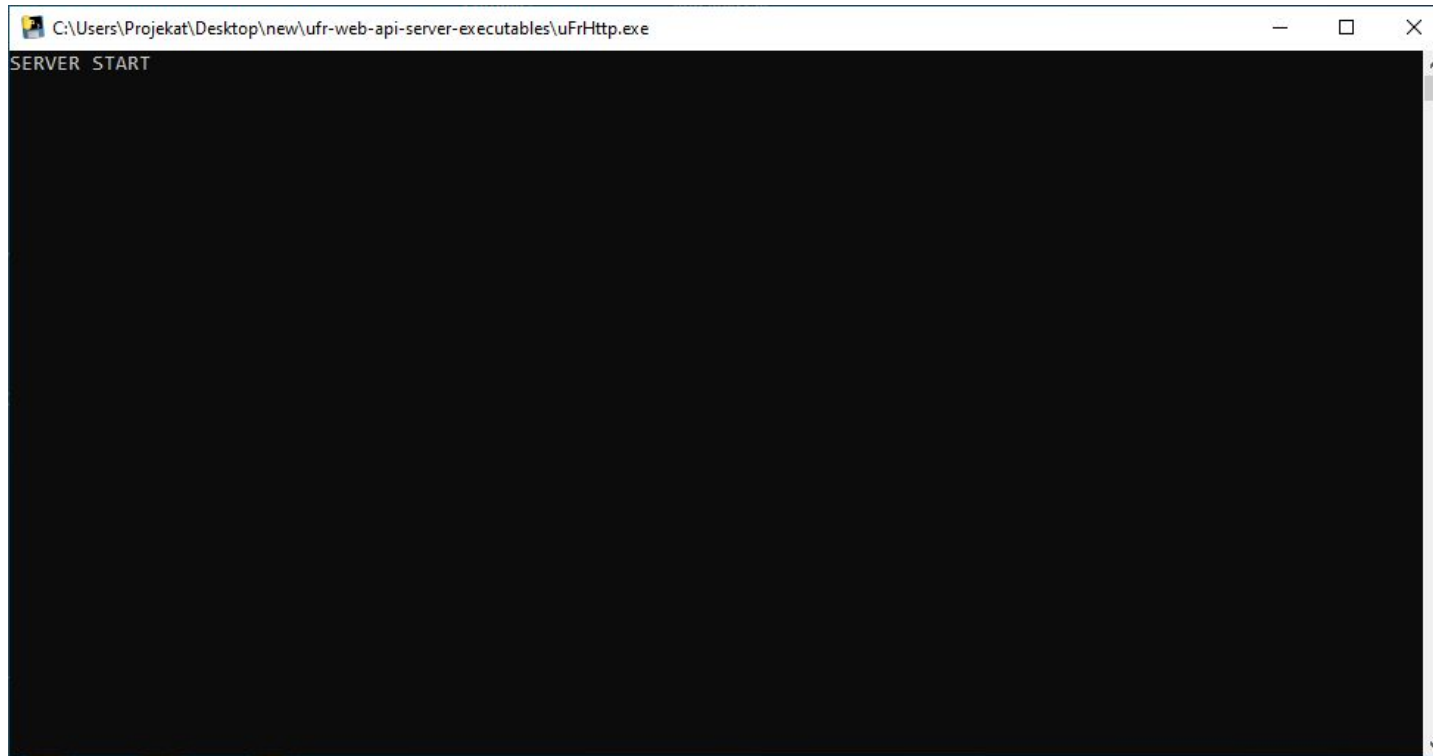


uFR Web API Javascript example

1.0

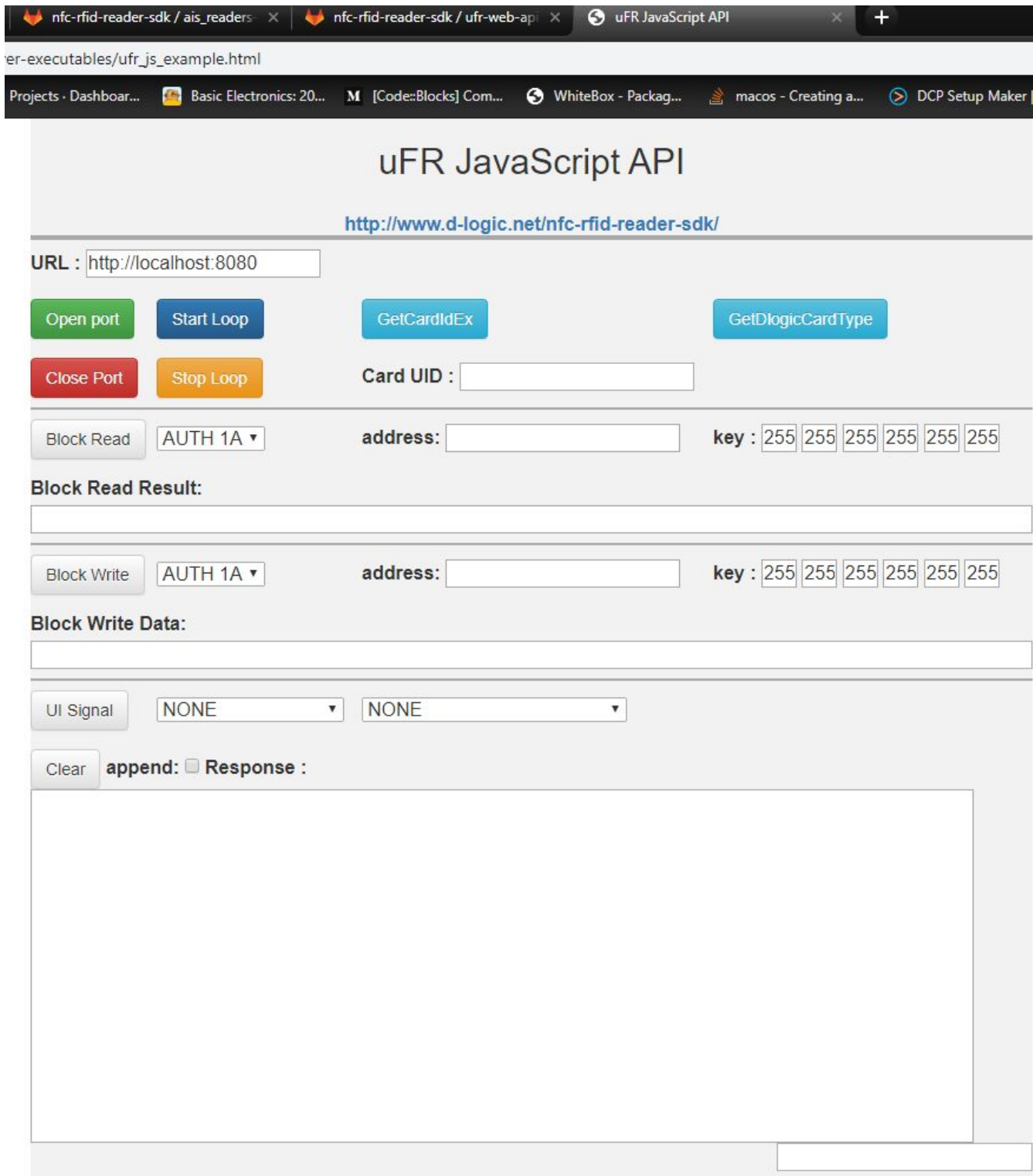
About software

uFR NFC Reader Cross Platform Server is designed to communicate with uFR Reader and WEB Based Applications. As such: simply start the 'uFrHTTP.exe' as a mandatory requirement for working with uFR series readers via front-end.



Successful server start will prompt 'SERVER START'

Afterwards, open the 'ufr_js_example.html' file and you will see the following in your default browser.



uFR JavaScript API

<http://www.d-logic.net/nfc-rfid-reader-sdk/>

URL :

Card UID :

AUTH 1A ▼ address: key :

Block Read Result:

AUTH 1A ▼ address: key :

Block Write Data:

UI Signal NONE ▼ NONE ▼

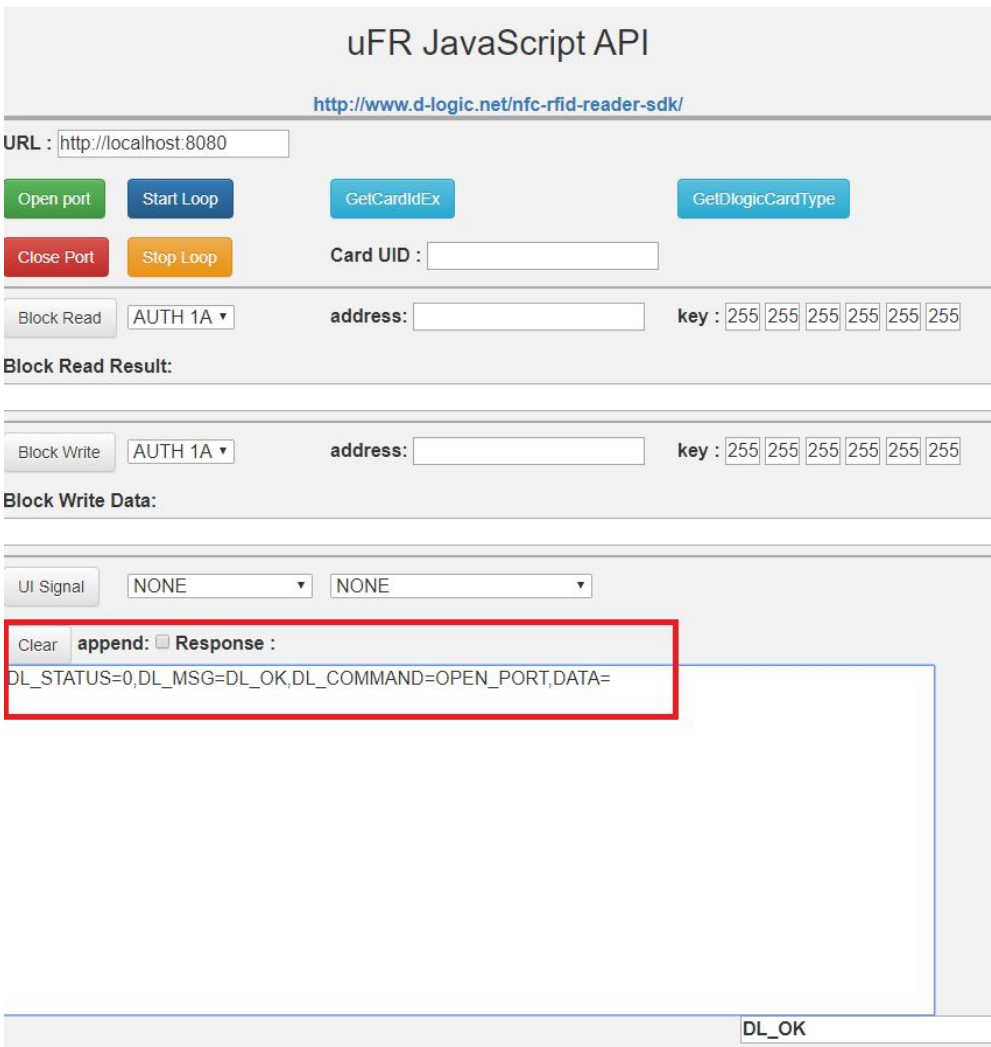
append: Response :

With reader connected, via front-end every function in this example will be sending HTTP requests as commands to the HTTP server we started beforehand.

Once a request has been received, in console window of the server - following will appear:

```
C:\Users\Projekat\Desktop\new\ufr-web-api-server-executables\uFrHttp.exe
SERVER START
127.0.0.1 - - [05/Sep/2019 17:05:25] "POST / HTTP/1.1" 200 -
```

Which means our server got the HTTP request, and on our front-end we will see it's response. Such as:



The screenshot shows the uFR JavaScript API web interface. At the top, it says "uFR JavaScript API" and provides the URL "http://www.d-logic.net/nfc-rfid-reader-sdk/". Below this, there is a "URL" input field containing "http://localhost:8080". There are several buttons: "Open port" (green), "Start Loop" (blue), "GetCardIdEx" (light blue), "GetDlogicCardType" (light blue), "Close Port" (red), and "Stop Loop" (orange). There is a "Card UID" input field. Below that, there are "Block Read" and "Block Write" buttons, each with a dropdown menu set to "AUTH 1A". There are also "address" and "key" input fields, with the key field containing "255 255 255 255 255 255". There are two sections for "Block Read Result:" and "Block Write Data:". At the bottom, there are two dropdown menus for "UI Signal", both set to "NONE". A red box highlights the "Response" area, which contains the text "DL_STATUS=0,DL_MSG=DL_OK,DL_COMMAND=OPEN_PORT,DATA=". Below the response area, there is a "DL_OK" label.

Depending on the command sent, front-end will display data according to the functions sent to uFR Reader. For example, using 'GetCardIdEx' and then 'Block Read' function, it's results will be displayed as shown in next image:



The screenshot shows the 'uFR JavaScript API' interface. At the top, the URL is 'http://www.d-logic.net/nfc-rfid-reader-sdk/'. Below this, there is a 'URL' field set to 'http://localhost:8080'. There are several control buttons: 'Open port' (green), 'Start Loop' (blue), 'Close Port' (red), and 'Stop Loop' (orange). Two main function buttons are highlighted with red boxes and numbered: 'GetCardIdEx' (1) and 'GetDlogicCardType' (blue). Below these, the 'Card UID' field displays '04E217C28F4980'. The 'Block Read' section is highlighted with an orange box and numbered 2. It includes a 'Block Read' button, a dropdown menu set to 'AUTH 1A', an 'address' field set to '0', and a 'key' field with six '255' values. Below this, the 'Block Read Result' field displays the hexadecimal string '04 e2 17 79 c2 8f 49 80 84 48 00 00 e1 10 12 00'. The 'Block Write' section is also visible, with a 'Block Write' button, a dropdown menu set to 'AUTH 1A', an empty 'address' field, and a 'key' field with six '255' values. Below this, the 'Block Write Data' field is empty. At the bottom, there are 'UI Signal' dropdowns set to 'NONE', a 'Clear' button, and an 'append' checkbox. The 'Response' field displays the command output: 'DL_STATUS=0,DL_MSG=DL_OK,DL_COMMAND=BLOCK_READ_PK,DATA=04e21779c28f498084480000e1101200'. A 'DL_OK' status indicator is visible at the bottom right.

- 1 - GetCardIdEx function will display cards UID as shown
- 2 - Block Read function will display data of block specified in 'address' field using key to the right of the function for authentication.

All functions will return results and it's details will be always shown in appropriate field.

However, in this example, some options for Mifare Ultralight® and NTAG® are supported but is hidden while working with other types of cards. For example, if you put NTAG213 card in readers field, first press 'GetDlogicCardType' button and following will be displayed:



"Authentication options" shown here consist of the following:

- PWD/PACK key combination, used for authenticate options in this group including 'AUTH1A/B' combobox selection.
- NEW PWD/PACK key, new PWD/PACK key combination to be stored in reader, or in card with above mentioned PWD/PACK combination used as authentication. When entering new key in reader, in this example it will be always stored in reader key index 0.

- Card access options, used for determining access to the card such as read/write protection and page address from which password verification shall be required when card is either write protected or read/write protected.

These authentication options support following card types:

- Mifare Ultralight EV1®
- NTAG213
- NTAG215
- NTAG216

Otherwise, if 'GetDlogicCardType' function does not detect one of the supported types in readers field - these authentication options will not be shown.

Test rest of the functions in this demo and see how front-end example interacts with HTTP server.

Revision history

Date	Version	Comment
2019-04-09	1.0	Base document