

# Managing Your Database Layer

---



**Kevin Jones**

@kevinrjones [www.rocksolidknowledge.com](http://www.rocksolidknowledge.com)



# Overview



**Manage the database using migrations**

**Access the database using repositories**



# Migrations




Snippets that change the schema

SQL ...

...or some other language

Ruby ActiveRecord



```
CREATE TABLE articles (  
  ID INT NOT NULL AUTO_INCREMENT,  
  URL VARCHAR(1024) NOT NULL,  
  ARCHIVED BIT(1) NULL,  
  FAVORITE BIT(1) NULL,  
  PRIMARY KEY (`id`));
```

Name	Type
ID	Int
URL	VarChar
Archived	Bit
Favorite	Bit
Title	VarChar

```
ALTER TABLE articles  
ADD TITLE VARCHAR(1024) NOT NULL;
```



# Migrations



Easy to manage database changes

Easy to recreate database

All code is now in source control

Not managed by a DBA

Multiple independent upgrades

Work across databases

Not managed by a DBA

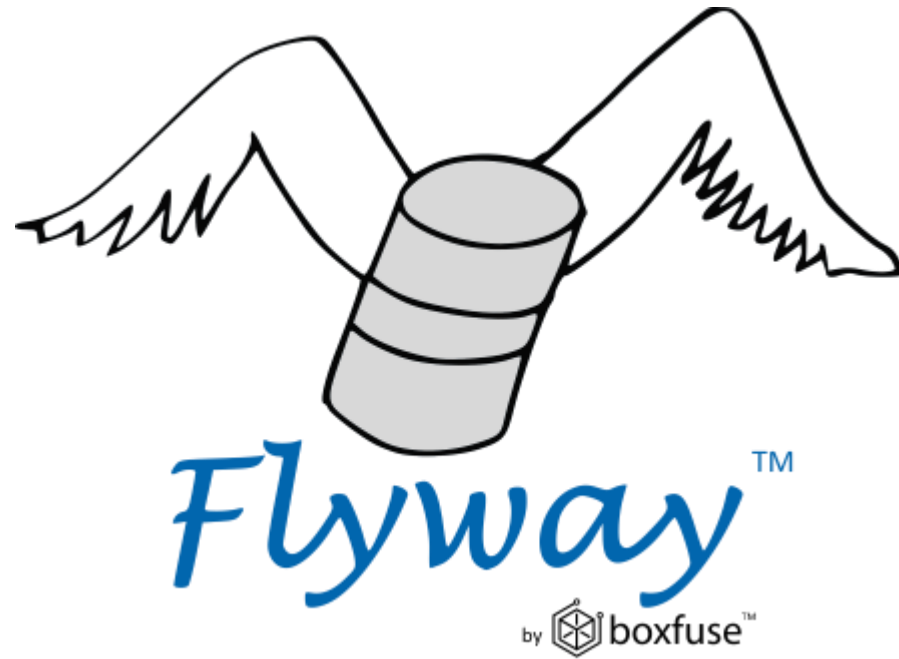
Can be difficult to version

Can end up with many small files

Don't always work across databases



# Java Migrations Using Flyway



# Flyway

<http://flywaydb.org/>

Cross database support

Maven and Gradle Support

SQL Migrations

Java Migrations



# Adding Flyway to Gradle

```
dependencies {
```

```
    classpath 'org.flywaydb:flyway-gradle-plugin:3.2.1'
```

```
}
```

```
apply plugin: 'org.flywaydb.flyway'
```



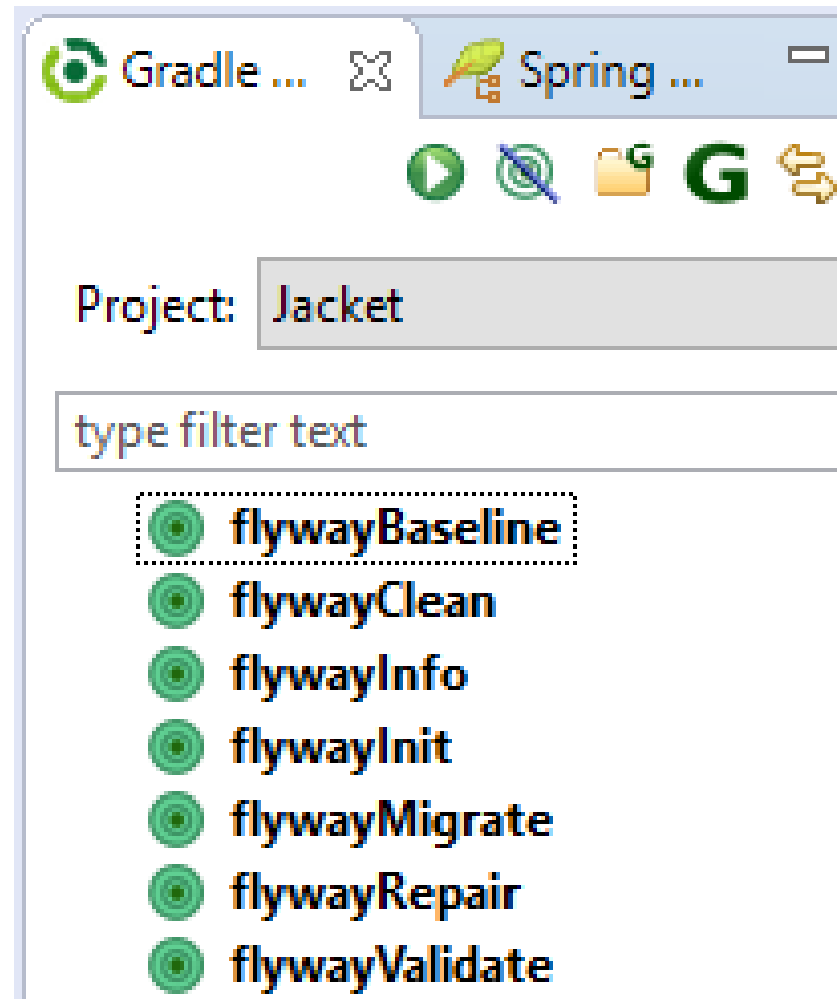


# Setting up Flyway

```
flyway {  
  
    url = 'jdbc:mysql://localhost:3306?useSSL=false'  
  
    user = 'root'  
  
    password = 'p4ssw0rd'  
  
    schemas = ['jacketdb']  
  
    locations = ["filesystem:common", "filesystem:mysql"]  
  
    outOfOrder = true  
  
}
```



# Flyway Gradle Tasks



# Spring Data Repository

An abstraction that provides access to many different data stores. Proxies are generated for each interface, that proxy contains functionality based on the names of the methods in the interface and the generic type of the repository



# Basic Repository

```
public interface ArticleRepository
```

```
    extends Repository<Article, Long> {}
```



# CRUD Repository

```
public interface ArticleRepository
```

```
    extends CrudRepository<Article, Long> {}
```



# Extending the Repository

```
public interface ArticleRepository  
    extends CrudRepository<Article, Long> {  
    List<Article> findByTitle(String title);  
}
```



# Summary



**We've Used a Repository but...**

**... It's in the Wrong Place**

**We've Coded a Physical Layer**

**What About the Logical Layers?**

