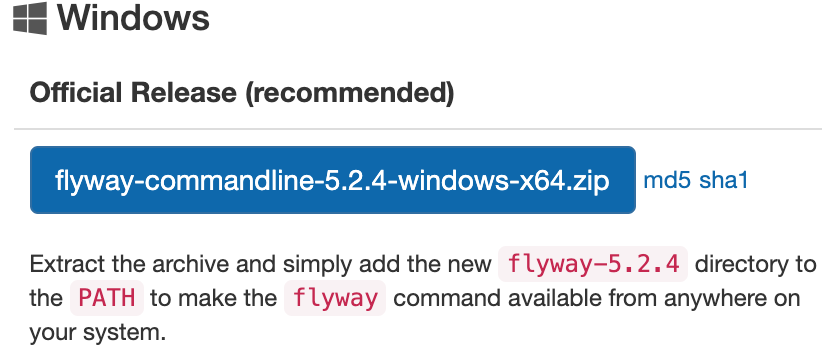
命令行使用flyway方法

pre-work：首先需要一个数据库软件，用于被管理。现在下载的是mysql。

ps：现在 使用的方法是命令行方法。

Working：

1. 使用之前需要下载下面网站中的链接，下载路径需记录，之后会用命令行进入此目录打开。<https://flywaydb.org/documentation/commandline/>



1. 在配置文件里配置信息，主要有四项：

flyway.url=jdbc:mysql://localhost:3306 #可以更改为不同的数据库

flyway.user=root #账户名

flyway.password=123456 #自己设置的密码

flyway.schemas=default #表单的名称，没有则自行创建

上述均为需修改项，具体修改方式可以参考配置文件。

配置文件名称为flyway.conf，一般在flyway-5.2.4/conf文件夹下。

文件配置好，数据库就可以连接了。

1. 尝试版本控制。

3.1 首次migration。在flyway-5.2.4//sql文件夹下放入第一个sql文件，名称V1\_\_Create\_person\_table.sql，内容为create table PERSON (

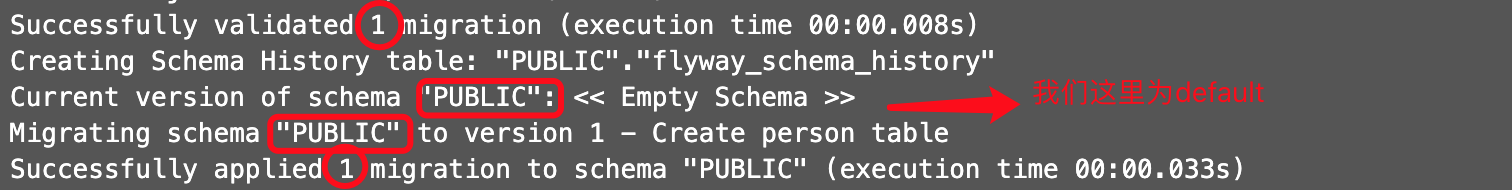
ID int not null,

NAME varchar(100) not null);

在命令行中进入flyway



进入后输入 flyway migrate 命令，会出现以下提示信息。



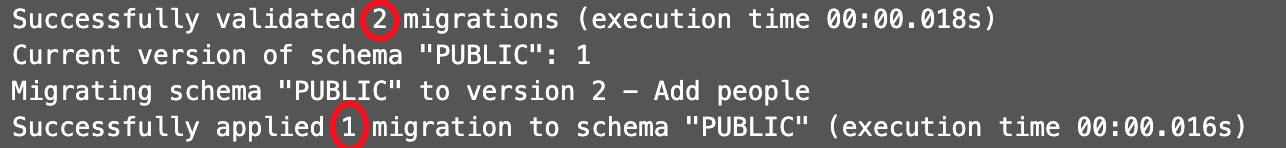
* 1. 第二次migration。在flyway-5.2.4//sql文件夹下放入第二个sql文件，名称V2\_\_Add\_people.sql，内容为

insert into PERSON (ID, NAME) values (1, 'Axel');

insert into PERSON (ID, NAME) values (2, 'Mr. Foo');

insert into PERSON (ID, NAME) values (3, 'Ms. Bar');

在命令行中使用flyway migrate命令，完成第二次migration。给出的提示信息大致如下：



**Attention：在sql文件夹下的sql文件必须遵循以下命名规则。**

SQL脚本的命名规则

前缀：V用于版本化（可配置）， U用于撤消（可配置）和 R用于可重复迁移（可配置）

版本：带点或下划线的版本可以根据需要分开多个部分（不适用于可重复的迁移）

分隔符 :( \_\_两个下划线）（可配置）

描述：下划线或空格分隔单词

后缀：.sql（配置）



**flyway命令的种类（6种）**

|  |  |  |
| --- | --- | --- |
|  | 命令 | 功能 |
|  | flyway migrate | 此命令会自动检查数据库脚本是否有变化，如果有变化，则执行脚本，更新数据库版本，如果数据库初始状态是空库，则会自动创建schema\_version 表，用于存储数据库操作的版本记录，只要数据库脚本有变化，都需要执行此命令。 |
|  | flyway clean | 清除schema\_version中记录所有表结构，视图，存储过程，函数以及所有的数据等都会被清除。 |
|  | flyway info | 打印schema\_version中记录信息 |
|  | flyway validate | Validate是指验证已经Apply的Migrations是否有变更，Flyway是默认是开启验证的。 |
|  | flyway baseline | Baseline 是指数据库非空状态下使用flyway首先执行的命令，用于创建schema\_vision表。 |
|  | flyway repair | Repair会修复Metadata表的错误，通常有两种用途：  1、移除失败的Migration记录，该问题只是针对不支持DDL事务的数据库。  2、重新调整已经应用的Migratons的Checksums值，比如：某个Migratinon已经被应用，但本地进行了修改，又期望重新应用并调整Checksum值。尽量避免这样操作。 |

使用Java API配置一个flyway项目：

1. 新建一个maven项目（使用QuickStart即可）。
2. 在pom.xml中配置依赖项，加入org.flywaydb/mysql-connector-java　具体配置代码如下：

<dependency>  
 <groupId>org.flywaydb</groupId>  
 <artifactId>flyway-core</artifactId>  
 <version>5.2.4</version>  
</dependency>  
<dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.15</version>  
</dependency>

其中版本号，需要按照具体情况做出改动。

1. 在项目下的src/main/java/（包名/）目录下新建一个文件，在此文件中创建Flyway的对象，及使用flyway的方法，使用前需要import org.flywaydb.core.Flyway；

Flyway flyway = Flyway.*configure*().dataSource("jdbc:mysql://localhost:3306/public", "root", "123456").load();

flyway.migrate();

1. 所有的需要进行数据库版本迁移的文件仍然放在src/main/resources/db/migration目录下，文件需遵循命名方法。
2. 运行flyway方法所在的java程序即可。

Flyway使用的几个测试及注意事项

1. 要执行的SQL语句语法错误（比如在不该加分号的地方加了分号）：程序报错，停止运行，schema-history表会出现在数据库中，但是该表中success一项值为0【这是错误记录】；重新改正之后，无法执行迁移，因为schema-history已经存在记录，需使用flyway.repair()命令修复（删除）schema-history中的错误项，后才能使用flyway.migrate()迁移成功。
2. 一个表中有多个insert记录，其中除第三条错误外，其余均正确，是否可以将其他条成功insert？ 不可以，flyway的迁移是以一个sql文件为一个事务的，在一个文件的全部内容均可以正确执行之后，才会commit。不可以自动回滚，需要repair

\*具体可见TransactionTemplate.java类中的public <T> T execute(Callable<T> transactionCallback)函数，在flyway5.2.4版本中为72-75行。

1. db/migration文件夹下的文件是按照版本号排序并执行的，而不是字符串排序。
2. Undo功能只有企业版和专业版才有，无法测试

执行sql语句的方法

1. 现有的跑通的方法有两种，一种是根据文件所在目录及名称直接加载sql文件资源并运行，另一种是放在db/migration目录下，通过类路径加载运行sql文件（类比flyway的方法）；
2. 两种方法分别写在两个java文件中，分别是AppWithClassPath.java和AppWithFilePath.java