



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
Name	Balaji		
Role	Developer		
Signature			
Date	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.

Project Name	ws_loadplanning_integration_backfill
Branch Detail	branch_springboot2.1.6_migration
Name Space	loadplanning-test-env1
URL	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.

Project Name	Branch Details
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:

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

Spring Boot 2.1.6.RELEASE – OpEx – Migration

Cognizant

Project Name NEXT

Create Date: Oct 21, 2019

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

com.sun.activation	Optional	<pre><dependency> <groupId>com.sun.activation</groupId> <artifactId>javax.activation</artifactId> <version>1.2.0</version> </dependency></pre>	This dependency added for javax.activation package support.
feign	Optional	<pre><dependency> <groupId>org.springframework.cloud</groupId> <artifactId>spring-cloud-starter-openfeign</artifactId> </dependency></pre>	Required for feign client support
org.apache.cxf	Optional	<pre><dependency> <groupId>org.apache.cxf</groupId> <artifactId>cxf-rt-frontend-jaxws</artifactId> <version>3.3.3</version> </dependency> <dependency> <groupId>org.apache.cxf</groupId> <artifactId>cxf-rt-transports-http</artifactId> <version>3.3.3</version> </dependency> <dependency> <groupId>org.apache.cxf</groupId> <artifactId>cxf-rt-databinding-aegis</artifactId> <version>3.3.3</version> </dependency></pre>	Apache cxf spring boot2 support dependencies.
org.json	Optional	<pre><dependency> <groupId>org.json</groupId> <artifactId>json</artifactId></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-xjc-dv</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 needs 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 needs 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.IMPORATION)
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
RequestSpecification	Import RequestSpecification from io.restassured.specification.RequestSpecification	com.jayway.restassured.specification.RequestSpecification
DocumentationConfiguration	Import org.springframework.restdocs.restassured3.RestAssuredRestDocumentation.documentationConfiguration;	org.springframework.restdocs.restassured.RestAssuredRestDocumentation.documentationConfiguration;
modifyUris	Import org.springframework.restdocs.operation.preprocess.Preprocessors.modifyUris	org.springframework.restdocs.restassured.operation.preprocess.RestAssuredPreprocessors.modifyUris
Document	Import org.springframework.restdocs.restassured3.RestAssuredRestDocumentation.document	org.springframework.restdocs.restassured.RestAssuredRestDocumentation.document
Given	Import io.restassured.RestAssured.given	com.jayway.restassured.RestAssured.given
Response	import io.restassured.response.Response	import com.jayway.restassured.response.Response
ContentType	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:

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

OAuth2 Admin

HOME

Edit Client: B82B58F0-8021-4D57-BEEE-922529BB4C72

Application Name: ws_loadplanning_integration_backfill

Resource Type: Internal

Grant Types: Client Credentials, Implicit, Refresh Token

Redirect URLs: app_security_nexus_ui, patch_capacity_emptyplantemplate, patch_finance_no_loads_ingestion

SAVE RESET

Authorization URL: https://oauth2-tst.nonprod.jbhunt.com/oauth2provider/oauth/authorize
Token URL: https://oauth2-tst.nonprod.jbhunt.com/oauth2provider/oauth/token

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

To get the secret code click the clipboard icon.

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

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: ZTVhODUyNzMtMDc4Yi00ZGViLWFkZTItMzA0NjFkYjkyZjE4

Scope: e.g. read:org

Client Authentication: Send as Basic Auth header

Request Token

env-createFleet

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 Cor

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

Access Token eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWliOiJqaXNhYnMwliwidXNlcI9uYW1ljoiamlzYWJzMClslmzcyI6Imh0dHBzOi8vb2F1dGgyLXRzdC5ub25wcm9kLmpiaHVudC5jb20vb2F1dGgycHJvdmlkZXliLCJpc0ludGVybmbFsVXNlcil6dHJ1ZSwiY2xpZW50X2lk

Get New Access Token

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

Pretty Raw Preview Visualize BETA JSON

Request Payload:

POST /ws_loadplanning_integration_backfill/backfill/operationalevents HTTP/1.1

Content-Type: application/json

Authorization: Bearer

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWliOiJqaXNhYnMwliwidXNlcI9uYW1ljoiamlzYWJzMClslmzcyI6Imh0dHBzOi8vb2F1dGgyLXRzdC5ub25wcm9kLmpiaHVudC5jb20vb2F1dGgycHJvdmlkZXliLCJpc0ludGVybmbFsVXNlcil6dHJ1ZSwiY2xpZW50X2lk

IjoiQjyQjU4RjAtODAyMS00RDU3LUJFRUUtOTIyNTI5QkI0QzcyliwiYXBwX25hbWUiOij3c19sb2FkcGxhbm5pbmdfaW50ZWdyYXRpb25fYmFja2ZpbGwiLCJ1c2VyX2lkIjoiamlzYWJzMCIsInNjb3BlIjpblmRIZmF1bHQiXswiY29tcGFueSI6eyJuYW1lljoiSi5CLiBldW50IFRyYW5zcG9ydCBjbMuliwiaWQiOm51bGwslnN0YXRlIjpudWxsLCJjb3VudHJ5IjpudWxsLCJuYXRpb25hbEFjY291bnRDb2RlljoiSkJIVU5UliwiZG9jdW1bnRSZXbvc2l0b3J5VVJMljpudWxsLCJjb2RIQXNb2NpYXRpb25EVE9MaXN0IjpudWxsLCJwcmltYXJ5UmVmZXJlbmNVHlwZSI6bnVsbCwiY29tcGFueVNlY3VyaXR5RW51bSI6bnVsbCwibGlua2VkJQ2FycmlcklkljpudWxsLCJkb2N1bWVudFJlcG9zaXRvcnI0YW1lljpudWxsLCJiaWxsMkNvZGUoijkQkhVTIQiLCJidXNTaXRJljpudWxsLCJwaG9uZSI6bnVsbCwicGVyc29uYUxpc3QiOm51bGwsImNpdHkiOm51bGwsImFkZHJlc3MxljpudWxsLCJ6aXAiOm51bGwsImFkZHJlc3MyljpudWxsLCJyb2xlTGldCl6bnVsbH0sImV4cCI6MTU3MTgzMTI2OCwiaWF0ljoxNTcxNzQ0ODY4LCJqdGkiOii5YjU3OGM4OS1iMjQxLTRINjgtYjYxZi1iMDU1NWYyMWRIzGUiLCJ1c2VybmtZSI6IkjhGFqaS5TYXRoeWFuYXJheWFuYW5AamJodW50LmNvbSlslnJlc291cmNVHlwZSI6InByaXZhdGUiLCJ1c2VybmtZSI6IkjhGFqaS5TYXRoeWFuYXJheWFuYW4ifQ.kEn0cOEQ0640F1TKBVYVhoEBftlfmuknlJL6DtZWu2_yFxV4n2ECGvSjQYbzaUF3ydOUr2GtCu8X9mfp91Wljp6dCSbVq__2Vzm7CJRopPMHgrejkYHJBxDfk-1U48J7iJU1s0kACIW1JzgRduVT1jXjPcDGob3z7OogPCF7uqOvBb0YKwCPS3sWCKjhGO-IAHz7EyvVe-Yt8My95-iZYB0tlm8QoelxK52_-5v9YpiBbjZ9QDx1sz66Lxjkyknl-d55JFywqiYF_nOSbb2BFIrZFN_CT3peRy2rdbwgRlwu_roO0fsqPRHI_2S3yjRbOm_rbpUPh3PEb5CTNg

User-Agent: PostmanRuntime/7.18.0

Accept: */*

Cache-Control: no-cache

Postman-Token: 4400895c-9200-4a44-9783-c3df7150d2a8

Host: loadplanning-test-env1.nonprod.jbhunt.com

Accept-Encoding: gzip, deflate

Content-Length: 1382

Connection: keep-alive

```
{
  "operationalGroupEventType": "CREATED",
  "operationalGroupEventSubTypes": [
    "OPERATIONAL_GROUP",
    "MEMBER"
  ],
  "operationalGroupDetailsDTO": {
    "operationalGroupCode": "ADM SB1CSL",
    "operationalGroupDescription": "SB1CA SHOP FLEET CODE",
    "operationalGroupTypeCode": "Fleet",
    "operationalGroupTypeDescription": "Fleet",
    "operationalGroupSubTypeCode": "OTR",
    "operationalGroupSubTypeDescription": null,
    "financeBusinessUnitCode": "HJBT JBDCS",
    "lastUpdateProgramName": "System",
    "lastUpdateTimestamp": "2019-08-12T09:03:26.075Z",
    "lastUpdateUserId": "System",
    "effectiveTimestamp": "2019-08-12T09:03:26.059Z",
    "expirationTimestamp": "2199-12-31T23:59:59Z",
    "totalMembers": 5,
    "status": "Active",
    "utilizationStatusDTOs": null,
    "driverAssignmentDTOs": [],
    "driverUnAssignmentDTOs": null,
    "equipmentAssignmentDTOs": [
      {
        "operationalGroupEquipmentAssignmentId": 30528,
        "equipmentId": 237720,
        "equipmentUnitId": "344750",
        "equipmentClassificationCode": "STRTRKBOX",
        "effectiveTimestamp": "2019-08-01T09:27:12.717-05:00",
        "expirationTimestamp": "2019-08-01T23:59:59.009999999-05:00"
      }
    ]
}
```

```
"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. Canonical 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. Canonical 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)