BATTLE CONTRACT & BATTLE TOKEN AIRDROP

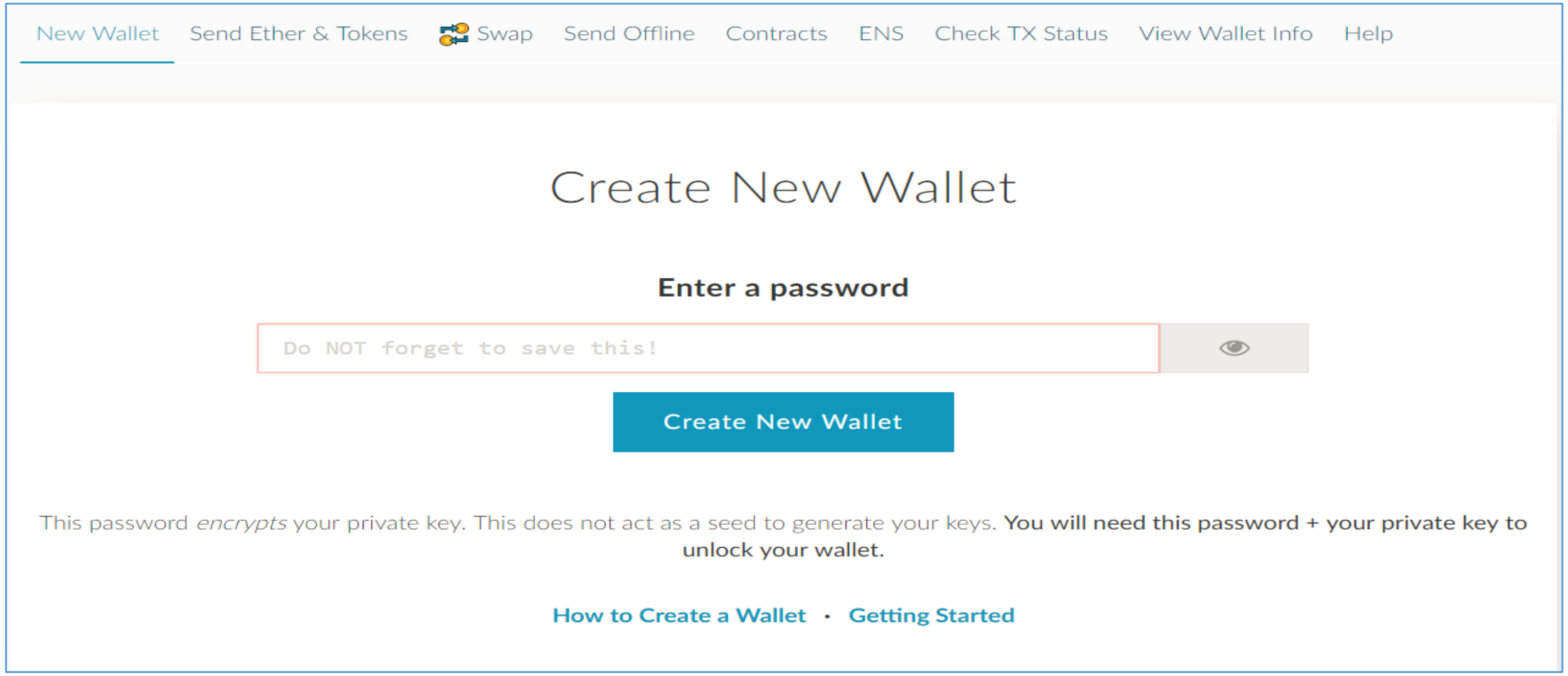
We are proud to announce the official release of the first **Ethereum Battle Smart Contract**

As a **gift** for our earlier investor we announce the **AIRDROP** of the **Battle Token** to all participant in our first Token battle

Battle Contract Guide

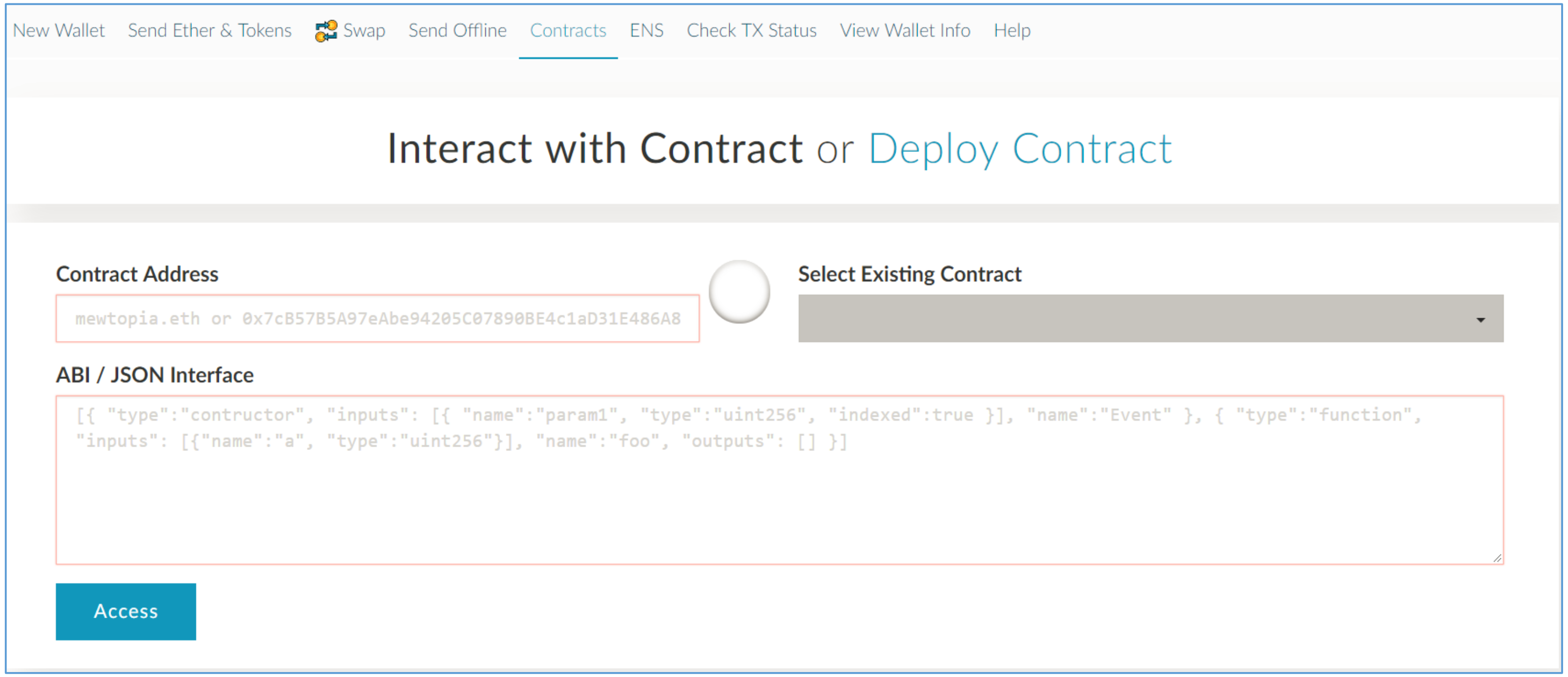
This guide will show how to deploy troop on the battle contract using MyEtherWallet.

Step 1 - Open MyEtherWallet
Double-check the URL (.com) before unlocking your wallett



This password *encrypts* your private key. This does not act as a seed to generate your keys. **You will need this password + your private key to unlock your wallet.**

Step 2 - Open Contracts tab



Interact with Contract or Deploy Contract

Contract Address

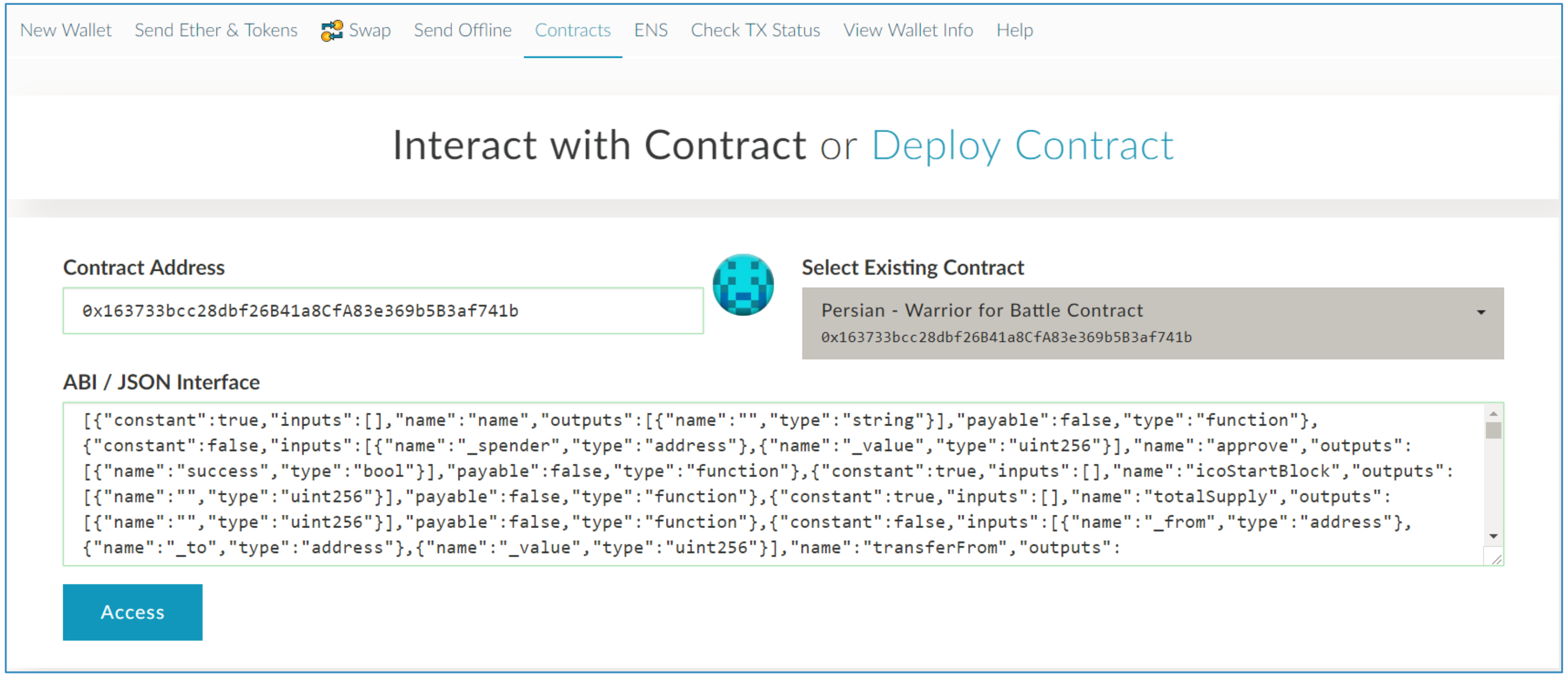
mewtopia.eth or 0x7cB57B5A97eAbe94205C07890BE4c1aD31E486A8

Select Existing Contract

ABI / JSON Interface

```
[{ "type":"contructor", "inputs": [{ "name":"param1", "type":"uint256", "indexed":true }], "name":"Event" }, { "type":"function",
"inputs": [{"name":"a", "type":"uint256"}], "name":"foo", "outputs": [] }]
```

Step 3 - Under heading Select Existing Contract is a dropdown, select **Persian - Warrior for Battle Contract** from the list and click the Access Button



Interact with Contract or Deploy Contract

Contract Address

0x163733bcc28dbf26B41a8CfA83e369b5B3af741b

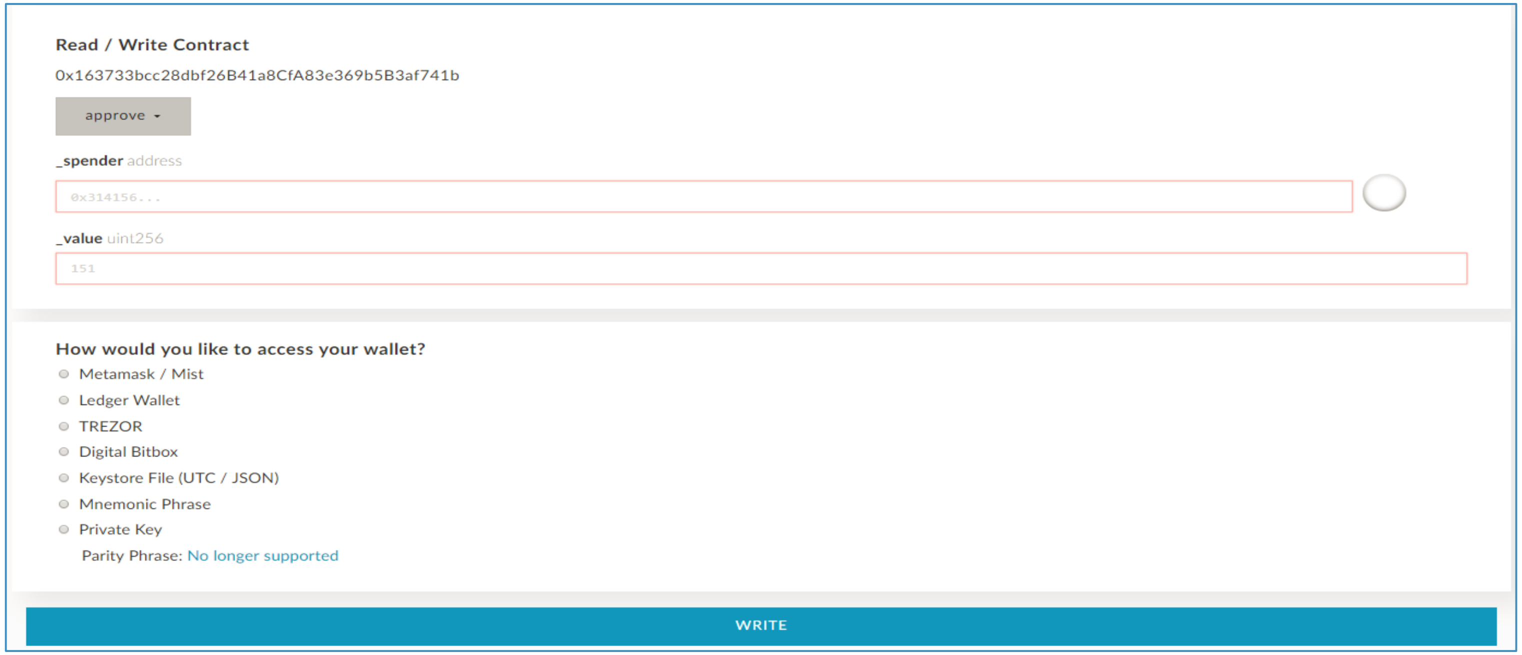
Select Existing Contract

Persian - Warrior for Battle Contract
0x163733bcc28dbf26B41a8CfA83e369b5B3af741b

ABI / JSON Interface

```
[{"constant":true,"inputs":[],"name":"name","outputs":[{"name":"","type":"string"}],"payable":false,"type":"function"},
{"constant":false,"inputs":[{"name":"_spender","type":"address"},{"name":"_value","type":"uint256"}],"name":"approve","outputs":
[{"name":"success","type":"bool"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"icoStartBlock","outputs":
[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"totalSupply","outputs":
[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_from","type":"address"},
{"name":"_to","type":"address"},{"name":"_value","type":"uint256"}],"name":"transferFrom","outputs":
```

Step 4 - Under **Read / Write Contract** heading, open the dropdown **Select function** and click on **approve**



Read / Write Contract

0x163733bcc28dbf26B41a8CfA83e369b5B3af741b

approve ▾

_spender address

0x314156...

_value uint256

151

How would you like to access your wallet?

○ Metamask / Mist
○ Ledger Wallet
○ TREZOR
○ Digital Bitbox
○ Keystore File (UTC / JSON)
○ Mnemonic Phrase
○ Private Key

Parity Phrase: No longer supported

WRITE

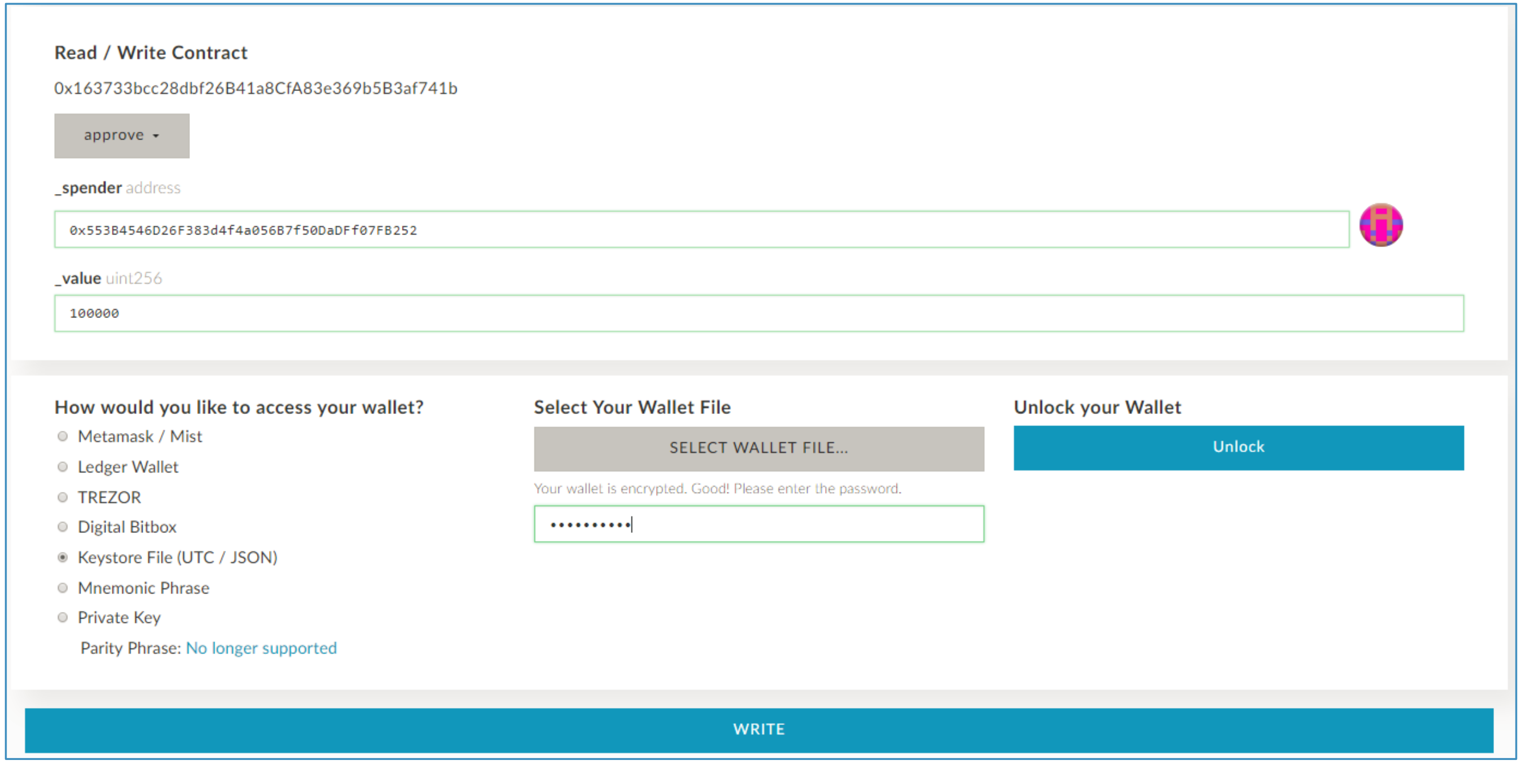
Step 5

In the new section

- In field **_spender** insert the Battle Contract Address **0x553B4546D26F383d4f4a056B7f50DaDFf07FB252**
- In field **_amount** insert the amount of warrior token that you want to deploy on the battlefield

Load your wallet using any one of the methods listed, for this example we used the Keystore File:

- Select the wallet file from your file system with the grey **SELECT WALLET FILE**… button
- Enter your password and click the **unlock button**.



Read / Write Contract

0x163733bcc28dbf26B41a8CfA83e369b5B3af741b

approve ▾

_spender address

0x553B4546D26F383d4f4a056B7f50DaDFf07FB252

_value uint256

100000

How would you like to access your wallet?

○ Metamask / Mist
○ Ledger Wallet
○ TREZOR
○ Digital Bitbox
● Keystore File (UTC / JSON)
○ Mnemonic Phrase
○ Private Key

Parity Phrase: No longer supported

Select Your Wallet File

SELECT WALLET FILE...

Your wallet is encrypted. Good! Please enter the password.

••••••••••

Unlock your Wallet

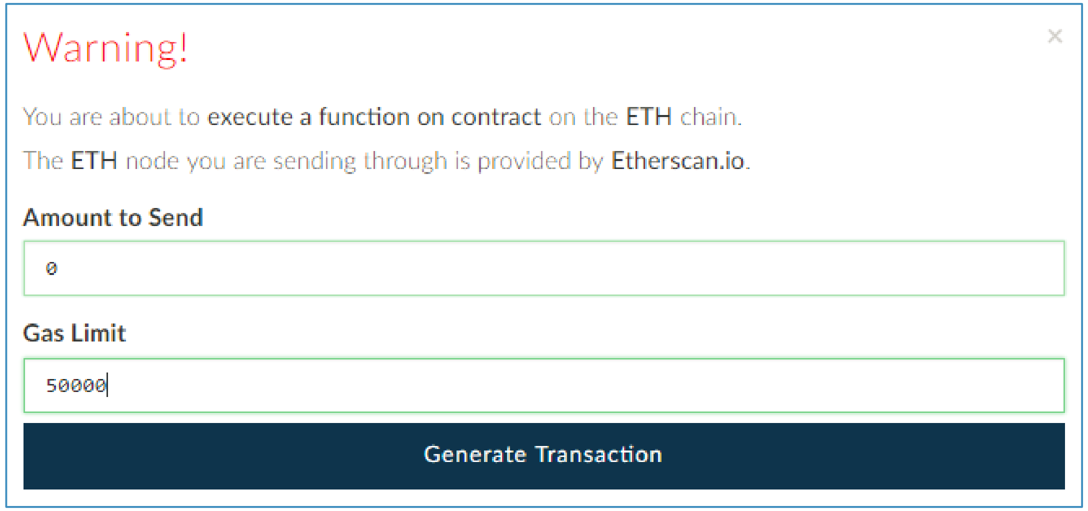
Unlock

WRITE

Step 6

Click on the write button, a popup will appear:

- leave the value of **Amount to Send to 0 ETH**
- allow the wallet to suggest a Gas Limit. If it does not autopopulate, enter 50000 or more. If your TX fails, increase Gas Limit.



Warning!

You are about to execute a function on contract on the ETH chain.

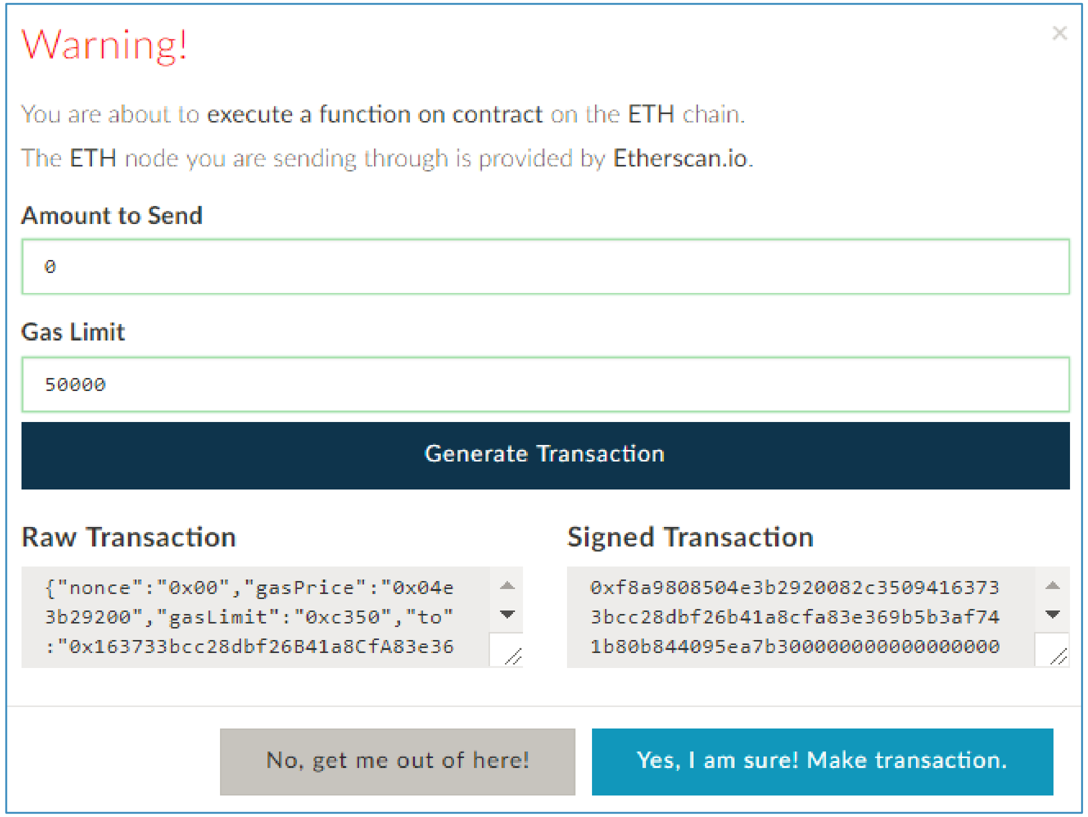
The ETH node you are sending through is provided by Etherscan.io.

Amount to Send: 0

Gas Limit: 50000

Step 7

Click on the **Generate Transaction** button, wait a while, when ready click on the blue button **Yes, I am sure! Make transaction.**



Warning!

You are about to execute a function on contract on the ETH chain.

The ETH node you are sending through is provided by Etherscan.io.

Amount to Send: 0

Gas Limit: 50000

Raw Transaction

```
{"nonce":"0x00","gasPrice":"0x04e
3b29200","gasLimit":"0xc350","to"
:"0x163733bcc28dbf26B41a8CfA83e36
```

Signed Transaction

```
0xf8a9808504e3b2920082c3509416373
3bcc28dbf26b41a8cfa83e369b5b3af74
1b80b844095ea7b300000000000000000
```

Step 8

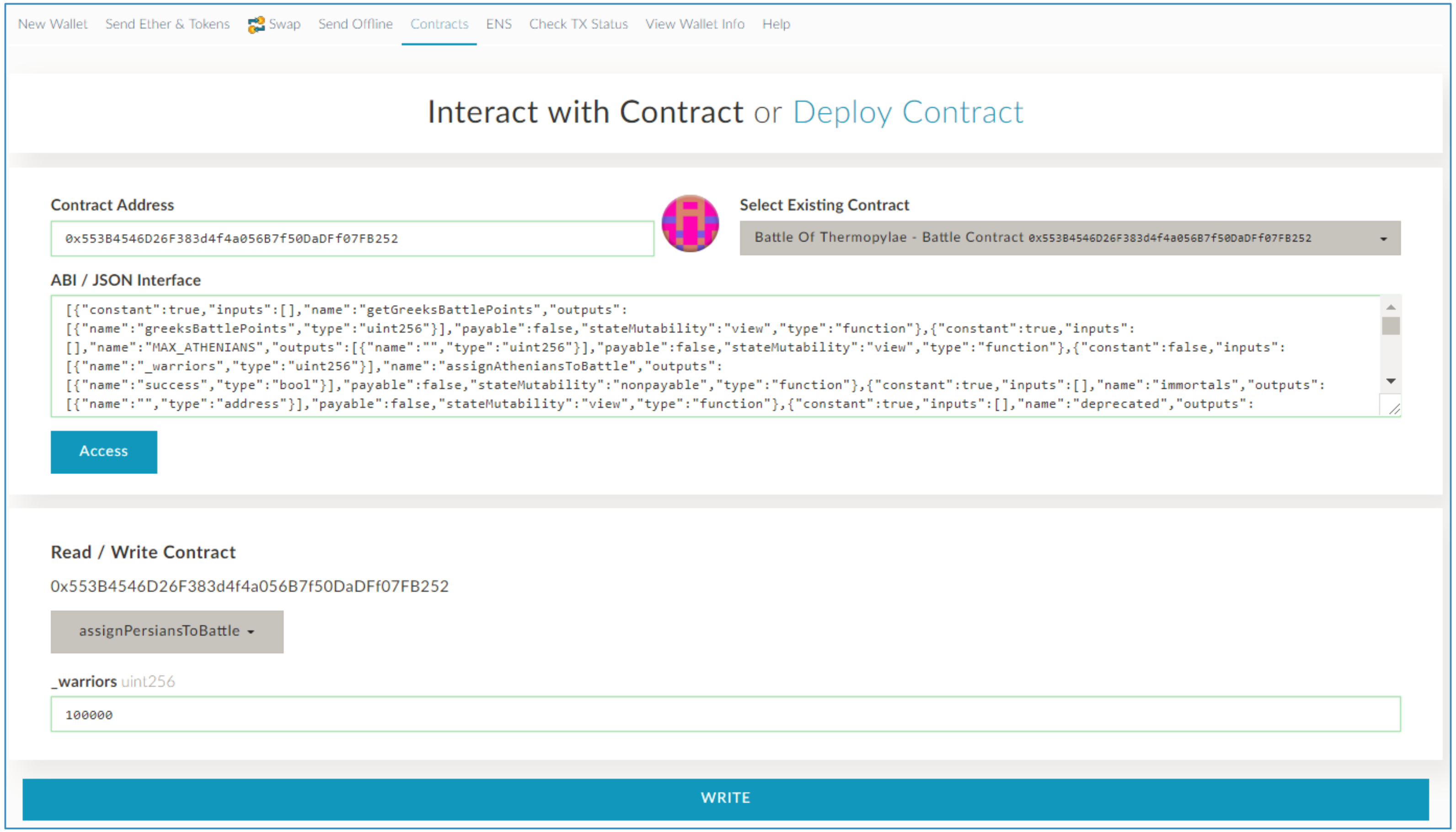
Reload the MyEtherWallet Contracts tab, under heading Select Existing Contract is a dropdown, select **Battle of Thermopylae - Battle Contract** from the list and click the Access Button

Under **Read / Write Contract** heading, open the dropdown **Select function** and click on **assignPersiansToBattle**, a new section will appear:

- In field **_warriors** insert the amount of warrior token that you want effectively deploy on the battlefield, this amount must be minor or equal to the amount set in the approve function

Load your wallet using any one of the methods listed, for this example we used the Keystore File:

- Select the wallet file from your file system with the grey **SELECT WALLET FILE…** button
- Enter your password and click the **unlock button**.



Interact with Contract or Deploy Contract

Contract Address

0x553B4546D26F383d4f4a056B7f50DaDFf07FB252

Select Existing Contract

Battle Of Thermopylae - Battle Contract 0x553B4546D26F383d4f4a056B7f50DaDFf07FB252

ABI / JSON Interface

```
[{"constant":true,"inputs":[],"name":"getGreeksBattlePoints","outputs":
[{"name":"greeksBattlePoints","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":true,"inputs":
[],"name":"MAX_ATHENIANS","outputs":[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs":
[{"name":"_warriors","type":"uint256"}],"name":"assignAtheniansToBattle","outputs":
[{"name":"success","type":"bool"}],"payable":false,"stateMutability":"nonpayable","type":"function"},{"constant":true,"inputs":[],"name":"immortals","outputs":
[{"name":"","type":"address"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":true,"inputs":[],"name":"deprecated","outputs":
```

Read / Write Contract

0x553B4546D26F383d4f4a056B7f50DaDFf07FB252

assignPersiansToBattle

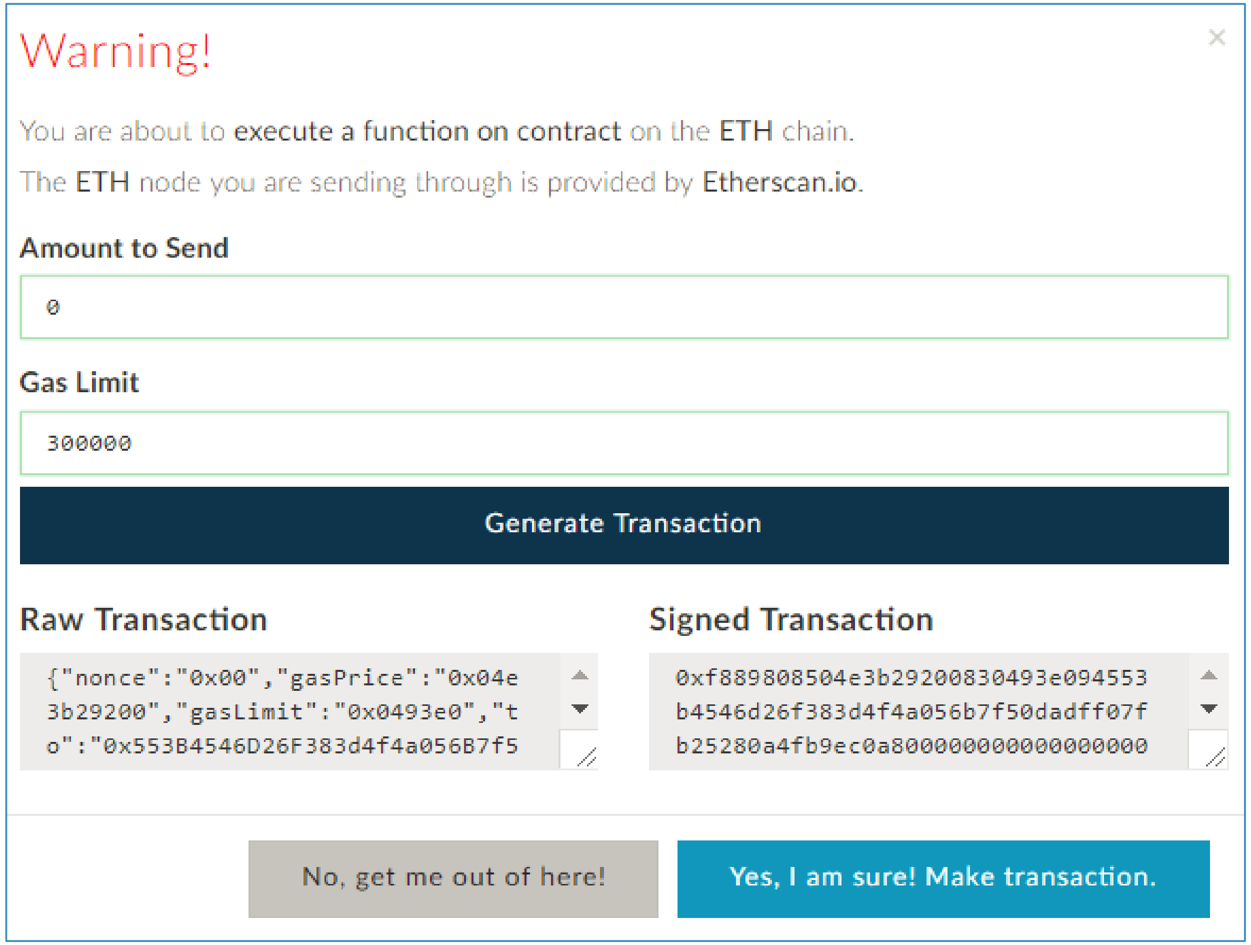
_warriors uint256

100000

Step 9

Click on the write button, a popup will appear:

- leave the value of **Amount to Send to 0 ETH**
- allow the wallet to suggest a Gas Limit. If it does not autopopulate, enter 300000 or more. If your TX fails, increase Gas Limit.
- Click on the Generate Transaction button, wait a while, when ready click on the blue button **Yes, I am sure! Make transaction**.



Warning!

You are about to execute a function on contract on the ETH chain.

The ETH node you are sending through is provided by Etherscan.io.

Amount to Send

0

Gas Limit

300000

Raw Transaction

```
{"nonce":"0x00","gasPrice":"0x04e
3b29200","gasLimit":"0x0493e0","t
o":"0x553B4546D26F383d4f4a056B7f5
```

Signed Transaction

```
0xf889808504e3b29200830493e094553
b4546d26f383d4f4a056b7f50dadff07f
b25280a4fb9ec0a800000000000000000
```