|  |  |
| --- | --- |
| TAK ML MX Plugin Development  Model Execution plugin development with TAK ML  June 8th, 2023 |  |

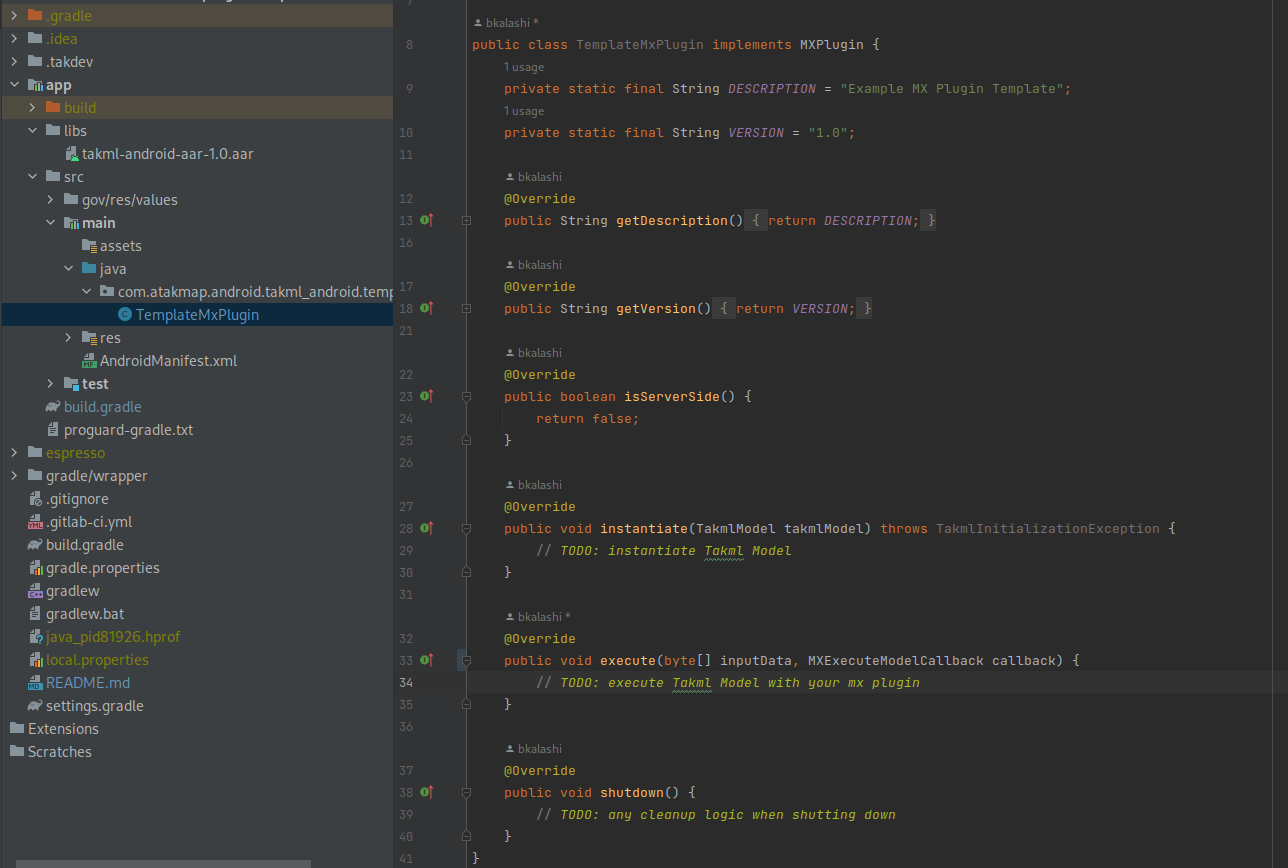
**Definition**

The MX Plugin, or Model Execution Plugin, is a plugin that TAK ML uses to execute ML models. The MX Plugin is deployed as an AAR (Android Archive), which is analogous to a Jar in the Java world. MLA Plugins (Machine Learning Application) add these MX Plugin AAR dependencies, along with the TAK ML AAR dependency, to use TAK ML. Mx Plugins are discovered via a MX Plugin Token and instantiated through Reflection. Development is explained below.

**1. Create the Android ATAK Project**

The **TAK ML MX Plugin Template** can be used as a starting point. Currently ATAK 4.8.1 is supported, but in future releases other versions will be supported.

Each TAK ML MX Plugin is defined via implementing the MxPlugin interface. In the TAK ML Mx Plugin Template, you will find a TemplateMxPlugin.java class:



**Interface Method Descriptions**

* **getDescription()**: Returns a String representing a friendly description of what the plugin does.
* **getVersion()**: The version of the Mx Plugin.
* **isServerSide()**: Whether the plugin operates in client side or server side. Note, only client side is supported at this time.
* **instantiate(TakmlModel takmlModel)**: Instantiates of a TAK ML model with the Mx Plugin. This is invoked when a TakmlExecutor is instantiated with with a TAK ML Model, or when the TAK ML Model is changed.
* **execute(byte[] inputData, MXExecuteModelCallback)**: Invoked when TakmlExecutor executes the model with input data. The MXExecuteModelCallback provides a callback for the output of the model and result of execution.
* **shutdown():** Called when TAK ML is shut down. Any cleanup logic occurs here.

**2. Create a MX Plugin Token**

A token must be generated so that the MX Plugin can be found via TAK ML MLA plugins. Please use the accompanying generateToken.sh found in the TAK ML SDK.

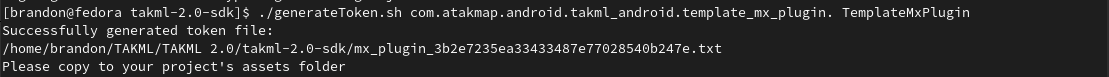
To use, run:

|  |
| --- |
| ./generateToken.sh [full classname of your MX Plugin] |

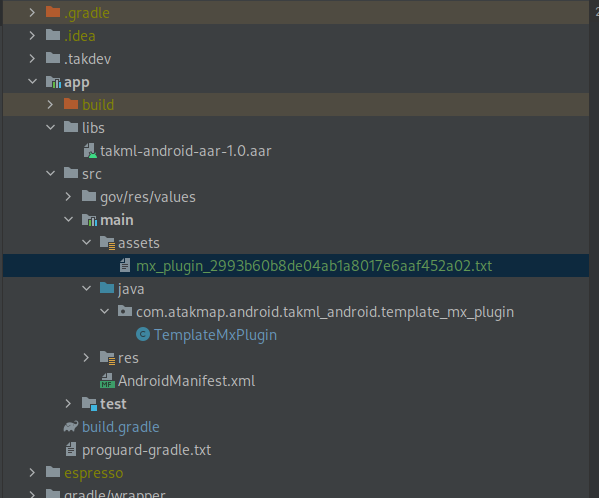
e.g.:

|  |
| --- |
| ./generateToken.sh com.atakmap.android.takml\_android.template\_mx\_plugin. TemplateMxPlugin |

An MX Plugin token txt file will be generated starting with “mx\_plugin\_”:



Copy this token to your Mx Plugin’s Assets folder:



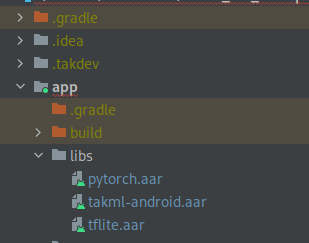
**3. Building an AAR**

To build for ATAK Civ:

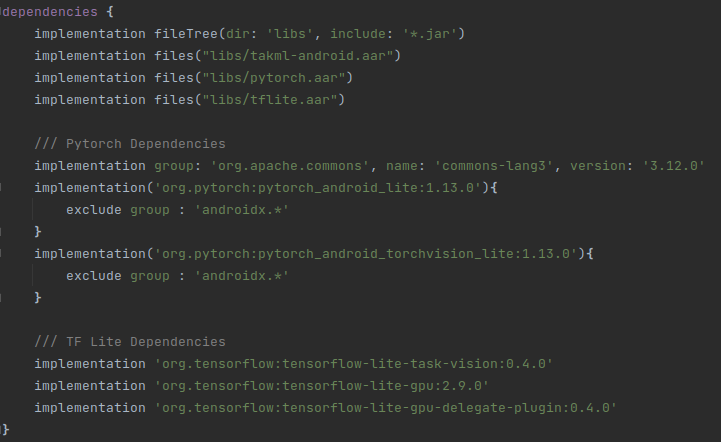
|  |
| --- |
| ./gradlew clean assembleCivDebug |

Or ATAK Mil:

|  |
| --- |
| ./gradlew clean assembleMilDebug |

The AAR can now be added to an MLA plugin. An MLA plugin also requires the TAK ML AAR. For example, the following MLA plugin has the TAK ML AAR and two MX Plugins (Pytorch and Tflite) as dependencies:  


With their build.gradle dependencies:



TAK ML will find the Mx Plugins automatically upon instantation.

**4. Document**

The dependencies need to be clearly outlined so MLA plugin developers can add them to their projects when using your MX Plugin.

Finally, the inputs and outputs of the MX Plugin should be documented. For example, does the MX plugin support Image Classification or Object Detection. What is the type of expected model (e.g. Torchscript).