

Java Developers Guide

- [Gradle](#)
- [Jacoco - Code Coverage](#)
- [Java Asynchronous Computation - CompletableFuture](#)
- [Java EE Developers Guide](#)
- [Java Lambda/Streams Developers Guide](#)
- [MapStruct](#)
- [Maven](#)
- [Mockito](#)
- [Reactive Streams](#)

TOC: see [Developer Guide](#)

- [JVisualVM](#)
- [Java Versions](#)
 - [Java 8 LTS](#)
 - [Java 11 LTS](#)
 - [Java 11 removal of JAXB - add 2.3.0 directly to your pom.xml](#)
- [Testing](#)
 - [JUnit Testing](#)
 - [CI JUnit Testing](#)
 - [Mockito Testing](#)
 - [Integration Testing](#)
 - [CD JUnit Testing](#)
- [Maven](#)
 - [Maven Central HTTPS support as of 15 Jan 2020](#)
 - [Add HTTPS support in Maven Builds](#)

https://en.wikipedia.org/wiki/Java_version_history

JVisualVM

JVisualVM is distributed with the JDK until JDK8. In Java 9 and above run it separately via <https://visualvm.github.io/download.html>

Java Versions

Java 8 LTS

Java 11 LTS

In general even switching your JDK to 11 will add a 40% increase in single thread performance even running JDK8 compiled bytecode.

see [Performance#Java11support](#)

Get 11.0.5 directly from <https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>

It makes sense to move from Java 8 to 11 (not 9/10 or 12/13) because 11 is the LTS until 17 becomes LTS. Support for Java11 also is in Spring 5.1

Required for JDK 11 support (spring 5.1+)

spring-boot 2.1.x = Spring core 5.1.5

spring-boot 2.2.0-RC = Spring core 5.2.0

Therefore we need to up our spring-boot version from 2.0 to 2.1 to pickup spring 5.1 which support Java 11 <https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.1-Release-Notes>

To deploy SB 5.1 apps we will also need to upgrade from Tomcat 8.5 to 9.0, Hibernate to 5.3 and JUnit to 5.2

Java changes from 8 to 11 - primarily I see [reactive streams](#) as the main feature in J9 and local variable changes, concurrency improvements

Java 11 adds

+ local var lambda params and (http client, JRockit Flight Recorder, TLS 1.3)

Java 10 (discontinued) adds

+ local var type inference, threadlocal handshakes

Java 9 (discontinued) adds

+ reactive streams and (modules, jshell, concurrency updates)

Prototype with but do not put into production Java 12-13 for now until we are close to Java 17 LTS (in 18+ months)

Java 11 removal of JAXB - add 2.3.0 directly to your pom.xml

<https://github.com/obrienlabs/biometric/issues/51>

<https://jira.onap.org/browse/LOG-126>

[OBRIENLABS-18](#) - Getting issue details... STATUS

<https://github.com/obrienlabs/biometric/pull/52>

Testing

JUnit Testing

CI JUnit Testing

Reference

Mockito Testing

There several ways to shallow or deep mock controller, service and repository components.

<https://github.com/obrienlabs/magellan/blob/master/magellan-nbi/src/test/java/global/packet/magellan/ForwardingControllerTest.java>

```
public class ForwardingControllerTest {
    // controller is unmocked, but service bean inside is mocked
    private @InjectMocks ForwardingController controller = Mockito.mock(ForwardingController.class);

    @Before
    public void setup() {
        MockitoAnnotations.initMocks(this);
    }

    @Test
    public void testForwardRequest() {
        String expected = "ok";
        Mockito.when(controller.getHealth())
            .thenReturn(expected);
        String response = controller.getHealth();
        Assertions.assertNotNull(response);
    }
}
```

Integration Testing

CD JUnit Testing

Add the following spring annotations - added automatically by the spring boot wizard

This test will actually start the container and perform anything on startup - like connecting to a persistent store for example - it should be run only if the jenkins CI server can also run CD.

```
@RunWith(SpringRunner.class)
@SpringBootTest
public class NbiApplicationTests {
    @Test
    // fix jpa db context later
    public void contextLoads() {
    }
}
```

Citrus

Maven

Maven Central HTTPS support as of 15 Jan 2020

I am running fine on my macbook pro but not on my jenkins server

Issue: During a maven build pulling new artifacts - for example switching from Spring boot 2.1.3 to 2.2.4 you can no longer pull from maven central without SSL support

```
Parsing POMs
Failed to transfer Could not transfer artifact org.springframework.boot:spring-boot-starter-parent:pom:2.2.4.RELEASE from/to central (http://repo.maven.apache.org/maven2): Failed to transfer file: http://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.2.4.RELEASE/spring-boot-starter-parent-2.2.4.RELEASE.pom. Return code is: 501 , ReasonPhrase:HTTPS Required.
ERROR: Failed to parse POMs
org.apache.maven.project.ProjectBuildingException: Some problems were encountered while processing the POMs:
[FATAL] Non-resolvable parent POM: Could not transfer artifact org.springframework.boot:spring-boot-starter-parent:pom:2.2.4.RELEASE from/to central (http://repo.maven.apache.org/maven2): Failed to transfer file: http://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.2.4.RELEASE/spring-boot-starter-parent-2.2.4.RELEASE.pom. Return code is: 501 , ReasonPhrase:HTTPS Required. and 'parent.relativePath' points at no local POM @ line 5, column 10

    at org.apache.maven.project.DefaultProjectBuilder.build(DefaultProjectBuilder.java:364)
    at hudson.maven.MavenEmbedder.buildProjects(MavenEmbedder.java:361)
    at hudson.maven.MavenEmbedder.readProjects(MavenEmbedder.java:331)
    at hudson.maven.MavenModuleSetBuild$PomParser.invoke(MavenModuleSetBuild.java:1328)
    at hudson.maven.MavenModuleSetBuild$PomParser.invoke(MavenModuleSetBuild.java:1125)
    at hudson.FilePath.act(FilePath.java:1075)
```

see <https://github.com/microsoft/azure-pipelines-tasks/issues/12156>

Add HTTPS support in Maven Builds

Use Maven 3.6.3 - and in the case of Jenkins upgrade from Maven 3.2 to 3.4 at least so that we can pull from maven central