

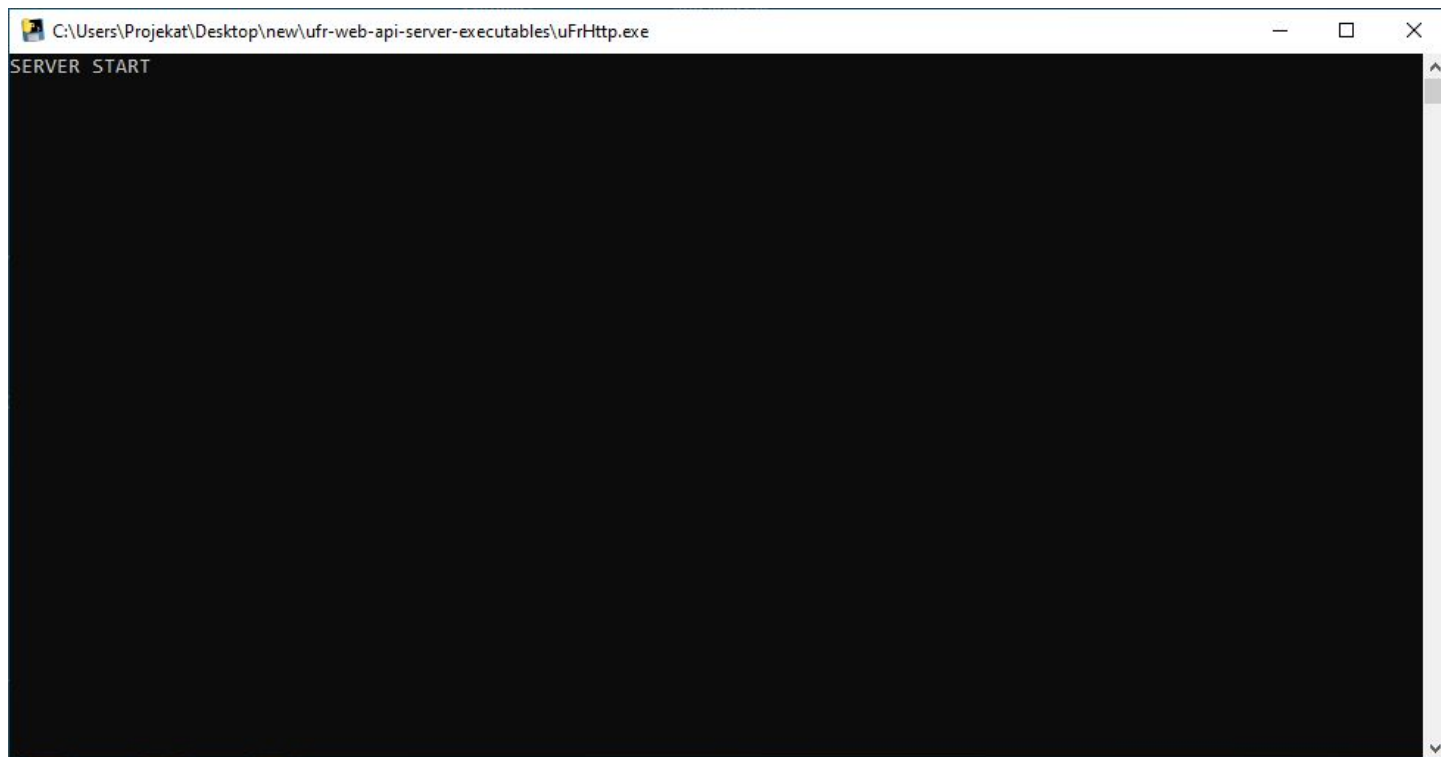
# uFR Web API Javascript example 1.0

# Table of contents

<b>About software</b>	<b>3</b>
<b>Revision history</b>	<b>8</b>

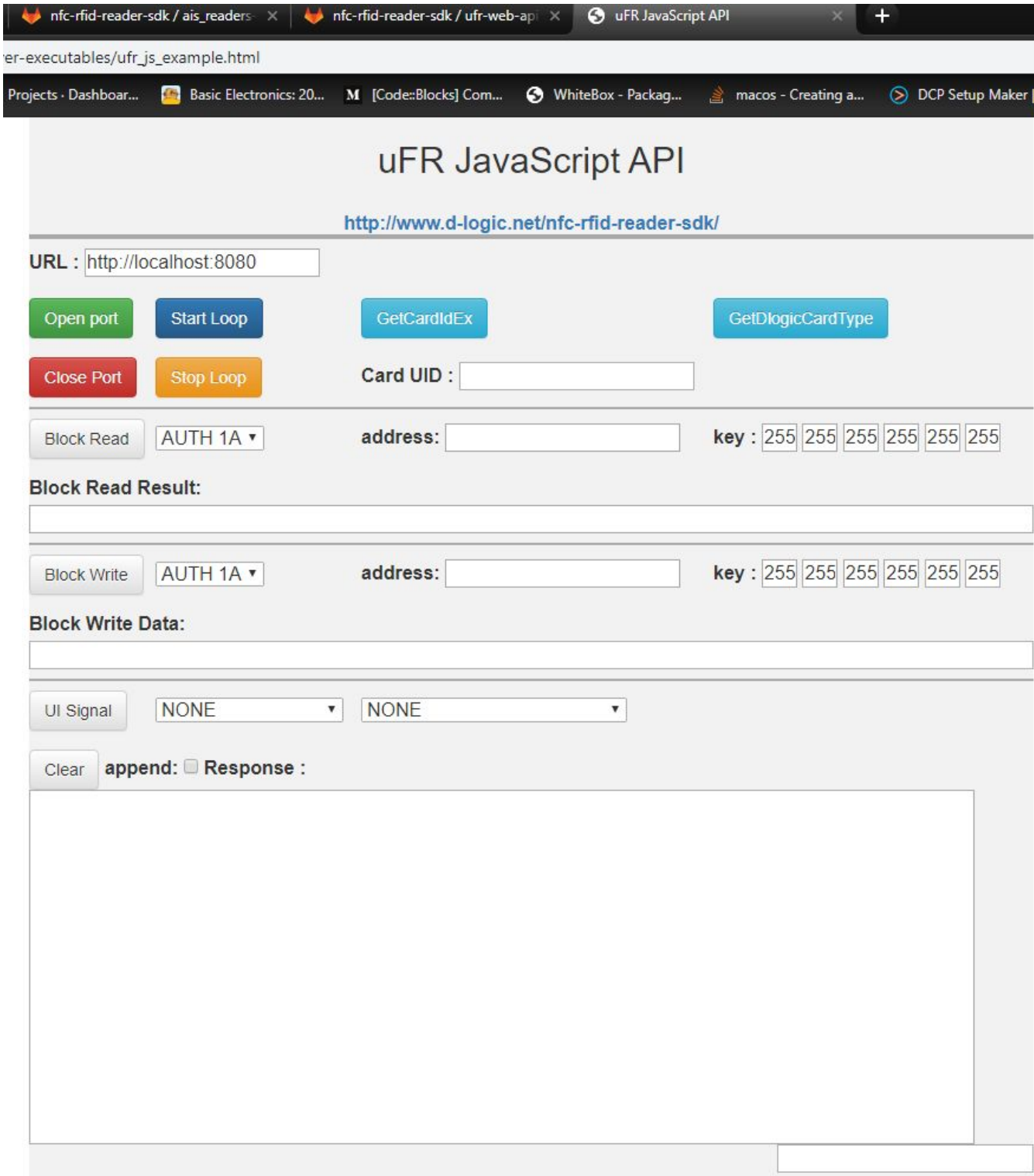
## About software

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 :

**Block Read Result:**

**Block Write Data:**

☐ 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:

### uFR JavaScript API

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

URL :

Open port
Start Loop
GetCardIdEx
GetDlogicCardType

Close Port
Stop Loop

Card UID :

Block Read AUTH 1A ▾      address:       key :

**Block Read Result:**

---

Block Write AUTH 1A ▾      address:       key :

**Block Write Data:**

---

UI Signal NONE ▾ NONE ▾

Clear    append: ☐ **Response :**

DL\_STATUS=0,DL\_MSG=DL\_OK,DL\_COMMAND=OPEN\_PORT,DATA=

DL\_OK

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:

### uFR JavaScript API

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

URL :

Open port
Start Loop
GetCardIdEx
GetDlogicCardType

Close Port
Stop Loop

Card UID :

Block Read

AUTH 1A

address:

key :

**Block Read Result:**  
 04 e2 17 79 c2 8f 49 80 84 48 00 00 e1 10 12 00

Block Write

AUTH 1A

address:

key :

**Block Write Data:**

UI Signal

NONE

NONE

Clear

append: ☐ Response :

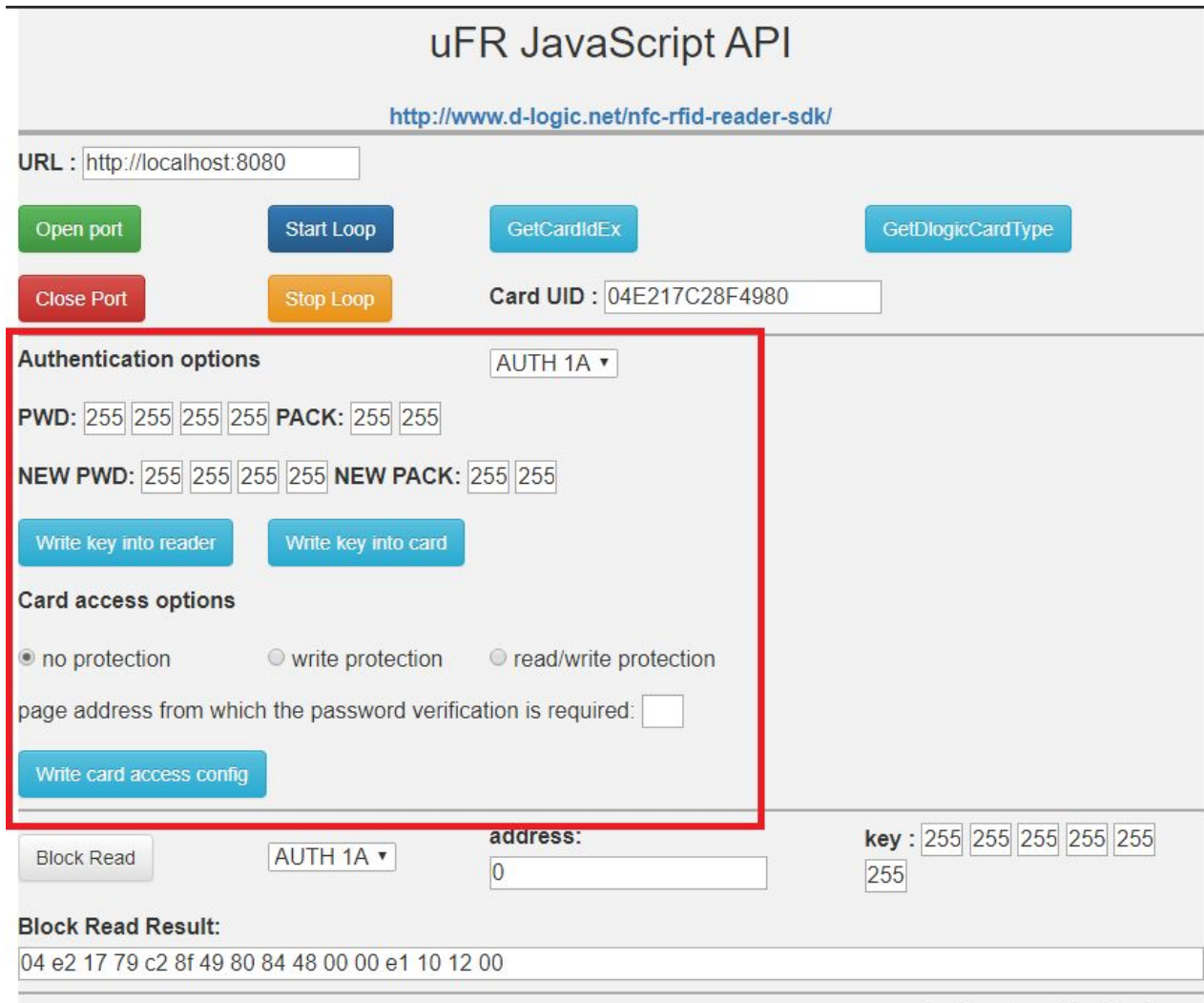
DL\_STATUS=0,DL\_MSG=DL\_OK,DL\_COMMAND=BLOCK\_READ\_PK,DATA=04e21779c28f498084480000e1101200

DL\_OK

- 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