CardConnect Consumer SDK Integration Guide

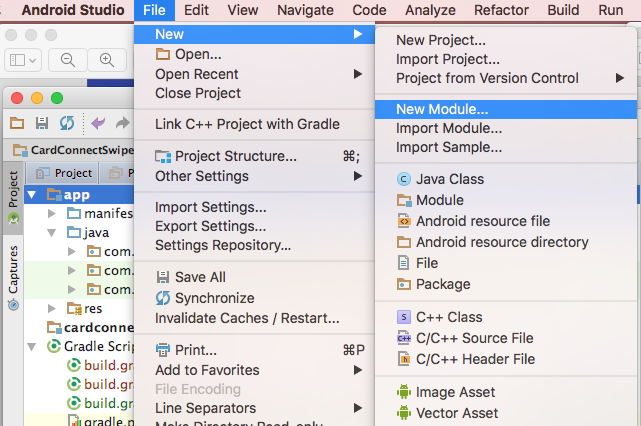
Purpose of the SDK

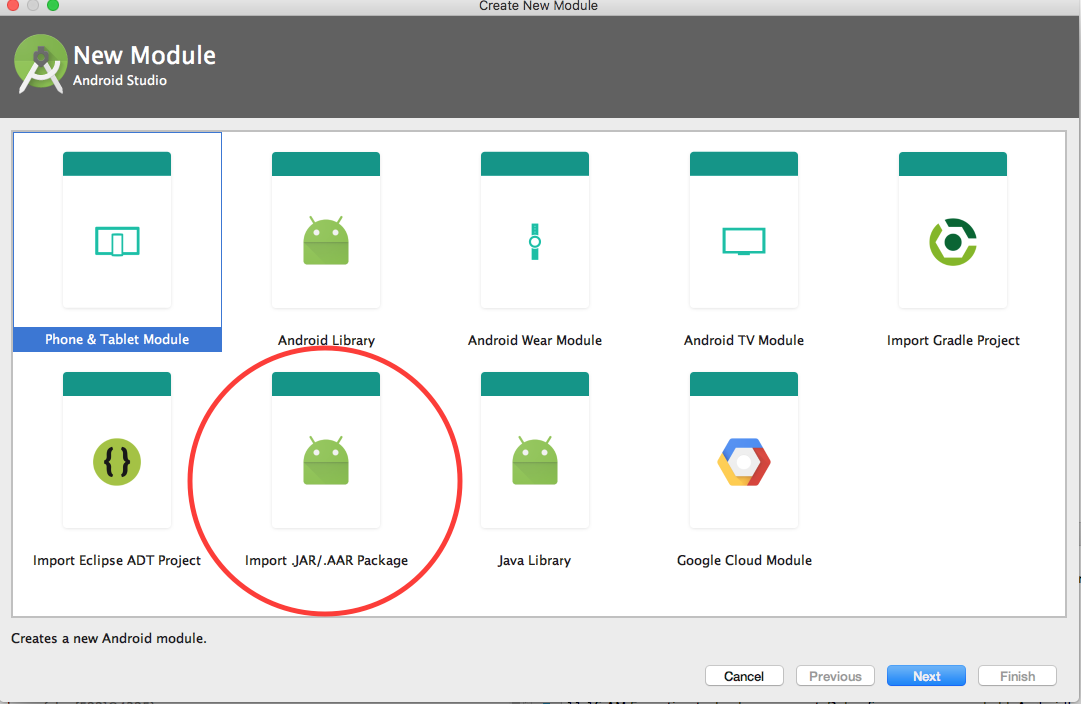
* The CardConnect Consumer SDK enables application to tokenize payment information data through CardConnect’s card secure gateway and provides useful custom user interface components with formatting and validation options for payment information.
* For tokenizing User Information entry methods are Magnetic Strip Read (MSR), and manual card entry.
* Allows Android Pay integration.

List of Included Files: ccconsumersdk-consumerSwiper-release.aar, Javadocs, demo application, Integration Guide.docx, Implementation Guide.docx and Android Pay Integration Guide.docx.

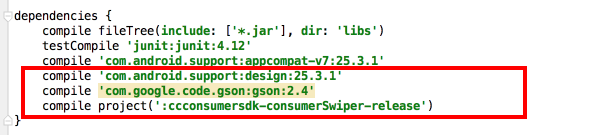
Importing SDK into Payment App

Open Android Studio and go to the project containing the payment app that will utilize the SDK. The SDK will be imported into the project as a new module constructed from the AAR file (ccconsumersdk-consumerSwiper-release.aar). To create the new module, go to the menu and click File->New->New Module->Import .JAR/.AAR Package. Locate the ccconsumersdk-consumerSwiper-release.aar and use confirm the selection.





Once the module has been created from the AAR file, the next part of the installation is to make new newly created module available to the main payment application. To do so, locate the build.gradle file pertaining to the application and add the newly created module as an additional dependency in the dependencies block. The GSON and design support library dependency will also be added if it hasn’t been as is required by the sdk.



Once those steps are completed, classes from the new module will be available to use inside of the payment application after you sync or rebuild the project.

If the SDK has been installed successfully into the current project, the directory structure in Android Studio should look like this. Notice that there is a build.gradle file created for the newly added module. Also note that the settings.gradle file has an entry for both the main project and the newly added module. Each submodule will have an entry in the settings.gradle file to tell the build system that the module is now available.

