SPRING FRAMEWORK: TIPS AND TRICKS

About presenter

- Java Engineer with a decade of experience in development and architecture.
- Open source contributor to Spring and Apache Camel Frameworks.

- Bouldering enthusiast
- Cycling fan

https://www.linkedin.com/in/kirilnugmanov/
k.nugmanov@mail.com

Links and refs

- Git Hub repo with code: https://github.com/cynicLT/spring-stuff
- To ask questions: https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

Agenda

- What is Spring Framework
- How good/bad is Spring
- Tips and tricks

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

What is Spring Framework

- A powerful open-source framework and inversion of control container for the Java platform.

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

How good/bad is Spring?

Pros

- High modularity
- Open for extensions and integration
- De facto most popular development framework

Cons

- AOP
- Dynamic proxies / Reflection
- Opinionated framework

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

Tips and tricks

- **Note:** We're savvy, so we **understand** the **repercussions** of **copying** and **pasting** in **production**.
- **Disclaimer:** The **presenter isn't responsible** for any outcomes that may or may not occur after using these hacks. All this info is just for educational purposes.

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

Tips and tricks

- There are solutions for all Your problems in stack overflow.
- Do not copy from question part :)

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

1. Re-configure existing bean

- **Problem:** bean already created but its required to proceed extra configuration.

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

1. Re-configure existing bean

```
@SpringBootApplication
public class Configuration {

    @Autowired
    public void configureJackson(ObjectMapper objectMapper) {
        objectMapper
                .setDateFormat(DateFormat.getDateInstance(DateFormat.FULL))
                .setDefaultPropertyInclusion(JsonInclude.Include.NON_EMPTY);
    }
}
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

2. Print application configurations

- **Problem:** on application start it's unknown which configuration being used.

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

2. Print application configurations

```
@Component
public final class PropertiesApplicationListener implements ApplicationListener<ApplicationReadyEvent> {

    @Override
    public void onApplicationEvent(final ApplicationReadyEvent applicationReadyEvent) {
        AbstractEnvironment environment = (AbstractEnvironment) applicationReadyEvent.getApplicationContext()
                .getEnvironment();

        environment.getPropertySources() MutablePropertySources
                .stream() Stream<PropertySource<...>>
                .filter(this::isLoggable)
                .map(MapPropertySource.class::cast) Stream<MapPropertySource>
                .map(PropertySource::getSource) Stream<Map<...>>
                .flatMap(it -> it.keySet().stream()) Stream<String>
                .distinct()
                .map(it -> it + "=" + environment.getProperty(it))
                .forEach(LOGGER::info);
    }
}
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

2. Print application configurations

```
main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.5]
main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1429 ms
main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
main] org.cynic.Application                    : Started Application in 2.477 seconds (process running for 2.866)
main] o.c.f.PropertiesApplicationListener      : spring.jackson.deserialization.fail-on-unknown-properties=false
main] o.c.f.PropertiesApplicationListener      : spring.jackson.date-format=yyyy-MM-dd
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

3. Prevent start without single active profile

- **Problem:** on activating multiple profiles its crucial order of profiles defined

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

3. Prevent start without single active profile

```java
public final class SingleActiveProfileEnvironmentPostProcessor implements EnvironmentPostProcessor {

    @Override
    public void postProcessEnvironment(final ConfigurableEnvironment environment, final SpringApplica
        List<String> activeProfiles = Arrays.asList(environment.getActiveProfiles());

        if (CollectionUtils.isEmpty(activeProfiles)) {
            throw new RuntimeException("error.no.profiles.active");
        } else if (activeProfiles.size() > 1) {
            throw new RuntimeException("error.multiple.profiles.active");
        }
    }
}
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

3. Prevent start without single active profile

1. Create **spring.factories** in **META-INF** directory
2. Define environment post processor
3. Set value as created class

spring.factories

```
1  org.springframework.boot.env.EnvironmentPostProcessor
2  =org.cynic.framework.processor.SingleActiveProfileEnvironmentPostProcessor
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

4. Custom argument resolver

- **Problem:** for controllers and advices required to have custom argument to be injected

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

4. Custom argument resolver

```
@PostMapping("/")
public Map<String, String> process(@ValidJson("/schema.json") Map<String, String> data) {
    return Map.of(
            data.get("data"),
            data.computeIfAbsent("value", s -> "default")
    );
}
}
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

4. Custom argument resolver

```java
@SpringBootConfiguration
public static class ConfigInterceptor implements WebMvcConfigurer {
    private final ObjectMapper objectMapper;

    @Override
    public void addArgumentResolvers(List<HandlerMethodArgumentResolver> resolvers) {
        resolvers.add(new ValidJsonArgumentResolver(objectMapper));
    }
}
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

4. Custom argument resolver

```java
public class ValidJsonArgumentResolver implements HandlerMethodArgumentResolver {

    @Override
    public Object resolveArgument(MethodParameter parameter, ModelAndViewContainer mavContainer, NativeWebRequest webRequest,

        HttpServletRequest nativeRequest = Optional.ofNullable(webRequest.getNativeRequest(HttpServletRequest.class))
                .orElseThrow();

        JsonSchema jsonSchema = resolveSchema(parameter);
        JsonNode json = resolveJson(nativeRequest);

        Set<ValidationMessage> errors = jsonSchema.validate(json);

        if (CollectionUtils.isEmpty(errors)) {
            return objectMapper.treeToValue(json, parameter.getParameterType());
        } else {
            throw new RuntimeException(
                    errors.stream() Stream<ValidationMessage>
                            .map(ValidationMessage::getMessage) Stream<String>
                            .collect(Collectors.joining(","))
            );
        }
    }
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

5. Slow startup

- **Problem:** Application starts too slow

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

5. Slow startup

- Don’t add starters to dependency list if not actually using it.
- Be explicit in defining starters in configuration

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

5. Slow startup

```
@ImportAutoConfiguration({
    ConfigurationPropertiesAutoConfiguration.class,

    DataSourceAutoConfiguration.class,
    HibernateJpaAutoConfiguration.class,
    TransactionAutoConfiguration.class,
    JpaRepositoriesAutoConfiguration.class,
    LiquibaseAutoConfiguration.class,

    CacheAutoConfiguration.class,

    AopAutoConfiguration.class,

    OAuth2ResourceServerAutoConfiguration.class,

    JacksonAutoConfiguration.class,

    HttpMessageConvertersAutoConfiguration.class,

    ServletWebServerFactoryAutoConfiguration.class,
    DispatcherServletAutoConfiguration.class,
    ErrorMvcAutoConfiguration.class,
    WebMvcAutoConfiguration.class
})
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

6. JPA generates too many SQLs

- **Problem:** too many SQL queries due to lazy relations

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

6. JPA generates too many SQLs

- Explicitly define fields and/or chains which want to load eagerly

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

6. JPA generates too many SQLs

```
@EntityGraph(attributePaths = {"orders"})
List<Item> findAllByOrdersIsNullOrOrdersManagerEmail(String email);
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

6. JPA generates too many SQLs

```java
@org.springframework.stereotype.Repository
public interface OrganizationRepository extends Repository<Organization, Long>, JpaSpecificationExecutor<Organization> {
    /unchecked/
    static Specification<Organization> byManagerEmail(String email) {
        return (root, query, criteriaBuilder) -> {
            SetJoin<Organization, Manager> managers = (SetJoin<Organization, Manager>) root.fetch(Organization_.managers);

            return criteriaBuilder.equal(managers.get(Manager_.email), email);
        };
    }
}
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

7. Updated whole record in DB

- **Problem:** During updates/inserts JPA updates all columns in DB

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

7. Updated whole record in DB

- Add Hibernate annotation for dynamic operations
- Instrument byte code for dynamic SQL statement generation

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

7. Updated whole record in DB

```java
@Entity
@Table(name = "DOCUMENT")
@DynamicInsert
@DynamicUpdate
public class Document {
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

7. Updated whole record in DB

```xml
<plugin>
    <groupId>org.hibernate.orm.tooling</groupId>
    <artifactId>hibernate-enhance-maven-plugin</artifactId>
    <version>${hibernate.version}</version>
    <configuration>
        <enableAssociationManagement>true</enableAssociationManagement>
        <enableDirtyTracking>true</enableDirtyTracking>
        <enableLazyInitialization>true</enableLazyInitialization>
        <enableExtendedEnhancement>false</enableExtendedEnhancement>
    </configuration>
</plugin>
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

8. Bi-directional associations management

- **Problem:** on adding/removing JPA entity from other entity its required to update both sides

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

8. Bi-directional associations management

```xml
<plugin>
    <groupId>org.hibernate.orm.tooling</groupId>
    <artifactId>hibernate-enhance-maven-plugin</artifactId>
    <version>${hibernate.version}</version>
    <configuration>
        <enableAssociationManagement>true</enableAssociationManagement>
        <enableDirtyTracking>true</enableDirtyTracking>
        <enableLazyInitialization>true</enableLazyInitialization>
        <enableExtendedEnhancement>false</enableExtendedEnhancement>
    </configuration>
</plugin>
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

8. Bi-directional associations management

```
public void $$_hibernate_write_orders(Set var1) {
    Object[] var2;
    if ((true || Hibernate.isInitialized(this.orders)) && this.$$_hibernate_read_orders() != null) {
        var2 = this.orders.toArray();

        for(int var3 = 0; var3 < var2.length; ++var3) {
            if ((true || Hibernate.isPropertyInitialized(var2[var3], attributeName: "documents")) && (va
                ((Order)var2[var3]).$$_hibernate_read_documents().remove(this);
            }
        }
    }
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

9. Lazy loading of primitive types

- **Problem:** on query JPA selects all columns defined in Entity (in some cases even lazy one relations)

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

9. Lazy loading of primitive types

- **Note:**
  - ID column can not be lazy
  - Columns fetched during read (access)

```
<plugin>
    <groupId>org.hibernate.orm.tooling</groupId>
    <artifactId>hibernate-enhance-maven-plugin</artifactId>
    <version>${hibernate.version}</version>
    <configuration>
        <enableAssociationManagement>true</enableAssociationManagement>
        <enableDirtyTracking>true</enableDirtyTracking>
        <enableLazyInitialization>true</enableLazyInitialization>
        <enableExtendedEnhancement>false</enableExtendedEnhancement>
    </configuration>
</plugin>
```

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

Conclusion

- Use technologies wisely:
  - Read documentation
  - Surf stack overflow

- Repository to check code:
  https://github.com/cynicLT/spring-stuff

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL

Questions/answers/discussion

https://app.sli.do/event/cXNfQ3P8bMaXNcDBjZaAPL