



Mobile / Tools Developer Test

(1) Purpose

The Mobile / Tools Developer is responsible for creating native Android and iOS plugins that integrate with Mobile Flash Applications via ANE or Unity. We do not expect that you have all of these languages and concepts learnt by heart, but instead that you are able to learn, read and understand code.

(2) Criteria

Your code will be evaluated using the following criteria:

- Code structure, for instance, using SOLID principles when possible;
- Code convention, always considering the specific language that you are working on;
- Performance;
- Multiplatform support, as your applications could be compiled/interpreted in many OS.

(3) Android module and application (**required**)

3.1. Android Module

Create an Android module that will open the following external applications:

- WhatsApp, in the message view, for a particular number;
- WhatsApp, in the message view, for a particular number, with a predefined message;
- Telegram, in the message view, for a particular number/userId;
- Facebook Messenger, in the message view, for a particular number/userId;
- Skype, in the message view, for a particular user name.



If the application is not installed in the phone, it should redirect to the app Google Play Store page. This module should be developed and built externally from any application; it should not contain any user interface. The build should produce an aar file. Use only official APIs / techniques to open third-party applications.

3.2. Android Test application

Create an application to import and test the aar created before. There are no specific requirements for the user interface, a set of buttons will do just fine, but if you want to go for the extra mile, great!

(4) Build automation (optional)

Create a build automation script to build the aar module and copy it to the Android application project. After that, the script should build the Android application as an apk and install it to a connected device, if any.

You may use Powershell, bash or Python (preferred version 3.x).

(5) Artifacts

Your projects should be pushed to a git repository, using the following folder structure:

- README.md
- Android/module
- Android/application
- Python
- Unity