



**J. B. Hunt**

**Project NEXT**

**Spring Boot 2.1.6.RELEASE – OpEx –BackFill Migration v0.1**

**Spring Boot 2, Java 11 and Oauth 2 Migration**



	Prepared By / Last Updated By	Reviewed by	Approved By
<b>Name</b>	Balaji		
<b>Role</b>	Developer		
<b>Signature</b>			
<b>Date</b>	22/10/2019		



**Reference Implementation:**

Implemented the Spring Boot 2.1.6, Java 11 and OAuth2.0 in the below web service project and successfully deployed the project in a separate environment.

<b>Project Name</b>	ws_loadplanning_integration_backfill
<b>Branch Detail</b>	branch_springboot2.1.6_migration
<b>Name Space</b>	loadplanning-test-env1
<b>URL</b>	https://loadplanning-test-env1.nonprod.jbhunt.com/ws_loadplanning_integration_backfill/*

**Dependent Projects:**

Below are the dependent projects to the above web service. Implemented the changes and deployed in the same environment.

<b>Project Name</b>	<b>Branch Details</b>
lib_personnel_people_client	branch_springboot2.1.6_migration
lib_masterdata_asset_client	branch_springboot2.1.6_migration
lib_loadplanning_integration_backfill_entity	branch_springboot2.1.6_migration

**POM Dependency:**

Below dependencies are required for spring Boot 2.1.6 migration:

<b>Description</b>	<b>Mandatory / Optional</b>	<b>Dependency Name</b>	<b>Reason</b>
JBHunt starter parent includes all the basic spring boot dependencies	Mandatory	<pre>&lt;parent&gt;   &lt;groupId&gt;com.jbhunt&lt;/groupId&gt;   &lt;artifactId&gt;jbhunt_starter_parent&lt;/artifactId&gt;   &lt;version&gt;2.1.2.1&lt;/version&gt; &lt;/parent&gt;</pre>	<p>Jbhunt starter parent dependencies.</p> <p>Refer the list of available dependencies as below path  <a href="http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_starter_parent/2.1.2.1">http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_starter_parent/2.1.2.1</a>  <a href="http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_dependencies/2.1.2.1">http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_dependencies/2.1.2.1</a></p>
JBHunt starter webservice includes io.common, sleuth, autoconfigure, spring security oauth, starter actuator, starter config, starter web, stater cache and etc..	Mandatory	<pre>&lt;dependency&gt;   &lt;groupId&gt;com.jbhunt&lt;/groupId&gt;   &lt;artifactId&gt;jbhunt_starter_webservice&lt;/artifactId&gt; &lt;/dependency&gt;</pre>	<p>List of dependencies required to start a web service.</p> <p>Refer the list of available dependencies as below path  <a href="http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_dependencies/2.1.2.1">http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_dependencies/2.1.2.1</a></p>

			<a href="https://github.com/jbhunt/jbhunt_starter_web_service/2.1.2.1/">repos/com/jbhunt/jbhunt_starter_web_service/2.1.2.1/</a>
JBHunt starter test includes starter test, restassured, auto configure test, cloud contract wire mock, wiremock standalone, lib_infrastructure_jbhunt_test	Mandatory	<pre>&lt;dependency&gt;  &lt;groupId&gt;com.jbhunt&lt;/groupId&gt;  &lt;artifactId&gt;jbhunt_starter_test&lt;/artifactId&gt;  &lt;scope&gt;test&lt;/scope&gt;  &lt;/dependency&gt;</pre>	<p>List of dependencies required test case development.</p> <p>Refer the list of available dependencies as below path  <a href="http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_starter_test/2.1.2.1/">http://artifactory-prd.jbhunt.com/artifactory/maven-repos/com/jbhunt/jbhunt_starter_test/2.1.2.1/</a></p>
Spring boot configuration processor	Mandatory	<pre>&lt;dependency&gt;  &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;  &lt;artifactId&gt;spring-boot-configuration-processor&lt;/artifactId&gt;  &lt;optional&gt;true&lt;/optional&gt;  &lt;/dependency&gt;</pre>	Required for application context configuration.
Querydsl	Optional	<pre>&lt;dependency&gt;  &lt;groupId&gt;com.querydsl&lt;/groupId&gt;  &lt;artifactId&gt;querydsl-apt&lt;/artifactId&gt;  &lt;version&gt;4.2.1&lt;/version&gt;  &lt;/dependency&gt;</pre>	Required for querydsl support
javax.jws	Optional	<pre>&lt;dependency&gt;  &lt;groupId&gt;com.sun.xml.ws&lt;/groupId&gt;  &lt;artifactId&gt;jaxws-ri&lt;/artifactId&gt;  &lt;version&gt;2.3.0&lt;/version&gt;  &lt;type&gt;pom&lt;/type&gt;  &lt;/dependency&gt;</pre>	<p>Jax-ws package removed from java11.</p> <p>This dependency is added for jax-ws package support.</p>
javax.ws.rs	Optional	<pre>&lt;dependency&gt;  &lt;groupId&gt;javax.ws.rs&lt;/groupId&gt;  &lt;artifactId&gt;javax.ws.rs-api&lt;/artifactId&gt;  &lt;version&gt;2.1.1&lt;/version&gt;  &lt;/dependency&gt;</pre>	This dependency added for javax.ws package support.

com.sun.activation	Optional	<pre>&lt;dependency&gt;  &lt;groupId&gt;com.sun.activation&lt;/g roupld&gt;  &lt;artifactId&gt;javax.activation&lt;/arti factId&gt;     &lt;version&gt;1.2.0&lt;/version&gt; &lt;/dependency&gt;</pre>	This dependency added for javax.activation package support.
feign	Optional	<pre>&lt;dependency&gt;  &lt;groupId&gt;org.springframework.c loud&lt;/groupId&gt;     &lt;artifactId&gt;spring-cloud- starter-openfeign&lt;/artifactId&gt; &lt;/dependency&gt;</pre>	Required for feign client support
org.apache.cxf	Optional	<pre>&lt;dependency&gt;  &lt;groupId&gt;org.apache.cxf&lt;/group Id&gt;     &lt;artifactId&gt;cxf-rt-frontend- jaxws&lt;/artifactId&gt;     &lt;version&gt;3.3.3&lt;/version&gt; &lt;/dependency&gt; &lt;dependency&gt;  &lt;groupId&gt;org.apache.cxf&lt;/group Id&gt;     &lt;artifactId&gt;cxf-rt-transport- http&lt;/artifactId&gt;     &lt;version&gt;3.3.3&lt;/version&gt; &lt;/dependency&gt; &lt;dependency&gt;  &lt;groupId&gt;org.apache.cxf&lt;/group Id&gt;     &lt;artifactId&gt;cxf-rt-databinding- aegis&lt;/artifactId&gt;     &lt;version&gt;3.3.3&lt;/version&gt; &lt;/dependency&gt;</pre>	Apache cxf spring boot2 support dependencies.
org.json	Optional	<pre>&lt;dependency&gt;     &lt;groupId&gt;org.json&lt;/groupId&gt;     &lt;artifactId&gt;json&lt;/artifactId&gt;</pre>	Added for JSONException support

		<version>20190722</version> </dependency>	
org.apache.cxf- cxf-codegen-plugin	Optional	<plugin>  <groupId>org.apache.cxf</groupId> <artifactId>cxf-codegen-plugin</artifactId> <version>3.3.3</version>  </plugin>	Cxf-codegen plugin, spring boot2 support.
com.github.jaxb-xew-plugin	Optional	<dependency> <groupId>com.github.jaxb-xew-plugin</groupId> <artifactId>jaxb-xew-plugin</artifactId> <version>1.11</version> </dependency>	Jaxb spring boot2 support.
org.apache.cxf.xjcplugins	Optional	<dependency> <groupId> org.apache.cxf.xjcplugins </groupId> <artifactId>cxf-xjcdv</artifactId> <version>3.3.0</version> </dependency>	Cxf plugin spring boot2 support.
Java version	Mandatory	11	Added for java-11 compiler version.

**Exclude the below dependencies from all client :**

**Issue:** runtime error caused by lib client dependency.

Caused by: java.io.FileNotFoundException: class path resource

[org/springframework/security/config/annotation/authentication/configurers/GlobalAuthenticationConfigurerAdapter.class] cannot be opened because it does not exist

**Fix:** We need to exclude below dependency from lib client to avoid the runtime error.

```
<exclusions>
  <exclusion>
    <groupId>com.jbhunt</groupId>
    <artifactId>lib_security_autoconfigure</artifactId>
  </exclusion>
</exclusions>
```

**Code Level Changes:**

Type of changes	Imports to be used	Imports to be removed
DataSourceBuilder	import org.springframework.boot.jdbc.DataSourceBuilder	org.springframework.boot.autoconfigure.jdbc.DataSourceBuilder
SpringBootServletInitializer	import org.springframework.boot.web.servlet.support.SpringBootServletInitializer	org.springframework.boot.web.support.SpringBootServletInitializer
QuerydslPredicateExecutor	import org.springframework.data.querydsl.QuerydslPredicateExecutor	org.springframework.data.querydsl.QueryDslPredicateExecutor
FeignClient	import org.springframework.cloud.openfeign.FeignClient	org.springframework.cloud.netflix.FeignClient

**1. In Client @ConfigurationProperties property name should be canonical(kebab case) form.**

To meet Spring Boot 2.x requirements, @ConfigurationProperties prefix needs to update to canonical form (kebab-case)

Before:

```
@ConfigurationProperties("masterdataAsset")
public class MasterdataAssetProperties {
    private String baseUrl;
```

After:

```
@ConfigurationProperties("masterdata-asset")
public class MasterdataAssetProperties {
    private String baseUrl;
```

Reference: <https://github.com/spring-projects/spring-boot/issues/9545>

## 2. Security configuration:

In order to migrate basic authentication to oAuth token authentication, we need to update security authentication class with @JbhSecurity annotation.

Configure security with below configuration to direct access the web service without any ESA configuration.

### Before:

```
@Configuration
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

    private AuthorizationConfiguration authorizationConfiguration;

    public SecurityConfiguration(AuthorizationConfiguration authorizationConfiguration) {
        this.authorizationConfiguration = authorizationConfiguration;
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable();
        http.httpBasic();
        http.authorizeRequests().anyRequest().authenticated();
        //To be configured later with required URL authorization.
        http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
        http.addFilterAfter(authorizationConfiguration.getAuthorizationFilter(), BasicAuthenticationFilter.class);
        http.addFilterAfter(authorizationConfiguration.getSwitchUserFilter(), AuthorizationFilter.class);
    }
}
```

### After:

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

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

        http.requestMatchers().antMatchers(HttpMethod.GET).and().authorizeRequests()
            .antMatchers("/health", "/info", "/prometheus", "/metrics").permitAll().anyRequest().authenticated()
            .and().httpBasic().and().csrf().disable();
        http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
    }
}
```

**3. Docker File changes:**

Update the Docker image path from jdk 8 to jbh-alpine-jre11:latest

**Before:**

**FROM** azul/zulu-openjdk-alpine:8

**EXPOSE** 8080

**RUN** wget http://configuration-tst.jbhunt.com/ws\_infrastructure\_configuration/static/krb5.conf -O /etc/krb5.conf

**ADD** target/ws\_loadplanning\_integration\_backfill.jar ws\_loadplanning\_integration\_backfill.jar

**ENTRYPOINT** exec java \$**JAVA\_OPTS** -Djava.security.krb5.conf=/etc/krb5.conf -jar ws\_loadplanning\_integration\_backfill.jar

**After:**

**FROM** dockerimages-prd.jbhunt.com/jbh-alpine-jre11:latest

**EXPOSE** 8080

**RUN** wget http://configuration-tst.jbhunt.com/ws\_infrastructure\_configuration/static/krb5.conf -O /etc/krb5.conf

**ADD** target/ws\_loadplanning\_integration\_backfill.jar ws\_loadplanning\_integration\_backfill.jar

**ENTRYPOINT** exec java \$**JAVA\_OPTS** -Djava.security.krb5.conf=/etc/krb5.conf -jar ws\_loadplanning\_integration\_backfill.jar

**4. Test case changes:**

Type of changes	Imports to be used	Imports to be removed
<b>RequestSpecification</b>	Import RequestSpecification from io.restassured.specification.RequestSpecification	com.jayway.restassured.specification.RequestSpecification
<b>DocumentationConfiguration</b>	Import org.springframework.restdocs.restassured3.RestAssuredRestDocumentation.documentationConfiguration;	org.springframework.restdocs.restassured.RestAssuredRestDocumentation.documentationConfiguration;
<b>modifyUri</b>	Import org.springframework.restdocs.operation.preprocess.Preprocessors.modifyUri	org.springframework.restdocs.restassured.operation.preprocess.RestAssuredPreprocessors.modifyUri
<b>Document</b>	Import org.springframework.restdocs.restassured3.RestAssuredRestDocumentation.document	org.springframework.restdocs.restassured.RestAssuredRestDocumentation.document
<b>Given</b>	Import io.restassured.RestAssured.given	com.jayway.restassured.RestAssured.given
<b>Response</b>	import io.restassured.response.Response	import com.jayway.restassured.response.Response
<b>ContentType</b>	io.restassured.http.ContentType	com.jayway.restassured.http.ContentType



**5. Wire mock port setup for all clients:**

Wire mock port should be given inside the before class to mock the client end points.

Eg:

```
@Before
public void setup() {
    RestAssured.port = port;
    this.requestSpecification = new RequestSpecBuilder()

    .addFilter(RestAssuredRestDocumentation.documentationConfiguration((this.restDocumentation))).build();
    masterdataAssetProperties.setBaseURL(
"http://localhost:"+Integer.toString(wireMockServer.port())+"/ws_masterdata_asset");
    crossreferenceProperties.setBaseURL(
"http://localhost:"+Integer.toString(wireMockServer.port())+"/masterdatacrossreferenceservices");
    personnelPeopleProperties.setBaseURL(
"http://localhost:"+Integer.toString(wireMockServer.port())+"/personnelpeople");
    locationProperties.setBaseURL(
"http://localhost:"+Integer.toString(wireMockServer.port())+"/masterdatalocationservices");
}
```

**6. Oauth configuration for Test case:**

<b>JBH Test Annotation</b>	@JbhSecurityTest
<b>Auto Wire Mock configuration</b>	@Autowired private MockConfiguration mockConfiguration;
<b>ESA Mock configuration</b>	mockConfiguration.esa().user("jisatest").accessLevels(JBHAuthorizationAccessLevel.UPD ATE).and().and() .jwt().user("jisatest").and() .configure();  String bearerToken = (String) mockConfiguration.get("bearer");
<b>Authorization header</b>	given(this.requestSpecification).header("Authorization", "Bearer " + bearerToken).accept(ContentType.JSON)

**7. YML changes:****1. Context Path changes:**

Update context path from server.contextPath to server.servlet.context-path

Before:

```
server:  
  contextPath: /ws_loadplanning_integration_backfill
```

After:

```
server:  
  servlet:  
    context-path: /ws_loadplanning_integration_backfill
```

**2. Oauth :**

Update resource server name and oauth required endpoints

```
Jbhunt:  
  oauth2:  
    public-key-url: https://oauth2-tst.nonprod.jbhunt.com/oauth2provider/oauth/publickey  
    resource-server-name: ws_loadplanning_integration_backfill  
    resource-id-resolver-url: https://oauth2-tst.nonprod.jbhunt.com/oauth2admin/clientId/appName
```

**3. JPA Configuration changes:**

Update data source property url to jdbcUrl.

Before:

```
general:  
  datasource:  
    com.jbhunt.datasource.JBHDB2P:  
      url: jdbc:db2://DEVPJES2.jbhunt.com:5031/JBHDB2T
```

After:

```
general:  
  datasource:  
    com.jbhunt.datasource.jbhdb2p:  
      driver-class-name: com.ibm.db2.jcc.DB2Driver  
      jdbcUrl: jdbc:db2://DEVPJES2.jbhunt.com:5031/JBHDB2T
```

## Oauth 2 Implementation in oauth Admin dashboard:

1. **Jbhunt OAuth url:**  
**Dev:** oauth2-dev.jbhunt.com/oauth2admin/dashboard  
**Test:** oauth2-tst.jbhunt.com/oauth2admin/dashboard
2. **Project level OAuth 2 configuration in oauth admin dashboard.**

The screenshot shows the OAuth Admin dashboard for the application 'ws\_loadplanning\_integration\_backfill'. The 'Resource Type' is set to 'Internal'. Under 'Grant Types', 'Client Credentials', 'Password', 'Authorization Code', 'Implicit', and 'Refresh Token' are all selected. The 'Resources' section lists several resources with checkboxes: 'app\_security\_nexus\_ui', 'batch\_capacity\_emptyplantemplate', 'batch\_finance\_no\_loads\_ingestion', and others. At the bottom, there are 'SAVE' and 'RESET' buttons, and the 'Authorization URL' and 'Token URL' are displayed.

Once the backfill service is register in the oauth dashboard and it will appear as below.

	B82B58F0-8021-4D57-BEEE-922529BB4C72	ws_loadplanning_integration_backfill	J.B. Hunt Transport Inc. NAC: JBHUNT Bill To: JBHUNT	-	Implicit Refresh Token Client Credentials Password Authorization Code
		ws_loadplanning_integration_backfill	J.B. Hunt Transport Inc. NAC: JBHUNT Bill To: JBHUNT	-	Implicit Refresh Token Client Credentials Password Authorization Code

## Oauth implementation for getting New Access Token:

Provide the Access Token URL, Respective project client ID and Client Secret to get new access token for OAuth Authentication.

### GET NEW ACCESS TOKEN

Token Name

Backfill Token

Grant Type

Password Credentials

Access Token URL ⓘ

https://oauth2-tst.nonprod.jbhunt.com/oauth2provider/oauth/token

Username

jisabs0

Password

\*\*\*\*\*

☐ Show Password

Client ID ⓘ

B82B58F0-8021-4D57-BEEE-922529BB4C72

Client Secret ⓘ

ZTVhODUyNzMtMDc4Yi00ZGVhLWFKZTItMzA0NjFkYjkyZjE4

Scope ⓘ

e.g. read:org

Client Authentication

Send as Basic Auth header

Request Token

env -createFleet

Comments (0) Examples (0)

POST

https://loadplanning-test-env1.nonprod.jbhunt.com/ws\_loadplanning\_integration\_backfill/backfill/operationalevents

Send Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Co

TYPE

OAuth 2.0

The authorization data will be automatically generated when you send the request. [Learn more about authorization](#)

Add authorization data to

Request Headers

Preview Request

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

Access Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqaXNhYnMwliwidXNlcl9uYW1lIjoiamlZlZWJzMCIsImZcyI6Imh0dHBzOi8vb2F1dGgyLXRzdC5ub25wcm9kLmpiaHVudC5jb20vb2F1dGgyZHVudmlkZXliLCJpc0ludGVybmFsVXNlciI6dHJ1ZSwiY2xpZW50X2lk

Available Tokens

Get New Access Token

Body

Cookies

Headers (7)

Test Results

Status: 200 OK Time: 1121ms Size: 291 B Save Response

Pretty

Raw

Preview

Visualize BETA

JSON

1

2

3

4

5

"statusCode": "200",

"statusMessage": "success",

"retryEnabled": false

### Request Payload:

POST /ws\_loadplanning\_integration\_backfill/backfill/operationalevents HTTP/1.1

Content-Type: application/json

Authorization: Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqaXNhYnMwliwidXNlcl9uYW1lIjoiamlZlZWJzMCIsImZcyI6Imh0dHBzOi8vb2F1dGgyLXRzdC5ub25wcm9kLmpiaHVudC5jb20vb2F1dGgyZHVudmlkZXliLCJpc0ludGVybmFsVXNlciI6dHJ1ZSwiY2xpZW50X2lk



```
"equipmentUnAssignmentDTOs": null,  
"carrierAssignmentDTOs": null,  
"carrierUnAssignmentDTOs": null  
}  
}
```

**Response Payload:**

```
HTTP/1.1 200 OK  
Server: nginx/1.13.12  
Date: Tue, 22 Oct 2019 12:04:40 GMT  
Content-Type: application/json;charset=UTF-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Vary: Accept-Encoding  
Content-Encoding: gzip  
  
{  
  "statusCode": "200",  
  "statusMessage": "success",  
  "retryEnabled": false  
}
```

**Dependent Projects Changes:****lib\_personnel\_people\_client:****1. Parent pom changes:**

Update parent pom from spring-boot-starter-parent (1.5.9.RELEASE) to jbhunt\_starter\_parent (2.1.2.1)

**2. Cananical form changes:****Before:**

```
@Data  
@ConfigurationProperties("personnelPeople")  
public class PersonnelPeopleProperties {  
    private String baseUrl;  
}
```

**After:**

```
@Data  
@ConfigurationProperties("personnelpeople")  
public class PersonnelPeopleProperties {  
    private String baseUrl;  
}
```

**lib\_masterdata\_asset\_client:****1. Parent pom changes:**

Update parent pom from spring-boot-starter-parent (1.5.9.RELEASE) to jbhunt\_starter\_parent (2.1.2.1)

**2. Cananical form changes:****Before:**

```
@Data  
@ConfigurationProperties("masterdataAsset")  
public class MasterdataAssetProperties {  
    private String baseUrl;  
    private String assetDeactivationBaseUrl;  
}
```

```
private String geoCodeBaseUrl;  
private String geoCode;  
private Integer clientConnectionMaxTotal;  
private String masterdataBaseUrl;  
private int defaultMaxTotalPerRoute;  
private int maxTotalPerRoute;  
private String hostName;  
}
```

**After:**

```
@Data  
@ConfigurationProperties("masterdataasset")  
public class MasterdataAssetProperties {  
  
    private String baseUrl;  
    private String assetDeactivationBaseUrl;  
    private String geoCodeBaseUrl;  
    private String geoCode;  
    private Integer clientConnectionMaxTotal;  
    private String masterdataBaseUrl;  
    private int defaultMaxTotalPerRoute;  
    private int maxTotalPerRoute;  
    private String hostName;  
}
```

**lib\_loadplanning\_integration\_backfill\_entity:****1. Parent pom changes:**

Update parent pom from spring-boot-starter-parent (1.5.9.RELEASE) to jbhunt\_starter\_parent (2.1.2.1)