Applying Migrations to Existing Data Stores

Dan Geabunea
SENIOR SOFTWARE DEVELOPER

@romaniancoder www.romaniancoder.com

Overview

Data migrations

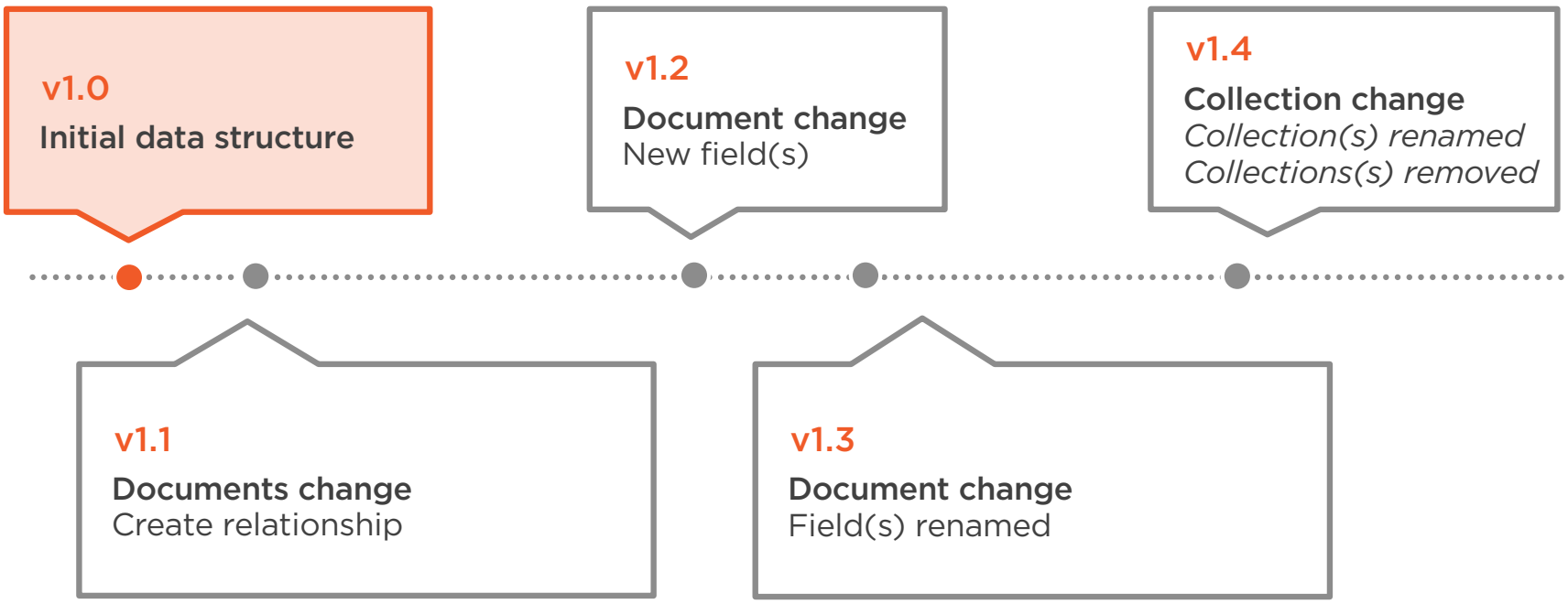
Creating Mongo data migrations in Spring applications

Demo: Implementing data migrations in Spring applications

Data Migrations

Data changes over time

Timeline


v1.0
Initial data structure
v1.2
Document change
New field(s)
v1.4
Collection change
Collection(s) renamed
Collections(s) removed
v1.1
Documents change
Create relationship
v1.3
Document change
Field(s) renamed

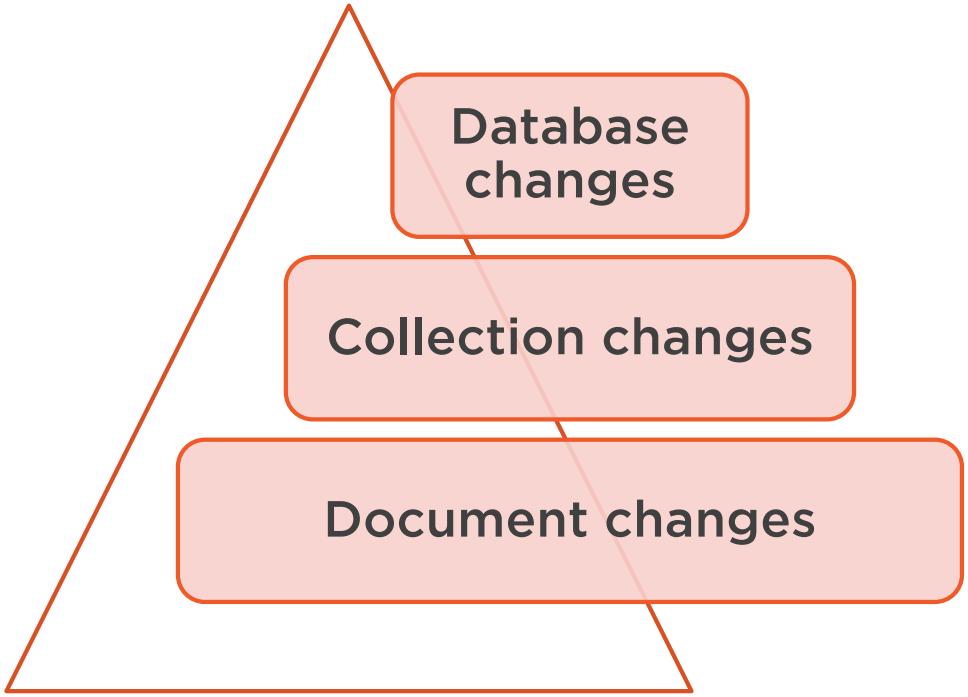
“Why are we discussing about data migrations. Didn’t we pick MongoDB because it has no schema enforcements.”

You

Reconciliation between Mongo documents and Java objects

- A change in a Mongo document will impact the serialization/deserialization process
- A change in a Mongo document will impact the queries
- Changes to collections will also impact queries

Migration Triggers



Implementation Options

Manual migrations using scripts

Create migration component yourself

Use an existing migration framework

If you're planning to sustain your project in a production environment, don't choose the manual approach

Migration Component Characteristics

Version detection

Detect existing DB version and upgrade automatically

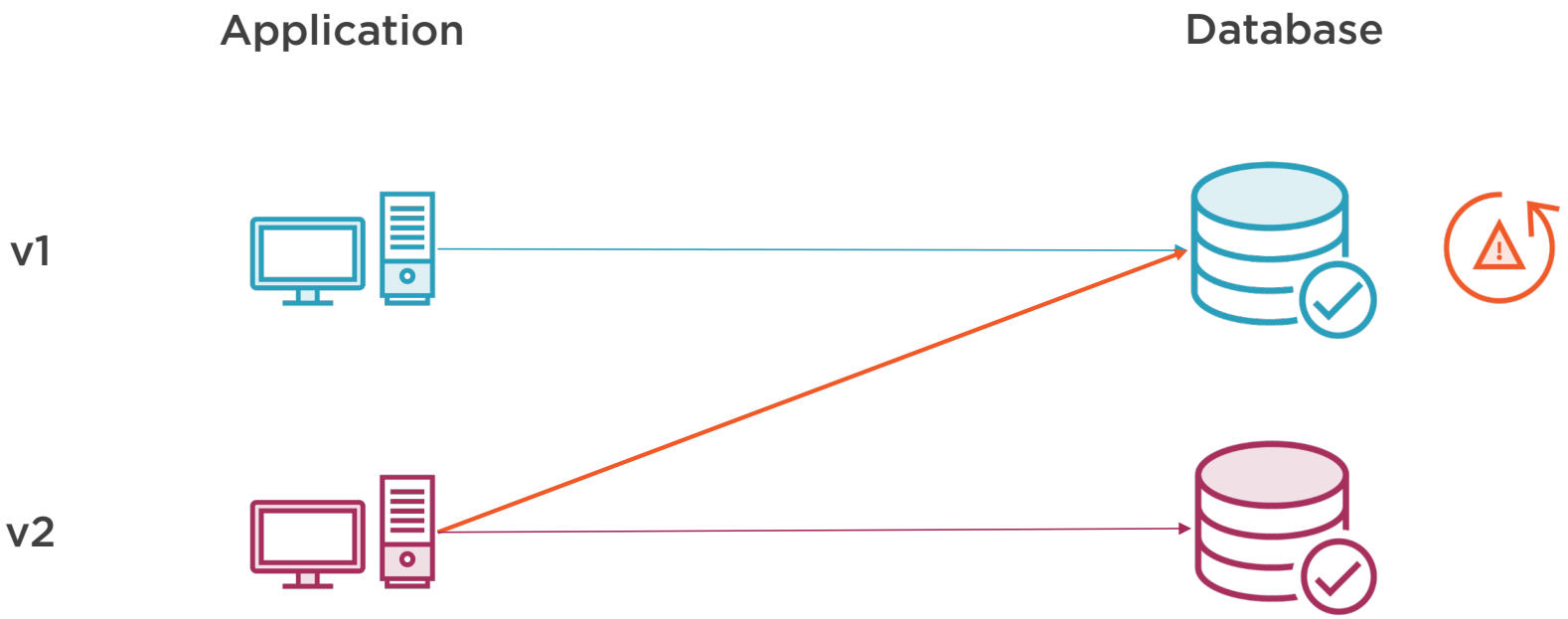
Simple configuration

Easy way to define ordered data changes

Logging

Auditing of all applied migrations

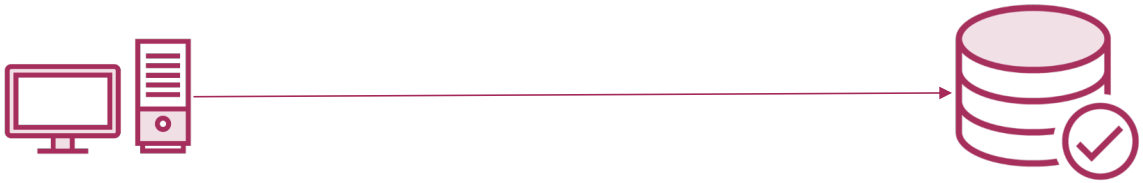
Automatic Database Version Detection



Automatic Database Version Detection

Application **Database**

v2



Creating Mongo Data Migrations in Spring Applications

Options

Create your own migration framework/components

Use an existing framework

“Poor Man’s” Migration Component

```
@Component
@Order(1)
public class DatabaseSeederRunner implements CommandLineRunner {
    private MongoTemplate mongoTemplate;
    public DatabaseSeederRunner(MongoTemplate mongoTemplate) {
        this.mongoTemplate = mongoTemplate;
    }

    @Override public void run(String... args) throws Exception {…}
}
```

So Many Questions

How is your migration component going to scale when you have plenty of changes accumulated over time?

How will you execute only the needed migrations based on the current application version and database version?

How will you keep track of the application and database versions?

Implementing a migration framework is not trivial so take your time and do it right

Using a Framework

Saves time

Increased focus on your application domain

It’s not under your control so you might loose flexibility

Mongobee

"A Java tool which helps you to *manage changes* in your MongoDB and *synchronize* them with your application. The concept is very similar to other database migration tools such as Liquibase or Flyway" – ***Official Definition***

Add Dependency

pom.xml

```
<dependency>
    <groupId>com.github.mongobee</groupId>
    <artifactId>mongobee</artifactId>
    <version>0.13</version>
</dependency>
```

Configure Bean

Configuration.java

```
@Bean public Mongobee mongobee() {
    Mongobee runner = new Mongobee(mongoUri);
    runner.setEnabled(true);
    runner.setChangeLogsScanPackage("pluralsight.*");
    runner.setMongoTemplate(this.mongoTemplate);
    return runner;
}
```

Change Logs & Change Sets

Change Log 1

- Change Set 1
- Change Set 2

Change Log 2

- Change Set 2

Create a Changelog

Changelog001.java

```
package pluralsight.db;


@ChangeLog(order = "001")
public class DbChangeLog001 {
    @ChangeSet(order = "001", id = “seedCrew", author = “John")
    public void seedCrewMembers(MongoTemplate mongoTemplate) {…}
}
```

Create Another Changelog

Changelog002.java

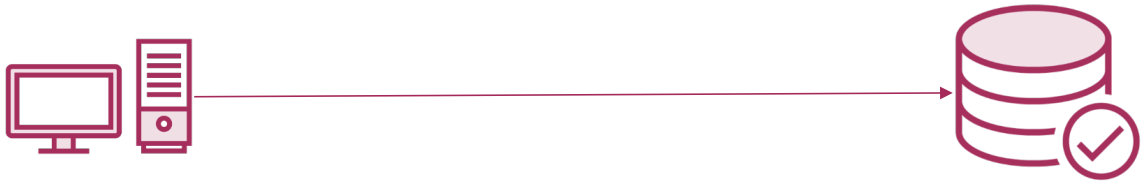
```
package pluralsight.db;


@ChangeLog(order = "002")
public class DbChangeLog002 {
    @ChangeSet(order = "001", id = “updateNationalIds", author = “Anne")
    public void updateNationalIds(MongoTemplate mongoTemplate) {…}
}
```

Automatic Database Version Detection

Application v010

Database v005



Execute ChangeLog 006

...

Execute ChangeLog 010

Demo

Demo: Implementing data migrations in Spring applications

- Configure Mongobee
- Implement migrations

Summary

Data migrations are a normal part for every software application

Try to automate the data migration process as much as possible

Create your own migration components or use a framework

Mongobee is a migration tool that integrates nicely with Spring and Mongo

“Should I use a framework or create my own migration components?”

You

The Choice Is Yours

Create migration components

Complete control of functionality

Can meet every aspect of your migration needs

Require more time to develop

Not trivial to create

Existing framework

Save time

Focus on your business logic instead of infrastructure code

Some applications might need more features than a framework can offer

You have limited control on the data migration process

“Quality is not an act, it is a habit.”

Aristotle