Spring Boot Migrator

Automated Code Migrations

Version 1.0
June 2022
Fabian Krüger

Agenda

- Introduction to Spring Boot Migrator
- Important Concepts
- Demo
- Q+A

spring

Introduction

“Spring Boot Migrator (SBM) aims to help developers upgrade or migrate to Spring Boot by providing recipes for automated migrations.”

Introduction - Automated Code Migration

Possible Automations / Recipes

- Upgrade existing Spring Boot application to recent release
- Migrating from X to Spring Boot
- Introduce a new Spring Module
- Generate context related code, e.g. Tests
- Check applications and apply migrations or provide contextual guidance (performance, cloud readiness, native readiness, ...)

➜ **These are all good use cases for SBM**
important and required but often tedious, time consuming and without additional user value

Introduction - Spring Boot Migrator

Spring Boot Migrator (SBM)

- Incubated SBM Q4 2020 with a Labs team
- Switched from JavaParser to OpenRewrite
- Jan 2021 Fabian full-time taking the lead for SBM
- March 2022 open-sourced under Spring experimental
- Opinionated API for migrations to Spring Boot
- Builds on top and is compatible with OpenRewrite (runs OR Recipes)

Introduction - SBM Recipes

JEE to Spring Boot

- JSF & Servlets
- JAX-RS / JAX-WS
- JMS
- Local Stateless EJB 3.x
- Transactions (CMP)
- EJB Deployment Descriptor
- Websphere EJB Depl. Desc.
- JPA (Hibernate & EclipseLink)

Mule to Spring Integration

- Core

  flow, logger, flow-ref, sub-flow, set-payload, choice, when, otherwise, set-property, byte-array-to-string-transformer, string-to-byte-array-transformer
- AMQP

  connector, inbound-endpoint, outbound-endpoint, publish, routing-keys, routing-key, config, connection
- WMQ

  connector, inbound-endpoint, outbound-endpoint
- HTTP

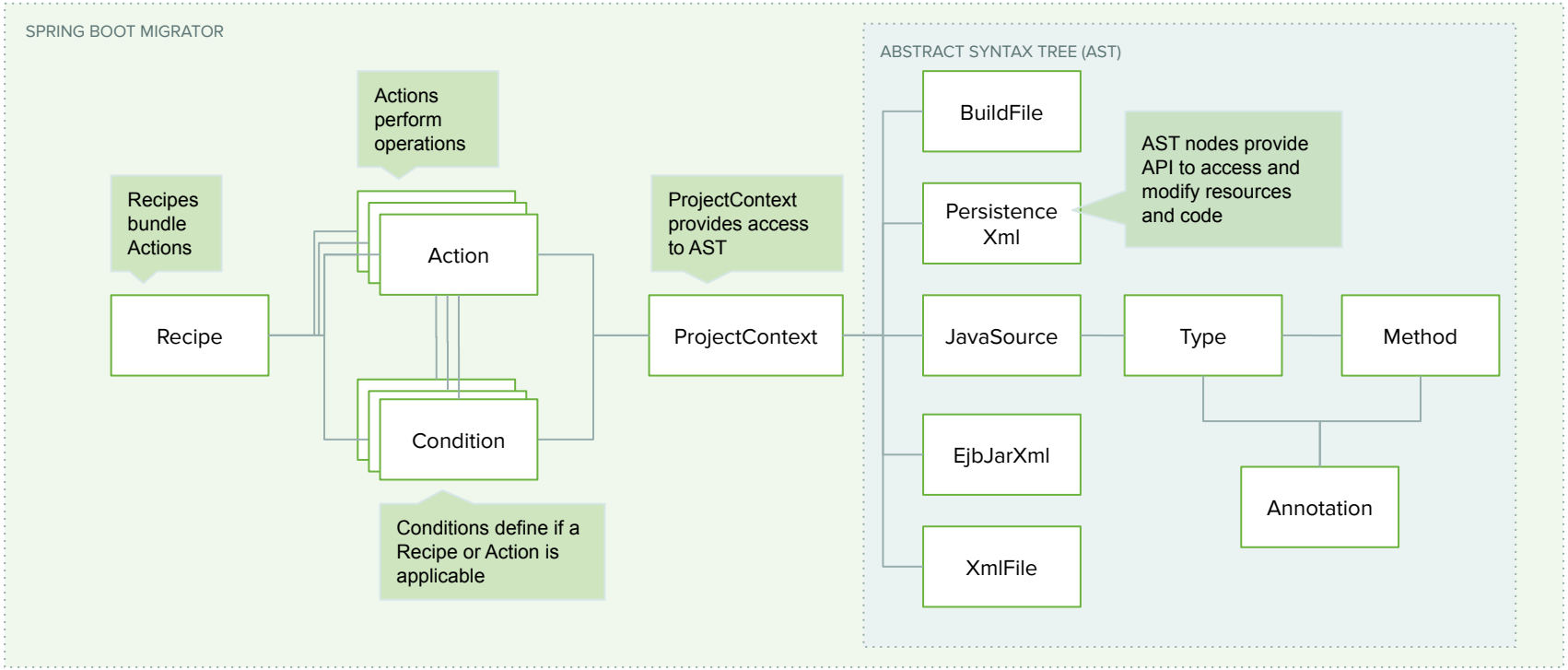
  listener-config, listener, request
- ...

Other

- Spring Boot 2.4 - 2.5 Upgrade*
- Migrate XML to Java Bean
- Introduce Spring Cloud Config
- Spring Boot 2.7 - 3.0.0-M3*

Coding SBM Recipes

SBM Concepts



spring

Recipe in YAML

- Declarative Recipe definition
- Bundles a set of Actions
- Must reside in classpath under `/recipes`

```yaml
- name: the-recipe-name
  description: Some description
  order: 70
  condition:
    description: Condition description
    type: org.springframework..SomeCondition
    value: some-condition-parameter
  actions:
   - type: org.springframework..AddDependencies
     dependencies:
       - groupId: org.springframework.boot
         artifactId: spring-boot-starter-web
         version: 2.5.6
     condition:
       type: org.springframework..NoDependencyExist
       dependency:
         groupId: org.springframework.boot
         artifactId: spring-boot-starter-web
   - type: ...second action...
   - type: ...third action...
```

spring

Recipe as Spring Bean

- Under package `org.springframework.sbm`
- Compiler verification
- Recipe and Action should have description

```
@Configuration
public class MyRecipe {

 @Bean
 public Recipe myRecipeDeclaration() {
   return Recipe.builder()
     .name("the-recipe-name")
     .description("Some description")
     .order(90)
     .condition(new ..SomeCondition("Some value"))
     .action(
       FirstAction
         .builder()
         .description("First action description")
         .condition(
           FileMatchingPatternExist.builder()
             .pattern("/**/ejb-jar.xml")
             .build()
         )
         .build()
       )
     .action(
       SecondAction
         .builder()
         ..
       )
     .build();
   }
}
```

spring

Conditions

- Define if Action/Recipe is applicable
- Conditions operate on AST
- AST accessible through ProjectContext
- Can be parametrized
- Parameters can be validated

```java
// Check if any type is annotated with annotation
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class HasAnnotation implements Condition {

  @Setter
  @NotNull
  private String annotation;

  @Override
  public String getDescription() {
    return "If there is annotation " +
           annotation + " present.";
  }

  @Override
  public boolean evaluate(ProjectContext context) {
    return context
            .getProjectJavaSources().asStream()
            .anyMatch(
              js -> js.hasAnnotation(annotation)
            );
  }
}
```

spring

Actions

- Actions operate on AST
- AST accessible through ProjectContext
- Can be parametrized
- Spring Beans can be injected
- Parameters can be validated

```
// Remove annotation from all types
@Setter
public class RemoveTypeAnnotationAction extends
  AbstractAction {

  @NotEmpty
  private String annotation;

  @Autowired
  @JsonIgnore
  private ApplicationProperties properties;


  @Override
  public void apply(ProjectContext context) {

    context.getProjectJavaSources()
        .asStream()
        .flatMap(
          js ->js.getTypes().stream()
        )
        .filter(
          t -> t.hasAnnotation(annotation)
        )
        .forEach(
          t -> t.removeAnnotation(annotation)
        );
   }
}
```

spring

ResourceFinder

- Encapsulation
- Reuse
- Access to Specialized Resources
- Different Finders exist
- See `ResourceFinder` type hierarchy

```
// ResourceFinder implementation
public class SpecializedResourceFinder
  implements ResourceFinder<Optional<SpecRes>> {

   @Override
   public Optional<SpecRes>
          apply(ProjectResourceSet rs) {

     return projectResourceSet
               .stream()
               .filter(...)
               .map(...)
               .collect(Collectors.toList());
         }
    }
}

// Using the Finder
public void apply(ProjectContext ctx) {

   Optional<SpecRes> match = ctx
       .search(new SpecializedResourceFinder());

   SpecRes sr = match.get();
   sr.specializedMethod();

}
```

spring

Apply OpenRewrite Recipe

- SBM supports subset of AST
- Use OpenRewrite Recipes and Visitors
- `apply(Recipe)` → modification, void
- Modifying method body → OpenRewrite

- `find(Recipe)` → finder, result

- Apply to single or set of resources
  - `ProjectJavaSources`
  - `JavaSourceSet`
  - `JavaSource`
  - `Type`
  - `BuildFile`

```
ProjectJavaSources pjs = projectContext
                            .getProjectJavaSources();

// Apply OpenRewrite recipe
pjs.apply(new SomeOpenRewriteJavaRecipe());


// Apply OpenRewrite finder recipe
List<RewriteSourceFileHolder<J.CompilationUnit>> m;
m = pjs.find(finderRecipe);
```

spring

Integrate OR Recipe (name)

- SBM supports subset of AST
- Use OpenRewrite Recipes and Visitors
- `apply(Recipe)` → modification, void
- Modifying method body → OpenRewrite

- `find(Recipe)` → finder, result

- Apply to single or set of resources
  - `ProjectJavaSources`
  - `JavaSourceSet`
  - `JavaSource`
  - `Type`
  - `BuildFile`

```
ProjectJavaSources pjs = projectContext
                            .getProjectJavaSources();

// Apply OpenRewrite recipe
pjs.apply(new SomeOpenRewriteJavaRecipe());


// Apply OpenRewrite finder recipe
List<RewriteSourceFileHolder<J.CompilationUnit>> m;
m = pjs.find(finderRecipe);
```

spring

Testing - TestProjectContext

- Helps Testing Actions and Conditions
- Fluent API to create ProjectContext
- No filesystem involved
- Creates resources from Strings
- Handles BuildFile and dependencies
- Support for “specialized resources”
- Support for SBM properties
- Tries to be close to “reality”

```
ProjectContext ctx = TestProjectContext
    .buildProjectContext()
    .withBuildFileHavingDependencies(
      "grouId:artifactId:1.0"
     )
     .addProjectResource(
       "src/main/java/MyClass.java", "source"
     )
     .addProjectResource(
       "src/main/resources/some.properties",
       "key=value"
     )
     .withSbmApplicationProperties(propertiesClass)
     .build();
```

spring

Demo

Demo

Upgrade Spring Boot 2.4 - 2.5

Roadmap

Roadmap

What's coming...

- Support for Maven reactor builds
- Splitting into artifacts
- Support for more Spring related resources
- YAML support
- Spring Boot 3.0 upgrade
- Spring Boot AOT guidance

Q&A

https://github.com/spring-projects-experimental/spring-boot-migrator

- Structure:
  - Problem, solution, benefit
- Concise, compelling story in 3 sentences
  - Audience: Technical people
  -
  - Topic: Spring Boot Migrator allows automated migrations

—

- Transitions between main points (leverage the language of the story structure)
  Now that you understood the problem … let me …
- Use “you, we and us” where possible
- Have “something” / Icebreaker in the beginning, maybe in the chat to get over the opening “white-noise”
-

Introduction

-