J. B. Hunt
Project NEXT
Spring Boot 2.1.6.RELEASE & Java 11 – OpEx – Migration v0.2

	Prepared By / Last Updated By	Reviewed by	Approved By
Name	Jayandh Vijayakumar	Aravindan	
Role	Technical Lead	Architect	
Signature			
Date	10/09/2019		

1.0   Table of Contents

Document Properties

Document Name	Spring Boot 2.1.6.RELEASE & Java 11 – OpEx – Migration
Version Number	0.2
Date of Last Revision	29-October-2019

1.0 Overview

Objective(Goal)	The objective of the document is to give an overview on **Spring Boot 2.1.6.RELEASE** migration for entire OpEx Project. It also speaks of the **Spring Framework 5** & **Java 11** migration and define in detail about the benefits, configuration details of the system involved and their features.

2.0 Targeted Upgrade

2.1 In Scope

2.1.1 Technology Upgrade

Technology	Version		Level
	From	To	
Spring Boot	1.5.9 RELEASE	2.1.8.RELEASE	Mandatory
Spring Framework	4.3.6.RELEASE	5.1.8.RELEASE	Mandatory
Spring Batch	spring-batch-admin-manager	spring-cloud-data-flow	Mandatory
Java	8	11	Mandatory
Maven	3.2.1	3.3+	Mandatory
Hibernate	5.0	5.3+	Mandatory
ehcache	2.x	3.x	Mandatory
Tomcat	8	9	Mandatory
Map struct	1.0.0.Final	1.3.0.Final	Mandatory

2.1.2 JBH Parent Pom Upgrade

Group Id	Artifact Id	Version
com.jbhunt	jbhunt_starter_webservice	2.1.8.1
com.jbhunt	jbhunt_starter_listener	

2.2 Out of scope

- Only Spring Boot and Java libraries planned for this upgrade. Apart that, all are beyond the scope of this upgrade.

2.3 Assumptions

- We can add only the jbhunt-subsystem's libraries as a maven dependency for any particular project pom.
- Any parent pom changes would not affect the consuming maven projects.
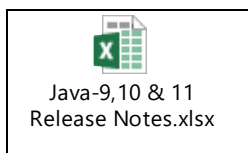
2.4 Open Items

- "spring-batch-admin-manager" service has already reached EOL, we do not supposed to use. Do we need to migrate to the Spring Cloud Data Flow?
- JBHunt Parent POM may have all the maven dependency that I do not want to use on my project, so I guess using single parent pom for all ws, listener, DTO and Entity project will not be suitable. Do we have any strategy to follow how can we solve this?
  - Yes, we have separate pom for web service and listener.
  - https://jbhunt.visualstudio.com/EngAndTech/_git/lib_infrastructure_jbhunt_parent?path=%2Fjbhunt_starters&version=GBmaster
- Jenkins builder is using java 8 to compile and deploy the library [DTO / Entity / Client] project's code. So do we have any plan to upgrade Jenkins?
- Which jbhunt parent we need to use for library projects such as DTO, Entity and Client?
  - No need to use JBHunt parent pom for DTO, Entity and Client libraries.
  -
- First is with reference to Compatibility for dependent systems. As when we upgrade our client libraries, we that also have to consider about other services those are dependent on ours. For e.g., tracks like OM, JBH Drive and OBC are utilizing our client libraries. Even though we are upgrading it, those systems may not have any plans to upgrade them in near future. This will result in a conflict. Due to this, we may have to maintain a working branch of Java 8 while we continue to add new features to the other branch where we are upgrading to Java 11. Also going forward, it would be best if each products/domains maintain their own clients for web services, which they have to call outside of their domain as suggested by Peter. If we are going to do so, those web services should be capable enough to handle both PID and OAuth.
-

2.5 Risks / Issues

- "spring-batch-admin-manager" service has already reached EOL.
- Functional break might be happen, if the consumers missed upgrade their program to match OPEX's techstack after the time span [two Sprints].
- Since Spring Boot2.x, configuration property types are concern about canonical format. Application start up failure will be occur, if any one of a subsystem's client does not follow.

3.0 Java 11 Upgrade

3.1 Java 8 to 11 Release notes xls

Java-9,10 & 11
Release Notes.xlsx

[Java-9, 10 & 11 Release Notes](#)

3.2 In scope for upgrade

Feature	Level
Complete the removal, begun in Java SE 8, of the underscore from the set of legal identifier names.	Mandatory

Remove the JavaEE Modules.
   • Java Transaction API (JTA) in the package javax.transaction
   • **JAXB in the packages javax.xml.bind.***
   • **JAX-WS in the packages javax.jws, javax.jws.soap, javax.xml.soap, andjavax.xml.ws.***
   • Commons Annotation in the package javax.annotation
Fixes.
   • JAF: with com.sun.activation:javax.activation
   • CORBA: there is currently no artifact for this
   • JTA: javax.transaction:javax.transaction-api
   • **JAXB: com.sun.xml.bind:jaxb-impl**
   • **JAX-WS: com.sun.xml.ws:jaxws-ri**
   • Commons Annotation: javax.annotation:javax.annotation-api
	Mandatory

Map Struct upgrade from mapstruct-jdk8 1.0.0.Final to mapstruct1.3.0.Final	Mandatory

3.3 Out scope for java 11 upgrade

- Java 11 is the scope of this section, other than that beyond the scope.
- Using Java Platform Module System, we can create our own custom JRE with only Java modules required for application need. This greatly reduces memory footprint. We can create it using jlink command.
- HTTP/2 – Reactive Programming [Which can be permanent solution for the STOMP delay issue.]

3.4 Detailed design for Inscope components

- TBD.

3.5 Reference projects

GIT URL	https://jbhunt.visualstudio.com/EngAndTech/_git/ws_loadplanning_integration_backfill
Project	ws_loadplanning_integration_backfill
Branch	branch_springboot2.1.6_migration

| GIT URL	https://jbhunt.visualstudio.com/EngAndTech/_git/listener_obc_message_sync
Project	listener_obc_message_sync
Branch	branch_springboot2.1.6_migration

3.6 Assumptions

- TBD.

3.7 Open Items

- Is JBHunt recommending jdk is Zulu jdk 11, which is available on software center?
    - Based on ws_infrastructure_ri **jbh-alpine-jre11 dockerimages-prd.jbhunt.com/jbh-alpine-jre11:latest**

3.8 Risks / Issues

- TBD.

4.0 Spring & Spring Boot Upgrade

4.1 Spring Boot 1.5.9 to 2.1.6 release notes xls

Spring Boot 2.0 &
2.1 Upgrade Release
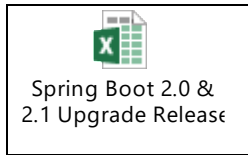
[Spring Boot 2.0 & 2.1 Upgrade Release](#)

4.2 In scope for upgrade

Spring Boot 1.5.9.RELEASE	Spring Boot 2.1.6.RELEASE	Level
org.springframework.boot.web.support.SpringBootServletInitializer	org.springframework.boot.web.servlet.support.SpringBootServletInitializer	Mandatory
org.springframework.boot.autoconfigure.jdbc.DataSourceBuilder	org.springframework.boot.jdbc.DataSourceBuilder	Mandatory
org.springframework.data.querydsl.QueryDslPredicateExecutor	org.springframework.data.querydsl.QuerydslPredicateExecutor	Mandatory
org.springframework.cloud.netflix.feign.EnableFeignClients	org.springframework.cloud.openfeign.EnableFeignClients	Mandatory
org.springframework.boot.autoconfigure.web.HttpMessageConverters	org.springframework.boot.autoconfigure.http.HttpMessageConverters	Mandatory
org.springframework.ws.soap.security.wss4j.Wss4jSecurityInterceptor	org.springframework.ws.soap.security.wss4j2.Wss4jSecurityInterceptor	Mandatory
org.springframework.web.context.request.RequestAttributes	org.springframework.web.context.request.WebRequest	Mandatory
org.springframework.boot.autoconfigure.web.DefaultErrorAttributes	org.springframework.boot.web.servlet.error.DefaultErrorAttributes	Mandatory
org.springframework.web.context.request.ServletRequestAttributes	org.springframework.web.context.request.ServletWebRequest	Mandatory
org.springframework.cloud.netflix.FeignClient	org.springframework.cloud.openfeign.FeignClient	Mandatory
	org.springframework.web.servlet.config.annotation.WebMvcConfigurer	Mandatory
ehcache 2.x is available under **net.sf.ehcache** package	**ehcatche 3.x** is available under **org.ehcache** package	Mandatory
Hibernate 5.0.11	Hibernate 5.3 [JPA 2.2]	Mandatory
spring-batch-admin-manager	spring-cloud-data-flow	Mandatory
spring-cloud-dependencies **Edgware.SR2**	spring-cloud-dependencies **Greenwich.RC1**	Mandatory

	Property type should be in canonical format	Mandatory
Application Monitoring with Prometheus	<mark>Application Monitoring with Micrometer - Spring Boot 2.0 no longer ships with its own metrics APIs. Instead, rely on micrometer.io for all application monitoring needs.</mark>	Mandatory

4.3 Out scope for upgrade

- TBD.

4.4 Detailed design for Inscope components

- TBD.

4.5 Reference projects

- TBD.

4.6 Assumptions

- TBD.

4.7 Open Items

- "spring-batch-admin-manager" service has already reached EOL, we do not supposed to use. Do we need to migrate to the Spring Cloud Data Flow?

4.8 Risks / Issues

- TBD.

5.0 OAuth Upgrade

5.1 In scope for upgrade

Feature	Level
Java will expose all services with OAuth compatibility.	Mandatory
Angular will call the java services with security token by passing on header detail	Mandatory

5.2 Out scope for upgrade

Feature	Level
Retain PID security for Active MQ communication.	Mandatory

5.3 Detailed design for Inscope components

5.3.1 Java Security Configuration Changes

```java
import static com.jbhunt.autoconfigure.security.esa.JBHAuthorizationAccessLevel.POWER;
import static com.jbhunt.autoconfigure.security.esa.JBHAuthorizationAccessLevel.SUPER;
import static com.jbhunt.autoconfigure.security.esa.JBHAuthorizationAccessLevel.UPDATE;

import org.springframework.context.annotation.Configuration;
import org.springframework.http.HttpMethod;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import com.jbhunt.autoconfigure.security.JbhSecurity;
import com.jbhunt.autoconfigure.security.oauth2.configuration.WebServiceSecurityConfigurerAdapter;

@Configuration
@JbhSecurity(esaMode = JbhSecurity.EsaMode.IMPERSONATION)
public class SecurityConfiguration extends WebServiceSecurityConfigurerAdapter {

    @Override
    protected void configureAppSecurity(HttpSecurity http) throws Exception {


        http.authorizeRequests()
                .antMatchers("/health", "/info", "/actuator", "/trace", "/docs/**", "/beans", "/metrics")
                .permitAll()
                .antMatchers(HttpMethod.POST,"/ruleexecutions")
                .hasAnyAuthority(UPDATE.getAuthority(), POWER.getAuthority(), SUPER.getAuthority())
                .anyRequest()
                .denyAll();
    }

}
```

5.3.2 YML entry

```yaml
jbhunt:
  oauth2:
      public-key-url: https://oauth2-dev.jbhunt.com/oauth2provider/oauth/publickey
      resource-server-name: ws_operationsexecution_ruleresult
      resource-id-resolver-url: https://oauth2-dev.jbhunt.com/oauth2admin/clientId/appName
```

5.4 Assumptions

- None.

5.5 Open Items

- Do we need to keep PID configuration after implemented the Oauth?
- Is there any solution to using OAuth in Active MQ settings?

5.6 Risks / Issues

6.0 OpEx POC Project details

6.1 Angular Project

6.1.1 GIT URLS

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/app_operationsexecution_loads_ui
Project	app_operationsexecution_loads_ui
Branch	OAuth_poc

6.2 Spring Zuul Proxy

6.2.1 GIT URLS

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/app_operationsexecution_loads_api
Project	app_operationsexecution_loads_api
Branch	branch_springboot2.1.6_migration

6.3 WS project

6.3.1 GIT URLS

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/ws_operationsexecution_resourcenotes
Project	ws_operationsexecution_resourcenotes
Branch	branch_springboot2.1.6_migration

6.4 Listener project

6.4.1 GIT URLS

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/listener_operationsexecution_resourcenotes
Project	listener_operationsexecution_resourcenotes
Branch	branch_springboot2.1.6_migration

6.5 Batch project

6.5.1 GIT URLS

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/batch_obc_equipmentpositions
Project	batch_obc_equipmentpositions
Branch	branch_springboot2.1.6_migration

6.6 Lib DTO / Client / Entity project

6.6.1   GIT URLS

6.6.1.1   DTO

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/lib_operationsexecution_resourcenotes_dto
Project	lib_operationsexecution_resourcenotes_dto
Branch	branch_springboot2.1.6_migration

6.6.1.2   Entity

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/lib_operationsexecution_resourcenotes_entity
Project	lib_operationsexecution_resourcenotes_entity
Branch	branch_springboot2.1.6_migration

6.6.1.3   Client

URL	https://jbhunt.visualstudio.com/EngAndTech/_git/ lib_operationsexecution_resourcenotes_elastic_client
Project	lib_operationsexecution_resourcenotes_elastic_client
Branch	branch_springboot2.1.6_migration

6.7 POC Reference Documents

Spring Boot2, Java 11 and Oauth2 Migration _V.3

Backfill_migration_springboot2.1.6_v0.1

7.0 Version Control

Version #	Modified By	Description of Changes
0.1	Jayandh Vijayakumar	Initial version
0.2	Jayandh Vijayakumar	Added additional information with POC reference documents.