

# uFCoder HTTP Service v1.0

# Table of contents

<b>About</b>	<b>3</b>
<b>Disclaimer</b>	<b>3</b>
<b>Renaming the application for commercial use</b>	<b>3</b>
<b>Usage</b>	<b>6</b>
HTTP example - Get token	6
HTTP example0 - APDU commands	7
Starting service from another application	8
<b>Revision history</b>	<b>9</b>

## About

uFCoder library as a HTTP service example can be found here:

[https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-lib\\_http\\_service.git](https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-lib_http_service.git)

## Disclaimer

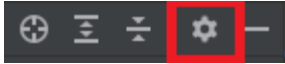
For the commercial use of uFCoder HTTP service, company & package name **must** be renamed.

Default package name “com.dlogic.ufrwebstarter” is reserved for our example only.

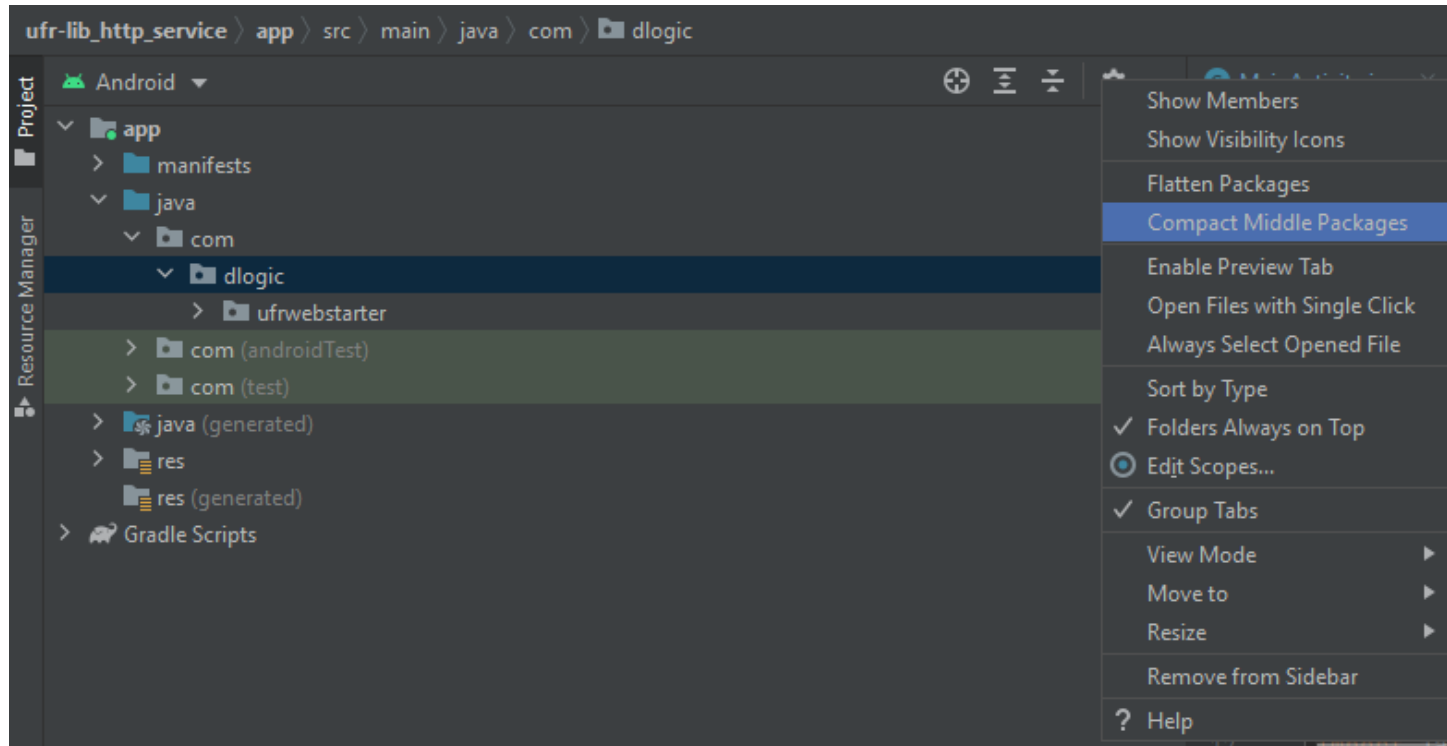
To do so, please follow the steps in the [Renaming the application for commercial use](#) section.

## Renaming the application for commercial use

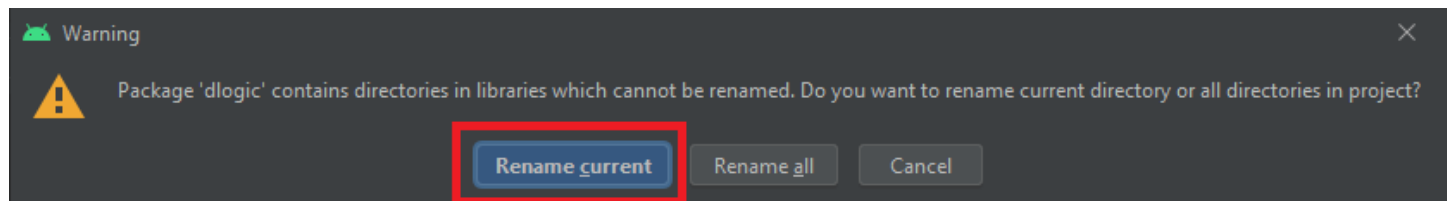
1. You can begin by downloading our example found in [this repository](#).
2. Load the project in Android Studio.

3. In the Project panel, click on the little gear icon  and uncheck “Compact Middle Packages” if it is checked (which it should be by default).

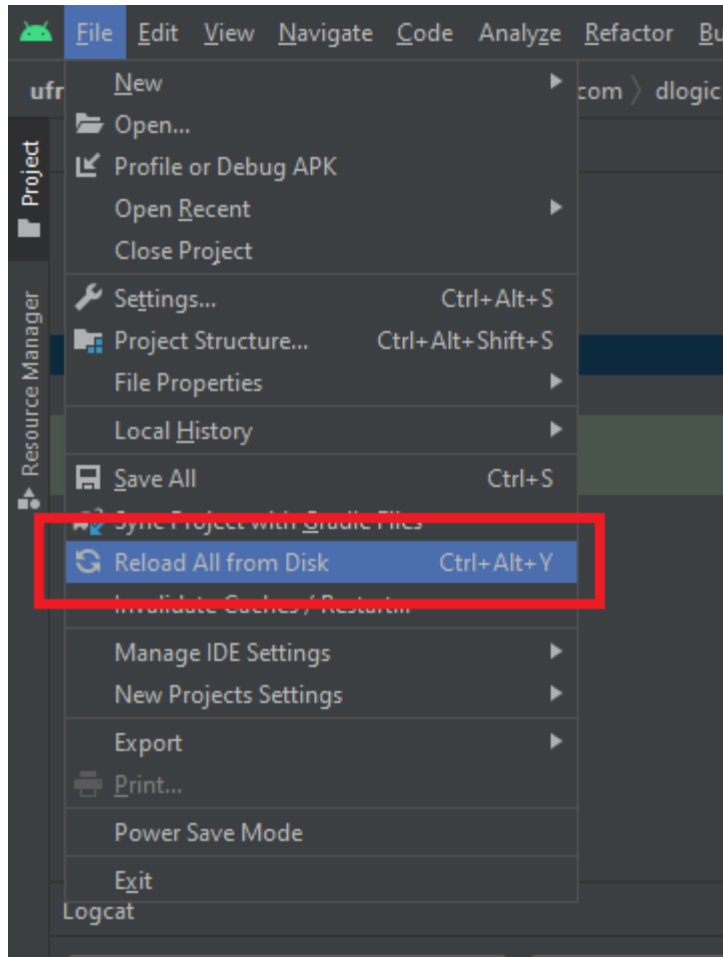
4. Now you should see Project structure as in the following image:



5. Right click on the "dlogic" folder, navigate to "Refactor" and in the dropdown menu select "Rename..." (Shift + F6)
6. (Optional) If you encounter a warning as the following here, simply click on "Rename current":



7. Reload All from Disk to see updated project structure (*CTRL+ALT+Y*):



8. **Important:** Before starting the service with a new package. Uninstall any other service with the default, or another name beforehand.

If you intend on having multiple services, keep in mind that the hardware will support one action at the time and may behave unpredictably. The source code provided in our git would also require further modification(s) such as setting the service to listen on a port different from the default one (default port in our example is 1234), and again, a different package name.

To change the port of the service open file "uFRWebStarter.java" and change the line 50, for example: `startServer(1234);` => `startServer(port_number);`

## Usage

Install *uFRHttpService.apk* and start one time. Reboot device. Service will be started automatically on device boot.

HTTP service endpoint: <http://ip-address:1234> (by default).

Available commands (send plain body text):

**ReaderOpenEx** 5 0 1 0 - NexGo contact card side slot

**ReaderOpenEx** 5 0 4 0 - NexGo PSAM1 slot

**ReaderOpenEx** 5 0 5 0 - NexGo PSAM2 slot

**ReaderOpenEx** 5 0 8 0 - Sunmi PSAM slot

**DL\_TLS\_Token** host path port pin

(e.g "DL\_TLS\_Token api.sandbox.suf.purs.gov.rs /api/v3/sdc/token 443 8440")

**APDU** "hexstring"

(e.g) APDU 00A4040009A00000003974254465900

**Restart** - Restarts service

## HTTP example - Get token

For this demonstration [Talend API Tester](#) was used.

1. Simply send "ReaderOpenEx" with necessary parameters via HTTP as plain text:

The screenshot shows the Talend API Tester interface. The METHOD is set to POST. The URL is http://192.168.1.210:1234. The BODY is set to ReaderOpenEx 5 0 1 0. The interface includes sections for HEADERS, QUERY PARAMETERS, and BODY, with a Send button in the top right corner.

Response should be in JSON format:

### Response

200 OK

#### HEADERS

Content-length: 44 bytes  
Content-type: application/json

▶ COMPLETE REQUEST HEADERS

pretty

#### BODY

```
{
  status: "[0x00 (0)] UFR_OK",
  response: ""
}
```

lines nums

2. Send "DL\_TLS\_TOKEN" with necessary parameters:  
Parameters in order are: URL, PATH, PIN.

METHOD: POST  
SCHEME://HOST[:PORT][PATH["?" QUERY]]  
http://192.168.1.210:1234

▶ QUERY PARAMETERS

HEADERS

+ Add header Add authorization

Form

BODY

```
1 DL_TLS_Token api.sandbox.suf.purs.gov.rs /api/v3/sdc/token 443 8440
```

## HTTP example0 - APDU commands

Sending APDU commands via uFCoder HTTP service is executed in the following manner:

1. Simply send "APDU" with the APDU command as a string following the keyword in plain text format.  
For example:

METHOD: POST  
SCHEME://HOST[:PORT][PATH["?" QUERY]]  
http://192.168.1.210:1234

▶ QUERY PARAMETERS

HEADERS

+ Add header Add authorization

Form

BODY

```
1 APDU 00A4040010A00000748464A492D546178436F726500
```

2. Response is received as JSON:

### Response

200 OK

#### HEADERS <sup>?</sup>

Content-length: 48 bytes  
Content-type: application/json

► COMPLETE REQUEST HEADERS

pretty ▼

#### BODY <sup>?</sup>

```
{  
  status: "[0x00 (0)] UFR_OK",  
  response: "9000"  
}
```

## Starting service from another application

To start uFR HTTP service from another application you need to use **com.dlogic.ufrwebstarter.uFRWebService** intent.

Example code:

```
Intent i = new Intent();  
i.setComponent(new  
ComponentName("com.dlogic.ufrwebstarter",com.dlogic.ufrwebstarter.uFRWebService));  
getApplicationContext().startService(i);
```



## Revision history

Date	Version	Comment
2019-10-31	1.0	Base document