For Android Applications

Contents

Introduction

The following describes how to run the ITL REST server using the Android_REST_API_Vx.x.x.apk and make endpoint calls to it and receive response data. There are many different Android versions available. This documentation has been compiled using Android Version 12 and version 1.3.2 of the Android REST API.

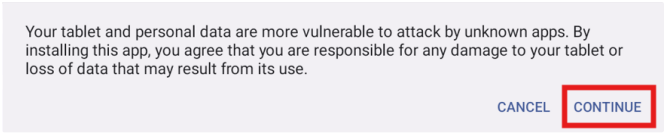
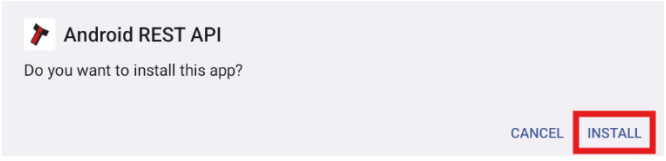
Pre-requisites

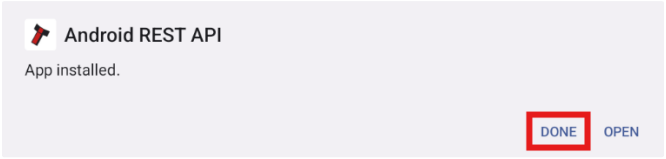
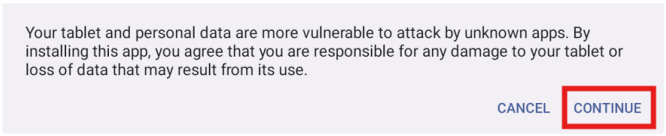
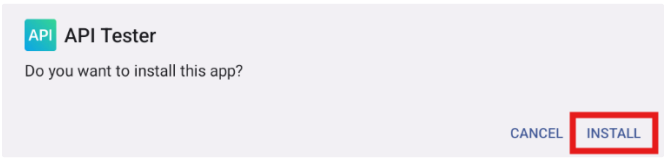
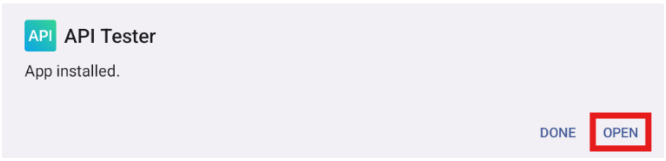
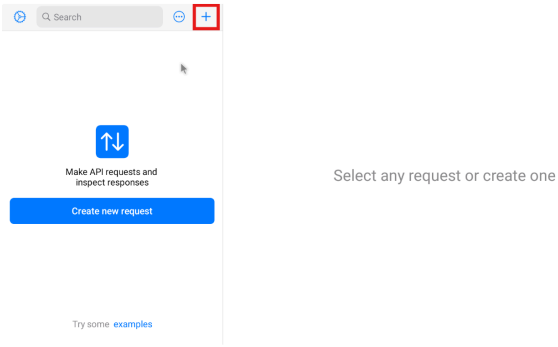
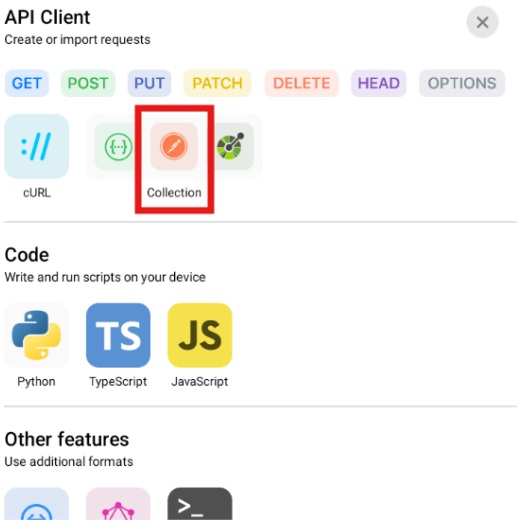
- Android Device, Android instance on a Virtual Machine or Emulator

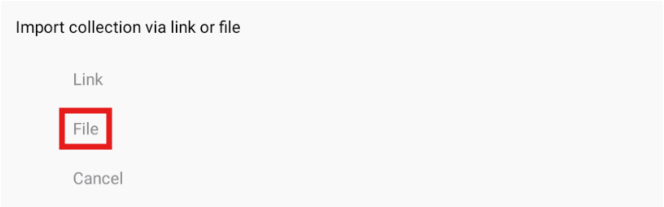
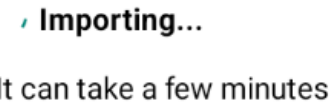
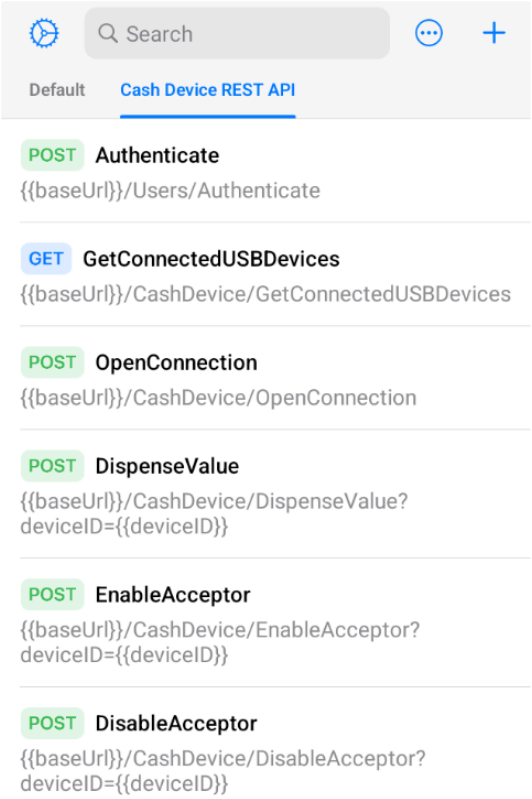
Minimum Android Version 10 (API 29)

- ITL cash handling device

Download and install the required resources

Navigate to the following link to download the latest SDK.	ITL SDK Package
Installing the Android_REST_API_Vx.x.x.apk Navigate to the APK download and launch the file.	
Select to CONTINUE on any warning messages.	
Select INSTALL to begin the process.	

Select DONE once the install completes.	
Download and install an API Tester like the following example:	API Tester
Installing the API Tester. Navigate to the APK download and launch the file.	
If sideloading the APK, select to CONTINUE on any warning messages.	
Select INSTALL to begin the process.	
Select OPEN to launch the application.	
Import the Collection to the API Tester. Select the + icon.	
Select Collection.	

Select File.	
Navigate to the saved collection file Android-API-Collection.json and select the file.	
Wait until the import completes.	
Once the import has completed you will find the collection list on the left of the API Tester app.	

Running the ITL REST Server

Launch the Android REST API application	
The server will launch and begin listening on http://127.0.0.1:5000/api/CashDevice	

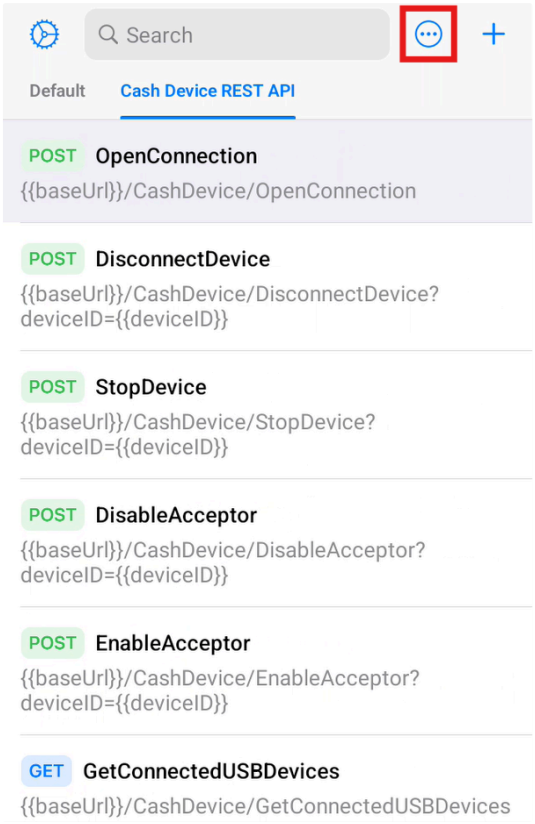
REST requests are case sensitive. Ensure upper case letters are used where advised.

The Android REST API application must remain active at all times. Closing the application will stop the API from communicating between your software and the cash device. A fix has been implemented from version 1.3.2 to ensure the server remains active until explicitly closed, however some Android versions can pause or stop apps running in the background e.g. switching between the server app and the API Testing app. Ensure your Android version can allow apps to remain active in the background.

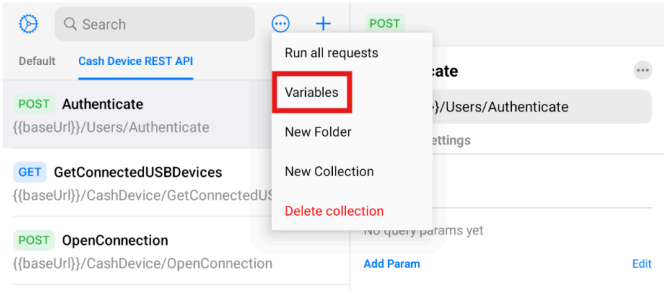
Sending REST Requests using API Tester

Set the baseURL in the Variables table.

To access the Variables table, select the 3 dots contained in the circle to the right of the search bar.

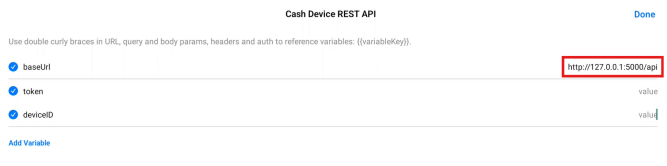


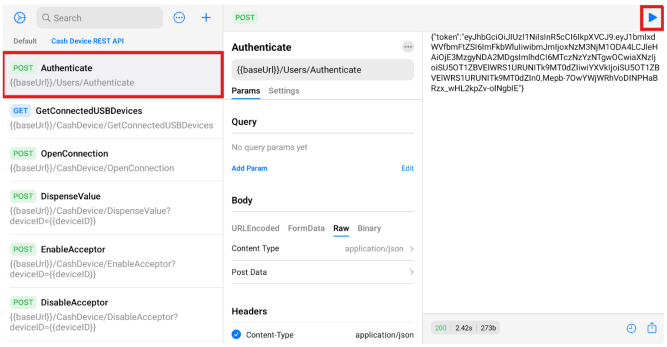
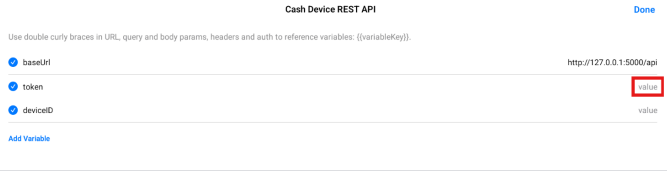
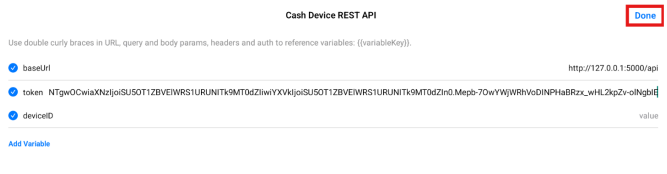
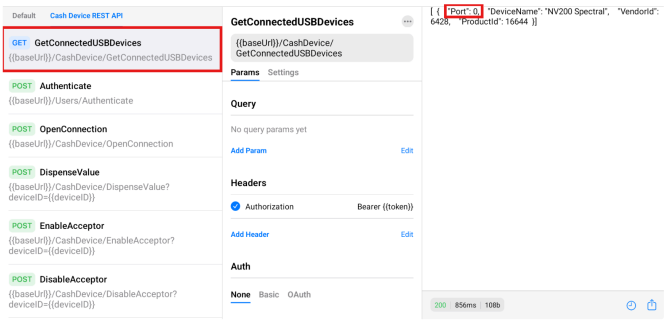
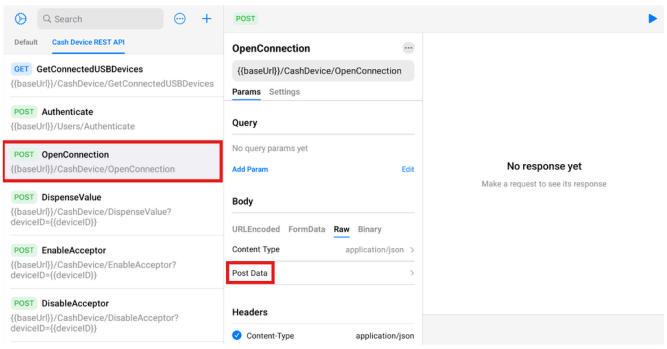
Select Variables.

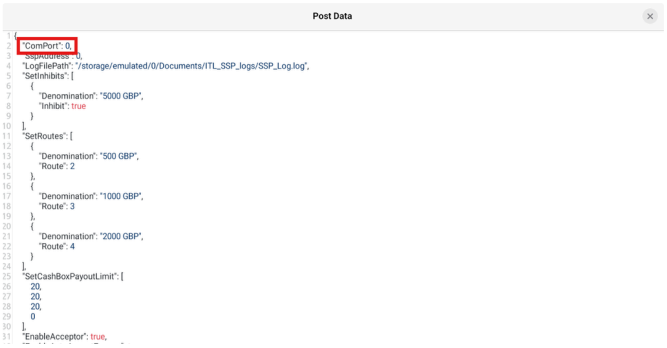
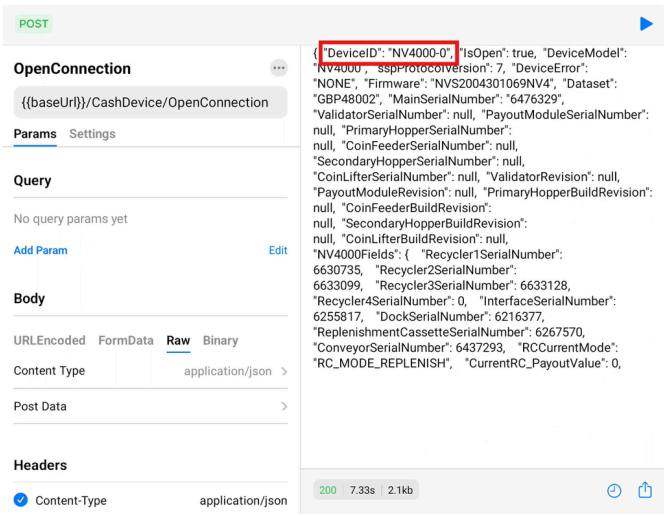


Ensure the correct baseUrl is set in the variables table.

```
http://127.0.0.1:5000/api
```



Select Done to update the table and return to the request list.	
Request a Bearer Token for authorisation of upcoming REST requests using the Authenticate request. Select the Authenticate request and select the play button from the top right to send the request.	
Copy the Bearer Token to the Variables table for future requests. Highlight the Bearer Token and select Copy.	
Paste the Bearer Token into the token value field of the variables table.	
Select Done to update the table and return to the request list.	
Send the GetConnectedUSBDevices request to obtain the correct Port.	
Navigate to the Post Data for the OpenConnection request.	

Update the Post Data to the correct Port returned for the GetConnectedUSBDevices request.	
Send the OpenConnection request once the Post Data has been updated. Please refer to the OpenConnection details for further information on the full Post Data body.	
Update the deviceID in the variables table. The OpenConnection request will return the correct deviceID to use on subsequent requests.	
Save the deviceID to the variables table.	

Your validator will be enabled and ready to accept currency if the below parameters of the OpenConnection request are set to true.
 	
`"EnableAcceptor": true,`
`"EnablePayout": true`	


```
"EnableAcceptor": true,
"EnablePayout": true
```

SSP Lower-Level Logging

Available from version 1.3.2

Android will store the Lower level SSP logs from version 1.3.2.

By default, logs are stored at the following path:
`/storage/emulated/0/Documents/ITL_SSP_logs/{default_log_name}.log`

If you wish to use a custom log location, simply specify it in the JSON request using the following format:
`"LogFilePath": "{directory}/{log_file_name}.log"`

To access the default logs on your Android device, navigate to:
Files -> Internal Storage -> Documents