

Dismiss

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Sign up

PMML evaluator library for the Android operating system (<http://www.android.com/>)

📄 5 commits

🌿 1 branch

📦 0 releases

👤 1 contributor

📄 AGPL-3.0

Branch: master ▾

New pull request

Find file

Clone or download ▾

 vruusmann

Updated JPMML-Evaluator dependency. Fixes #2

Latest commit 9e9fe07 on 7 Mar

📁 pmml-android-example	Updated JPMML-Evaluator dependency. Fixes #2	9 months ago
📁 pmml-android	Updated JPMML-Evaluator dependency. Fixes #2	9 months ago
📄 LICENSE.txt	Import	2 years ago
📄 README.md	Updated JPMML-Evaluator dependency. Fixes #2	9 months ago
📄 pom.xml	Import	2 years ago

📖 README.md

# JPMML-Android

PMML evaluator library for the Android operating system (<http://www.android.com/>).

## Features

- Full support for PMML specification versions 3.0 through 4.3. The evaluation is handled by the [JPMML-Evaluator] (<https://github.com/jpmml/jpmml-evaluator>) library.

## Prerequisites

- Android 3.0 or newer.

## Installation

Enter the project root directory and build using [Apache Maven] (<http://maven.apache.org/>):

```
mvn clean install
```

The build produces two files:

- pmml-android/target/pmml-android-1.0-SNAPSHOT.jar - Library JAR file.
- pmml-android-example/target/pmml-android-example-1.0-SNAPSHOT.apk - Example App APK file.

## Usage

Android does not provide its own JAXB runtime, and is generally unsuitable for working with standard JAXB runtimes such as GlassFish Metro or EclipseLink MOXy. Therefore, the `org.dmg.pmml.PMML` instance has to be obtained by other means.

The suggested workaround is to transmit models using Java Serialization.

Models can be converted from PMML data format to Java Serialization (SER) data format using the JPMML-Model Maven plugin:

```
<plugin>
  <groupId>org.jpmml</groupId>
  <artifactId>pmml-maven-plugin</artifactId>
  <version>${pmml-model.version}</version>
  <executions>
    <execution>
      <phase>process-resources</phase>
      <goals>
        <goal>ser</goal>
      </goals>
      <configuration>
        <modelSets>
          <modelSet>
            <dir>src/main/pmml</dir>
            <outputDir>target/generated-sources/combined-assets</outputDir>
          </modelSet>
        </modelSets>
        <visitorClasses>
          <visitorClass>org.jpmml.model.visitors.StringInterner</visitorClass>
        </visitorClasses>
      </configuration>
    </execution>
  </executions>
</plugin>
```

The App can use the `org.jpmml.android.EvaluatorUtil#createEvaluator(InputStream)` utility method to create a model evaluator instance based on a streamable SER asset or resource:

```
public ModelEvaluator<?> loadSer(String serName) throws Exception {
    AssetManager assetManager = getAssets();

    try(InputStream is = assetManager.open(serName)){
        return org.jpmml.android.EvaluatorUtil.createEvaluator(is);
    }
}
```

Extra caution is required to ensure that the SER producer (ie. the JPMML-Model Maven plugin) and the SER consumer (ie. the JPMML-Android library) are using exactly the same version of the JPMML-Model library at all times.

## License

JPMML-Android is licensed under the [GNU Affero General Public License (AGPL) version 3.0] (<http://www.gnu.org/licenses/agpl-3.0.html>). Other licenses are available on request.

## Additional information

Please contact [[info@openscoring.io](mailto:info@openscoring.io)] (mailto:[info@openscoring.io](mailto:info@openscoring.io))