Upgrading to Spring Boot 3.0

Spring Boot 3 Workshop - Devnexus 2023 - Atlanta

Jonatan Ivanov & Phil Webb

Why Upgrade?

- GraalVM Native Images
- Micrometer Observability
- Supported Version
  - We don't expect to release Spring Boot 2.8
  - Spring Boot 2.7 support will end at some point!

About the Sample

It's a very silly application 🤪

- Just enough code to demo what we want!
- Spring Boot 2.7
- JPA (Hibernate)
- Spring MVC
- Spring Security

Checkout the Code

```
$ git clone https://github.com/jonatan-ivanov/dn23-boot3-workshop
$ cd dn23-boot3-workshop

$ git checkout main
$ docker compose up

$ sdk use java 8.0.362-librca
$ ./mvnw -pl dog-service package
```

Import into your favorite IDE

If you only have Java 17 wait until we upgrade!

Check Out the Code

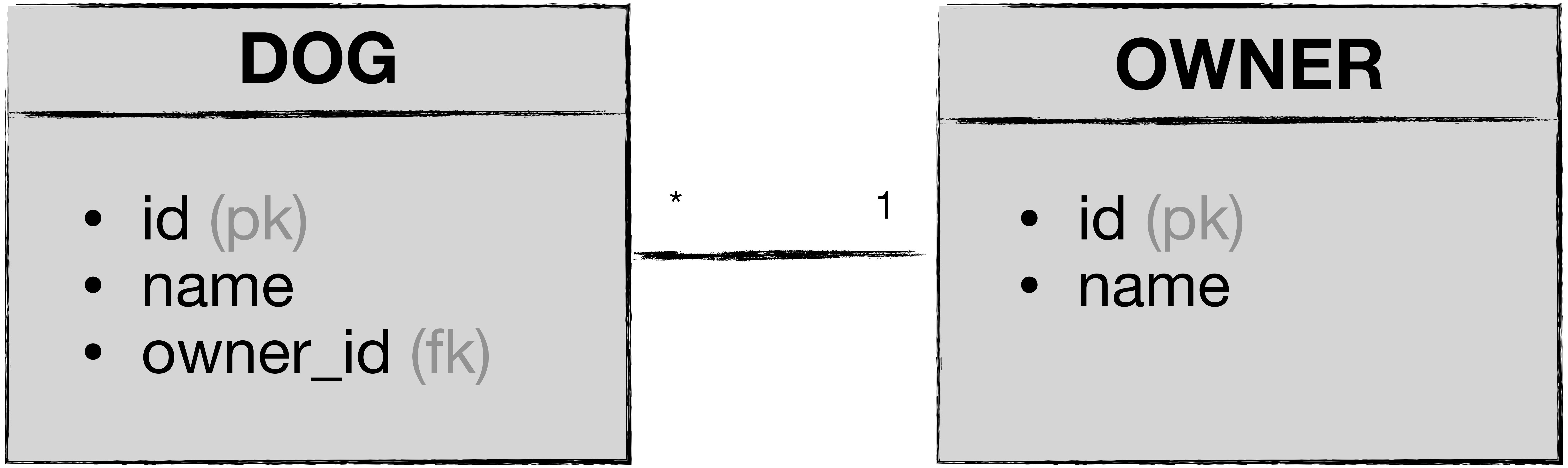
Open pom.xml

- We're using spring-boot-starter-parent 2.7
- The java.version property is 8
- We're using starters for:
  - web, security, data-jpa and actuator
- We're using postgresql

Check Out the Code

Open src/main/resources

- Look at schema.sql and data.sql files



DOG

- id (pk)
- name
- owner_id (fk)

* 1

OWNER

- id (pk)
- name

Check Out the Code

Open com.example.dogservice.domain package

- Dog and Owner JPA classes map to the schema
    - We're using javax imports but jakarta jars (Jakarta EE 8)
- DogRepository and OwnerRepository are Spring Data repositories
    - findByNameIgnoringCase is converted to JQL automatically
- InfoLogger is an ApplicationRunner to log info at startup

Check Out the Code

Open com.example.dogservice.service package

- OwnerService uses constructor injection
- Simple facade over repositories
- Throws custom NoSuchDogOwnerException

Check Out the Code

Open com.example.dogservice.web package

- DogController
  - Dirt simple controller used for testing
- OwnerController
  - Delegates to the OwnerService
  - Deals with NoSuchDogOwnerException
- Note: Meta-annoated @RestController and @GetMapping

Check Out the Code

Open com.example.dogservice.security package

- SecurityConfiguration
  - Defines our web security
  - Has SecurityProperties and UserProperties
    - @ConfigurationProperties
    - See src/main/resources/application.yml

Check Out the Code

Open src/main/resources/

- Inspect application.yml
    - Defines the database connection
    - Configures JPA
    - Enables JMX
    - Configures server errors
    - Exposes all actuator endpoints
    - Customizes a metric name
    - Defines the user details
- Look at banner.jpg

Run the code

For those with Java 8

```
$ sdk use java 8.0.362-librca
$ ./mvnw -pl dog-service clean spring-boot:run
```

Check the Java version in the logs.
If you've only got Java 17, hang tight...

Upgrade Java!

Upgrade Java

Switch to Java 17

- Spring Boot 2.7 supports Java 17, Spring Boot 3.0 requires Java 17
- A good first step is to keep Spring Boot 2.7 and upgrade Java
- Edit the pom.xml and change java.version to 17

```
$ sdk use java 17.0.6-librca
$ ./mvnw -pl dog-service clean spring-boot:run
```

Check the Java version in the logs.

Checkpoint

Everyone has a working application

Checkpoint

OwnersController Works

```
$ http "http://localhost:8080/dogs"
# Generates error JSON

$ http "http://localhost:8080/dogs?aregood=true"
$ http "http://localhost:8080/dogs?aregood=false"

$ http "http://localhost:8080/dogs/?aregood=false"
# Note the trailing slash
```

Checkpoint

DogsController Works

```
$ http "http://localhost:8080/owner/phil/dogs"
# Needs login

$ http -a user:password  "http://localhost:8080/owner/phil/dogs"
$ http -a user:password  "http://localhost:8080/owner/jonatan/dogs"

$ http -a user:password  "http://localhost:8080/owner/dave/dogs"
# NoSuchOwnerException
```

Checkpoint

Actuator Works

```
$ http -a user:password  "http://localhost:8080/actuator"
```

Open the following in a web browser:

```
http://localhost:8080/actuator
http://localhost:8080/actuator/env (search for password)
http://localhost:8080/actuator/metrics/
http://localhost:8080/actuator/metrics/http.server.in
```

Checkpoint

Actuator Over JMX Works

```
$ jconsole
```

Upgrade Spring Boot!

Upgrade Spring Boot

Update pom.xml

- Edit the pom.xml file and change the <version> tag in the parent to 3.0.5
- Things will break and we'll need to fix them!

Upgrade Spring Boot

Fix javax.persistence

- For legal reasons javax can't be used by Jakarta
- So the Dog and Owner classes now have invalid imports

- Replace javax.persistence with jakarta.persistence

Upgrade Spring Boot

Fix Spring Data repositories

- PagingAndSortingRepository no longer extends CrudRepository
- There are new convenient List... interfaces

- Change DogRepository to extend ListCrudRepository and ListPagingAndSortingRepository
- Change OwnerRepository to extend ListCrudRepository

Upgrade Spring Boot

Improve InfoLogger

- The List... interface means findAll now returns List not Iterable
- Change InfoLogger log calls to no longer use StreamSupport

Upgrade Spring Boot

Fix SecurityConfiguration

- We suppressed the WebSecurityConfigurerAdapter deprecation
- That class has gone so we need to fix things

- Define a SecurityFilterChain bean

```
@Bean
SecurityFilterChain securityFilterChain(HttpSecurity http)
   throws Exception {
   ...
   return http.build()
}
```

Upgrade Spring Boot

Fix SecurityConfiguration

- The mvcMatchers method has gone

- Replace mvcMatchers to requestMatchers

Upgrade Spring Boot

Fix SecurityConfiguration

- The http.authorizeRequests method is deprecated

- Replace http.authorizeRequests with http.authorizeHttpRequests

Upgrade Spring Boot

Fix @ConstructorBinding imports

- @ConstructorBinding has moved and we need to fix that
- Look at the Javadoc for details

- Update SecurityProperties and UserProperties with the new annotation

Upgrade Spring Boot

Remove @ConstructorBinding annotations

- Read the release notes on the wiki

- Remove @ConstructorBinding from SecurityProperties and UserProperties

Upgrade Spring Boot

Use records

- Since we're now using Java 17 we can use records

- Replace SecurityProperties and UserProperties with records
- Fix SecurityConfiguration

Upgrade Spring Boot

Use Stream.toList

- Since we're now using Java 17 we can use Stream.toList()

- Search for collect(Collectors.toList()) and replace with toList()
- SecurityConfiguration and OwnerService

Upgrade Spring Boot

Use String.formatted

- Since we're now using Java 17 we can use String.formatted

- Update Dog replacing String.format("...") with "...".formatted(...)

Upgrade Spring Boot

Use Map.of

- Since we're now using Java 17 we can use Map.of

- Update DogsController replacing Collections.singletonMap with Map.of

Be careful with Map.of and nulls!

Checkpoint

Everyone has a working Spring Boot 3.0 application

Checkpoint

The application works

```
$ ./mvnw -pl dog-service clean spring-boot:run

$ http "http://localhost:8080/dogs?aregood=true"
$ http -a user:password  "http://localhost:8080/owner/phil/dogs"
```

Upgrade Spring Boot

Fix deprecated configuration properties

- Depending on your IDE you might see warnings in your application.yml
- To help fix those, we can use the property migrator

- Update pom.xml and add spring-boot-properties-migrator artifact

Upgrade Spring Boot

Fix deprecated configuration properties

- Run the application and it will report deprecations and replacements

- Replace management.metrics.web.request.metric-name with management.observations.http.requests.name.

- Check the result in actuator

Upgrade Spring Boot

Fix deprecated configuration properties

- The properties migrator isn't free, so now we're done we can remove it

- Update pom.xml and remove spring-boot-properties-migrator artifact

Upgrade Spring Boot

Replace banner.jpg

- Sharp eyes may have noticed that the banner is missing
- Spring Boot 3.0 no longer supports image banners

- Replace banner.jpg with banner.txt and pick a dog from https://www.asciiart.eu/animals/dogs

Checkpoint

Everything works as expected

Checkpoint

The application works

```
$ ./mvnw -pl dog-service clean spring-boot:run
```

- Check the actuator env endpoint
  - Notice that all values are hidden
- Try JMX actuator
  - Notice it's missing

Upgrade Spring Boot

Expose JMX

- Actuator JMX is now aligned with web and needs to be exposed

- Add management.endpoints.jmx.exposure.include with a value of `"*"`
- Run the app again and check JMX

Upgrade Spring Boot

Always show env values

- To show env values again we need to set a property

- Add management.endpoint.env.show-values with a value of always

Upgrade Spring Boot

Sanitize env values

- Spring Boot 3.0 now leaves all sanitization to you!

- Add a SanitizingFunction to DogServiceApplication

```
@Bean
SanitizingFunction sanitizingFunction() {
    return (data) ->
        (data.getKey().toLowerCase().contains("password"))
        ? data.withValue(SanitizableData.SANITIZED_VALUE)
        : data;
}
```

Checkpoint

Everything really works as expected!

Checkpoint

Everything really works as expected!

```
$ http "http://localhost:8080/dogs"
$ http -a user:password  "http://localhost:8080/owner/dave/dogs"

# Notice the specifics of the error JSON
```

New Feature

Problem Details

- Spring Boot 3.0 supports problem details (RFC 7807)
- We need to enable it for errors handled by Spring MVC

- Set spring.mvc.problemdetails.enabled to true

New Feature

Problem Details

- We can use org.springframework.http.ProblemDetail in our own code

- Edit OwnerController.onNoSuchDogOwner

```
@ExceptionHandler
ProblemDetail onNoSuchDogOwner(NoSuchDogOwner ex) {
    ProblemDetail details = ProblemDetail.forStatus(HttpStatus.NOT_FOUND);
    details.setProperty("ownername", ex.getName());
    return details;
}
```

New Feature

Problem Details

```
$ http "http://localhost:8080/dogs"
$ http -a user:password  "http://localhost:8080/owner/dave/dogs"

# Notice the specifics of the error JSON
```

Checkpoint

Upgrade fully complete