



黑客派

数据库版本管理工具 - flyway

作者: [abinbinge](#)

原文链接: <https://hacpai.com/article/1486463037167>

来源网站: 黑客派

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

```
<p></p>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
</script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h3 id="写在前面">写在前面</h3>
<p>在日常开发中，经常遇到数据库变更，字段增减调整，需要维护多个不同数据库时是十分痛苦的情。代码可以使用 SVN，Git 等工具进行管理，那么数据库的版本可以使用 Flyway 一款小工具，就以大大简化这方面的工作。<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fflywaydb.org%2F" target="_blank" rel="nofollow ugc">官方网站</a></p>
<h3 id="介绍">介绍</h3>
<p>flyway 支持大部分主流的数据库，Oracle，SQL Server MySQL 等；<br> 并且拥有命令行工具，Maven，Gradle 插件等；<br> 配置简单，实用方便，支持 SQL 及 Java 配置，这里写一个简单的配置。</p>
<p>需要明白的几个概念：</p>
<ul>
<li>版本：对数据库的每一次变更可称为一个版本。</li>
<li>迁移：Flyway 把数据库结构从一个版本更新到另一个版本叫做迁移。</li>
<li>可用的迁移：Flyway 的文件系统识别出来的迁移版本。</li>
<li>已经应用的迁移：Flyway 已经对数据库执行过的迁移。</li>
</ul>
<h3 id="使用">使用</h3>
<ul>
<li><p>需要使用 Flyway 核心 jar 包；可直接下载 <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fojankqgpw.bkt.clouddn.com%2Fflyway-core-3.2.1.jar" target="_blank" rel="nofollow ugc">flyway-core-3.2.1.jar</a>。</p></li>
<li><p>在项目中创建文件夹 src/dbmigration，默认 src/db/migration，可以在配置中进行修改</p></li>
<li><p>对 Flyway 进行数据库配置，这里使用 Spring 配置文件配置，也可以使用 Java 对象进行配置，</p></li>
</ul>
<pre><code class="highlight-chroma">&lt;bean id="flyway" class="org.flywaydb.core.Flyway" init-method=" migrate" &gt;
    &lt;property name=" dataSource" ref=" dataSource" /&gt; &lt;!--数据源--&gt;
    &lt;property name=" locations" value=" dbmigration" /&gt; &lt;!--脚本文件夹地址 默认src/db/migration--&gt;
    &lt;property name=" validateOnMigrate" value=" false" /&gt; &lt;!--迁移验证 默认true--&gt;
    &lt;property name=" baselineOnMigrate" value=" true" /&gt; &lt;!--基线迁移 默认false--&gt;
    &lt;property name=" placeholderPrefix" value=" ignore" /&gt; &lt;!--占位符前缀--&gt;
    &lt;property name=" outOfOrder" value=" false" /&gt;
&lt;/bean&gt;
</code></pre>
<ul>
<li></li>
</ul>
```



```
<pre><code class="highlight-chroma">&lt;bean id=" sqlSessionFactory" class=" org.mybatis.spring.SqlSessionFactoryBean" depends-on=" flyway" &gt; &lt;!--建立前置依赖--&gt;
  &lt;property name=" configLocation" value=" classpath:/conf/configuration.xml" /&gt;
  &lt;property name=" dataSource" ref=" dataSource" /&gt; &lt;!--数据源--&gt;
&lt;/bean&gt;
</code></pre>
```

<p>然后，就完成了简单配置了，在 dbmigration 文件夹下创建 SQL 文件，比如：V1_0_1_初始.sql, V1_0_2_修改表.sql 等，其中 SQL 脚本文件需要按照一定的规则来命名，能够让 flyway 检并执行。</p>

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
>
```

```
<script>
```

```
(adsbygoogle = window.adsbygoogle || []).push({});
```

```
</script>
```

<p>命名规则也能进行定义比如 prefix、separator、suffix，默认 prefix 是 V（执行一遍，并且版本号唯一，如果有重复就会报错）或 R（重复执行，不需要版本号），separator 是双下划线_，suffix 是.sql。V 开头 + 版本号 + 双下划线 + 描述。</p>

<p>新版本更改脚本，版本数字大于之前的版本就可以了。</p>

<h3 id="注意">注意</h3>

<p>在启动 Tomcat 启动后 加载配置文件运行 flyway 组件 对数据库版本更新，它在维护更新时会数据库中 新建表 schame_version 用来对 SQL 脚本版本升级进行维护，一份 SQL 文件对应一条数据，所以对数据库版本升级时，会根据该表查询进行操作，如果没有记录，则会执行该 SQL 文件，如果存在数据则不执行，在对当前数据库中 SQL 脚本只会执行一次，</p>

<p>如果出现异常例如：org.flywaydb.core.api.FlywayException: Schema "xxx" contains a failed migration to version 1.1.043 !则表示数据库冲突，需要把表 schame_version 相应记录删除，并且保证 SQL 脚本操作不会与该数据库冲突。每次版本更新 schame_version 都会保存运行的结果。</p>