### Mavenプロジェクト構成

本節では、[pom.xml](https://github.com/logpresso/logpresso-app-examples/blob/main/logpresso-sample-app/pom.xml)の構成を項目ごとに解説します。全体の内容についてはソースをご参照ください。

#### アーティファクト情報

<groupId>com.logpresso.sonar</groupId><artifactId>logpresso-sample-app</artifactId><version>1.0.2306.0</version><packaging>bundle</packaging><name>Logpresso Sample App</name>

ログプレッソ・ソナーアプリのグループIDは常に com.logpresso.sonar と指定してください。アーティファクトIDは logpresso-VENDOR-MODEL 形式で定義することを推奨します。バージョンは Major.Minor.YYMM.REV 形式で定義します。アプリのバージョンにリリース時期を明記することで、対応するログプレッソ・ソナープラットフォームの推定が容易になります。

最後に、パッケージングが bundle に指定されている点に特にご注意ください。後述する maven-bundle-plugin プラグインが bundle パッケージングを実行します。

#### 環境変数設定

<properties> <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding> <jdk.version>11</jdk.version></properties>

OSロケールがビルドに影響しないよう、ファイルエンコーディングをUTF-8に明示し、JDKバージョンを11に指定します。jdk.version 変数は後述の設定で利用されます。

#### Mavenリポジトリ

<repositories> <repository> <id>logpresso-repo</id> <name>Logpresso Maven Repository</name> <url>https://maven.logpresso.com/</url> </repository></repositories><pluginRepositories> <pluginRepository> <id>logpresso-plugin-repo</id> <name>Logpresso Maven Repository</name> <url>https://maven.logpresso.com/</url> </pluginRepository></pluginRepositories>

ログプレッソ・ソナーは https://maven.logpresso.com リポジトリからサードパーティ開発に必要なライブラリを提供しています。ビルド時にバイトコード変換が必要なiPOJOプラグインをダウンロードできるよう <pluginRepository> 設定を追加し、アプリAPIおよびiPOJOライブラリを取得できるよう <repository> 設定を追加します。

#### ビルド設定

Javaコンパイル時のソースコードおよびバイトコードのバージョン、デバッグ情報の有無、最適化の有無などを指定します。

<plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-compiler-plugin</artifactId> <version>3.8.1</version> <configuration> <encoding>UTF-8</encoding> <source>${jdk.version}</source> <target>${jdk.version}</target> <debug>true</debug> <optimize>true</optimize> <showDeprecations>true</showDeprecations> </configuration></plugin>

maven-bundle-plugin プラグインは、JARファイルにOSGiマニフェストを追加し、OSGiバンドルを生成します。OSGiバンドルとは、META-INF/MANIFEST.MFファイルにOSGiマニフェストが追加されたJARファイルを指します。

<plugin> <groupId>org.apache.felix</groupId> <artifactId>maven-bundle-plugin</artifactId> <version>5.1.4</version> <extensions>true</extensions> <configuration> <instructions> <Bundle-SymbolicName>com.logpresso.sonar.sample</Bundle-SymbolicName> <Export-Package> com.logpresso.sonar.sample;version=${project.version}, </Export-Package> <Import-Package> org.json;version="1.1.0", org.araqne.codec;version="2.2", org.araqne.log.api;version="3.13.0", org.logpresso.api.profile;version="1.1.0", org.logpresso.api.profile.query;version="1.1.0", org.araqne.logdb;version="3.10.0", org.araqne.msgbus;version="1.12.0", com.logpresso.sonar.api.\*;resolution:=optional, \* </Import-Package> <Private-Package> com.logpresso.sonar.sample.impl, com.logpresso.sonar.sample.query, </Private-Package> </instructions> </configuration></plugin>

<instructions> の4つの項目について説明します：

* Bundle-SymbolicName: バンドル識別子を定義します。Javaパッケージ名の規則に従って記述してください。
* Export-Package: 他のOSGiバンドルに公開するパッケージ一覧を指定します。通常はインターフェースを含むパッケージのみを公開します。${project.version} マクロを利用して公開パッケージのバージョンを明記します。
* Import-Package: 他のOSGiバンドルからインポートするパッケージ一覧を指定します。パッケージ名の後ろに ;version="VERSION" を付与して最小バージョンを明記します。バージョンを指定しない場合、ビルドプラグインは <dependencies> で定義されたMavenアーティファクトのバージョンを使用するため、意図しないバージョン範囲が指定され、バンドル依存性が解決できない場合があります。
* Private-Package: 他のOSGiバンドルに公開しない内部パッケージ一覧を指定します。通常、実装を含むパッケージは公開しません。外部にはインターフェースのみを公開することで、実装の詳細をいつでも容易に変更できるようにします。

Import-Package には、ログプレッソ・ソナーアプリAPI呼び出しに必要なパッケージのバージョンがあらかじめ定義されています。sonar-app-api ライブラリのバージョン（現時点では4.0.2306.0）によって、利用可能なパッケージバージョンのセットが変わる場合があります。

<plugin> <groupId>org.apache.felix</groupId> <artifactId>maven-ipojo-plugin</artifactId> <version>1.12.1.asm8</version> <executions> <execution> <goals> <goal>ipojo-bundle</goal> </goals> </execution> </executions></plugin>

iPOJOは、宣言的にOSGiコンポーネントを開発できるようにするフレームワークです。OSGiはランタイムプラグインを前提としているため、各バンドルが依存する機能はいつでも新規インストールや削除が可能です。そのため、OSGiインターフェースのみで開発する場合、すべての状態変化に対して通知を受け、連鎖的に機能を無効化・有効化する必要があり、実装が非常に複雑になります。iPOJOはコンポーネントのライフサイクルモデルを定義し、簡単なアノテーション追加のみでOSGiサービスを宣言的に開発できるよう支援します。

iPOJOはビルド時にiPOJOアノテーションを認識し、バイトコードを挿入することで自動変換を行います。そのため、上記のようにビルド時に maven-ipojo-plugin が必ず呼び出される必要があります。

#### ライブラリ依存関係

<dependencies> <dependency> <groupId>junit</groupId> <artifactId>junit</artifactId> <version>4.8.1</version> <scope>test</scope> </dependency> <dependency> <groupId>org.apache.felix</groupId> <artifactId>org.apache.felix.ipojo.annotations</artifactId> <version>1.12.1.asm8</version> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.7.12</version> </dependency> <dependency> <groupId>com.logpresso.sonar</groupId> <artifactId>sonar-app-api</artifactId> <version>4.0.2306.0</version> </dependency></dependencies>

ログプレッソ・ソナーアプリでは、以下のライブラリを使用します：

* junit: ユニットテスト作成に使用します。ユニットテストクラスは src/test/java ディレクトリ配下に配置してください。
* org.apache.felix.ipojo.annotations: iPOJOコンポーネントアノテーションを利用します。
* slf4j-api: システムログ記録にSLF4Jを使用します。ログは log/araqne.log ファイルに記録され、ログレベルはランタイムで調整可能です。
* sonar-app-api: ログプレッソ・ソナーアプリAPIを利用します。このライブラリのバージョンは、対応するログプレッソ・ソナープラットフォームのバージョンを示します。