

ADIF for LOTW

Инструкция

1. Как завести Лог для спутника QO-100,
 2. для 144МГц,
 3. для 432МГц,
 4. для 1296МГц
- и других KB и УКВ диапазонов аналогично,
5. конвертировать его в .ADIF для LOTW и QO-100 DX Клуба для спутника QO-100,
 6. конвертировать его в .ADIF для LOTW для 144МГц Тропо,
 7. конвертировать его в .ADIF для LOTW для 144МГц Авроры,
 8. конвертировать его в .ADIF для LOTW для 432МГц Тропо,
 9. конвертировать его в Cabrillo для теста Кубок Гагарина через QO-100,
 10. конвертировать его в Cabrillo для теста QO100 Challenge
- и других аналогично

А. Завести себе аккаунт на Github, как у меня: <https://github.com/eu2aa/eu2aa.github.io>
Зарегистрироваться, ввести свой Позывной как Login, Email, password, Ф.И. и мобильник.
Использую его в основном для хранения файлов изображений, .PDF и ZIP, для чего завел соответствующие директории.
Обычно я ухожу из Github без Sign out, т. е. без отключения.

Б. Завести себе аккаунт на Observable, как у меня: <https://observablehq.com/@eu2aa>
Пароль заводить не нужно, подходит пароль от Github. Вот почему из Github лучше НЕ ВЫХОДИТЬ. Из Observable, кстати, тоже.
Если все таки Github или Observable Вас «забыл», то жмем Sign in, а Ваш компьютер пароль подскажет.

1. Лог для спутника QO-100

Жмем <https://observablehq.com/@eu2aa/log-book>

В открывшейся странице вверх жмем на надпись **Fork**.

Обратите ВНИМАНИЕ, что **Вы скопировали** эту страницу себе и Вы находитесь уже на СВОЁМ Observable! Так можно поступать с любой страницей любого пользователя Github.

Редактируем первый текстовый блок.

Вверху, слева возле каждого **текстового блока** (при наведении мышью) появляются три точки.

Жмем, и в выпадающем меню выбираем Edit.

После этого **ниже** выбранного **текстового блока** появляется аналогичный текст, который можно редактировать.

В данном случае, текст

`md``

`# EU2AA DXCC QO-100`

`### Log Book`

,

редактируем **только** текст

`# EU2AA DXCC QO-100` - это ИМЯ Вашей странички

`### Log Book` — это заголовок

а первую строчку `md`` и запятую в верхнем регистре ` оставляем **неизменными**.

Для сохранения жмем **голубой** треугольничек **>**, а потом **Publish**. Это в первый раз, при первом сохранении, потом будем нажимать **>** и **Republish**

Во второй строке оставляем только одну «шпалу» #, тогда Observable присвоит Вашей странице ИМЕННО ТАКОЕ ИМЯ.

Жмем на три точки и закрываем текстовый блок — Close.

Редактируем следующий текстовый блок

```
await logBook(`
  Date      | Time | Call | RST RX | RST TX | QTH | Name | Comment
2019-03-17 | 11:03 | LZ1ZB | 599 | 599 | KN12QO | Vlad|
```

и удаляем все мои QSO, кроме одного-двух — для образца.

Удобно просто копировать предыдущее QSO с последующим редактированием.

Вводим данные своих QSO, сохраняем. Это текущая страничка, **её лучше не закрывать и держать ОТКРЫТОЙ**. Ведь она будет периодически пополняться.

Примечание. В одном текстовом блоке максимум — 1000 строк, потом приходится открывать второй блок:

```
august=`
```

2. Лог для 144

Жмем <https://observablehq.com/@eu2aa/144logbook>

Ничего редактировать не нужно.

... остальное аналогично п.1

3. Лог для 432

Жмем <https://observablehq.com/@eu2aa/432logbook>

Ничего редактировать не нужно.

... остальное аналогично п.1

4. Лог для 1296

Жмем <https://observablehq.com/@eu2aa/1296logbook>

Ничего редактировать не нужно.

... остальное аналогично п.1

5. Конвертер в .ADIF для LOTW и QO-100 DX Клуба для спутника QO-100

Из файла Лога по п.1 создается универсальный файл ADIF для LOTW и QO-100 DX Club <https://qo100dx.club/>

Жмем <https://observablehq.com/@eu2aa/adif-for-lotw>

Жмем Fork, копируем себе, редактируем.

Сразу же идем в ПОСЛЕДНЮЮ строчку

```
toAdifQO100 = f(desc)
```

редактируем её **только в ОДНОМ** месте:

```
return `KO34KI<RST_RCVD:${rcvd.length}>${rcvd}<RST_SENT:${sent.length}>${sent}<GRIDSQUARE:${grid.length}>${grid}
```

Меняем локатор KO34KI на Ваш.

Close.

Дальше только ИСПОЛЬЗОВАНИЕ конвертера.

Их Лога по п.1 копируем голубую строчку

```
2019-03-17 | 11:03 | LZ1ZB | 599 | 599 | KN12QO | Vlad|
```

и вставляем её в блок **в строку ниже** после

```
await toAdifQO100(`
```

```
2019-03-17 | 11:03 | LZ1ZB | 599 | 599 | KN12QO | Vlad|
```

Вставлять можно и по несколько QSO одним блоком,

я пробовал по 1000 QSO,
но для этого нужна ловкость руки при владении мышью.
В предыдущем текстовом блоке появляется текст, но уже **в искомом формате .ADIF**
<QSO_DATE:8>20190317<TIME_ON:4>1103<CALL:5>LZ1ZB<MODE:2>CW<BAND:4>13CM
<FREQ:4>2400<BAND_RX:3>3CM
<FREQ_RX:5>10489<PROP_MODE:3>SAT<SAT_NAME:6>QO-
100<MY_GRIDSQUARE:6>KO34KI<RST_RCVD:3>599<RST_SENT:3>599<GRIDSQUARE:6>
KN12QO <EOR>

Примечание. Часть строки «заезжает» вправо за пределы экрана, но при копировании это не является помехой.

Этот текстовый блок вставляю в любой редактор, например просто по F4,
и сегодня, 5 декабря 2021 года, сохраняю с именем 211205.adif , но можно и с другим,
удобном для Вас.

6. Конвертер ADIF 144 LOTW 144MHz Tropo

Из файла Лога по п.1 создается файл ADIF для LOTW <https://qp100dx.club/>

Жмем <https://observablehq.com/@eu2aa/adif-for-lotw-144mhz-tropo>

Ничего редактировать не нужно.

... остальное аналогично п.5

7. Конвертер ADIF 144 LOTW 144MHz Aurora

Жмем <https://observablehq.com/@eu2aa/adif-for-lotw-144mhz-aurora>

Ничего редактировать не нужно.

... остальное аналогично п.5

8. Конвертер ADIF 144 LOTW 432MHz Tropo

Жмем <https://observablehq.com/@eu2aa/adif-for-lotw/2>

Ничего редактировать не нужно.

... остальное аналогично п.5

9. Конвертер в Cabrillo для теста Кубок Гагарина через QO-100

Из файла ADIF создается файл Cabrillo и текстовый файл для Лога по п.1

Жмем <https://observablehq.com/@eu2aa/adif2cabrillo-gc>

Жмем Fork, копируем себе.

Я использую в тестах CQRLOG <https://www.cqrlog.com/> , он мне нравится, работает под
Линуксом, отлично выдает отчет в .ADIF. Однако Cabrillo извлечь мне ни разу так и не
удалось. Требуется создавать Фильтры, но как, я так и не понял.

Работаем:

Жмем три точки в первой строчке

adif = ` и жмем Edit

В открывшееся ниже окно после

adif = `

вставляем текст файла .ADIF из CQRLOG, например:

ADIF export from CQRLOG for Linux version 2.5.2 (001)

Copyright (C) 2021 by Petr, OK2CQR and Martin, OK1RR

Internet: <http://www.cqrlog.com>

<ADIF_VER:5>3.1.0

<CREATED_TIMESTAMP:15>20210416 110826

<PROGRAMID:6>CQRLOG

<PROGRAMVERSION:11>2.5.2 (001)

<EOH>

<QSO_DATE:8>20210411<TIME_OFF:4>1943<STATION_CALLSIGN:5>EU2AA<CALL:5>E
W6FS<MODE:2>CW<FREQ:4>2400<BAND:4>13CM<RST_SENT:5>59929<RST_RCVD:5>5
9929<QSL_SENT:1>N<QSL_RCVD:1>N

<EOR>

<QSO_DATE:8>20210411<TIME_OFF:4>1920<STATION_CALLSIGN:5>EU2AA<CALL:5>DF
7CB<MODE:2>CW<FREQ:4>2400<BAND:4>13CM<RST_SENT:5>59929<RST_RCVD:5>599
28<QSL_SENT:1>N<QSL_RCVD:1>N

<EOR>

<QSO_DATE:8>20210411<TIME_OFF:4>1904<STATION_CALLSIGN:5>EU2AA<CALL:6>V
U2EEI<MODE:2>CW<FREQ:4>2400<BAND:4>13CM<RST_SENT:5>59929<RST_RCVD:5>5
9941<QSL_SENT:1>N<QSL_RCVD:1>N

<EOR>

Внимание! Смотрим, чтобы «не затоптать» запятую в верхнем регистре в конце
текстового блока!

«Шапка» игнорируется, её может и не быть.

Жмем > и Refrresh

В строчке

res3 = `

появляется текст в формате Cabrillo

QSO:	2.3G	CW	2021-04-11	1858	EU2AA	599	29	PR5KW	599	13
QSO:	2.3G	CW	2021-04-11	1904	EU2AA	599	29	VU2EEI	599	41
QSO:	2.3G	CW	2021-04-11	1904	EU2AA	599	29	4X1AJ	599	39
QSO:	2.3G	CW	2021-04-11	1920	EU2AA	599	29	DF7CB	599	28
QSO:	2.3G	CW	2021-04-11	1943	EU2AA	599	29	EW6FS	599	29

END-OF-LOG:

00000000011111111111222222222233333333333444444444455555555556666666666777777777788
123456789012345678901234567890123456789012345678901234567890123456789012345678901

В строчке

res4 = `

появляется текст в формате моего Лога

2021-04-11 | 18:58 | PR5KW | 599 | 599 | | |
2021-04-11 | 19:04 | VU2EEI | 599 | 599 | | |
2021-04-11 | 19:04 | 4X1AJ | 599 | 599 | | |
2021-04-11 | 19:20 | DF7CB | 599 | 599 | | |
2021-04-11 | 19:43 | EW6FS | 599 | 599 | | |

который вставляю в Лог по п.1

10. Конвертер из ADIF в Cabrillo для теста QO100 Challenge

Из файла ADIF создается файл Cabrillo и текстовый файл для Лога по п.1

Жмем <https://observablehq.com/@eu2aa/adif2cabrillo-qo-100-challenge>

Жмем Fork, копируем себе.

Я использую в тестах CQRLOG <https://www.cqrlog.com/>, он мне нравится, работает под
Линуксом, отлично выдает отчет в .ADIF. Однако Cabrillo извлечь мне ни разу так и не
удалось. Требуется создавать Фильтры, но как, я так и не понял.

Работаем:

Жмем три точки в первой строчке

adif = ` и жмем Edit

В открывшееся ниже окно после

adif = `

вставляем текст файла .ADIF из CQRLOG, например:

ADIF export from CQRLOG for Linux version 2.5.2 (001)

Copyright (C) 2021 by Petr, OK2CQR and Martin, OK1RR

Internet: <http://www.cqrlog.com>

<ADIF_VER:5>3.1.0
<CREATED_TIMESTAMP:15>20211121 161346
<PROGRAMID:6>CQRLOG
<PROGRAMVERSION:11>2.5.2 (001)
<EOH>
<QSO_DATE:8>20211121<TIME_OFF:4>0037<STATION_CALLSIGN:5>EU2AA<CALL:6>IK
5XLB<MODE:2>CW<FREQ:4>2400<BAND:4>13CM<RST_SENT:6>599047<RST_RCVD:6>5
99010<NAME:4>Luca<QSL_SENT:1>N<QSL_RCVD:1>N<GRIDSQUARE:4>JN53

<EOR>
<QSO_DATE:8>20211120<TIME_OFF:4>2350<STATION_CALLSIGN:5>EU2AA<CALL:6>DL
6NAV<MODE:2>CW<FREQ:4>2400<BAND:4>13CM<RST_SENT:6>599046<RST_RCVD:6>5
99019<NAME:7>Juergen<QSL_SENT:1>N<QSL_RCVD:1>N<GRIDSQUARE:4>JN49

<EOR>

«Шапка» игнорируется, её может и не быть.

Жмем > и Refrresh

В строчке

res3 = `

появляется текст в формате Cabrillo

```
QSO:  2.3G CW 2021-11-20 2350 EU2AA          599 046  DL6NAV          599 019  JN49
QSO:  2.3G CW 2021-11-21 0037 EU2AA          599 047  IK5XLB          599 010  JN53
END-OF-LOG:
0000000001111111111222222222233333333334444444444555555555566666666667777777777788
123456789012345678901234567890123456789012345678901234567890123456789012345678901
```

В строчке

res4 = `

появляется текст в формате моего Лога

2021-11-20 | 23:50 | DL6NAV | 599 | 599 | JN49 | Juergen | |

2021-11-21 | 00:37 | IK5XLB | 599 | 599 | JN53 | Luca | |

который вставляю в Лог по п.1

73! Будут вопросы или найдете ошибки, пишите eu2aa@tut.by

Владимир, EU2AA