**Доступ к блокчейн среде из внешней программы**

Блокчейн технологии развиваются стремительными темпами. Сегодня уже ни кого не удивишь Bitcoin ом или другой криптовалютой. Существует много разнообразных кошельков и систем.

Не редко перед разработчиками программного обеспечения встает задача связать свои программные продукты с системой блокчейн. В данной статье рассмотрим пример получения информации из системы Ерачейн.

Установленное программное обеспечение системы Ерачейн будем называть нодой.

Программу, которой необходимо получить доступ к системе будем называть внешней программой. Для примера внешней программой будет выступать интернет браузер.

Обмен информацией между нодой и внешней программой осуществляется по протоколу HTTP посредством запросов GET или POST. Ответ получаем в формате JSON.

Такой подход позволяет обеспечить доступ к системе блокчейн без участия кошелька.

Система Ерачейн

API

Внешняя программа

HTTP GET/POST запрос

Внутренний протокол системы Ерачейн

Кошелек

Нода

**Нам понадобятся:**

1. Java. Берем здесь <https://www.java.com/ru> . Устанавливаем согласно инструкции.

2. Программное обеспечение ноды. https://github.com/rty1966/ERA\_Wallet\_DEV\_V\_4\_3\_02.git Распаковываем и сохраняем в любой директории на компьютере. В нашем случае сохраним в c:\block

3. Браузер. Firefox, Chrom и т.д.

**Первый запуск**

Запускаем файл run.bat из директории c:\ block.

Выбираем язык системы и устанавливаем шрифт. Описание процесса приведено для Русского языка.

Если вы впервые работаете с Ерачейн «Создать новый кошелек»

Соглашаемся (или не соглашаемся) с лицензионным соглашением. Советую прочитать его внимательно т.к. там написано много интересного.

В открывшемся окне копируем Seed (Ctrl+C).

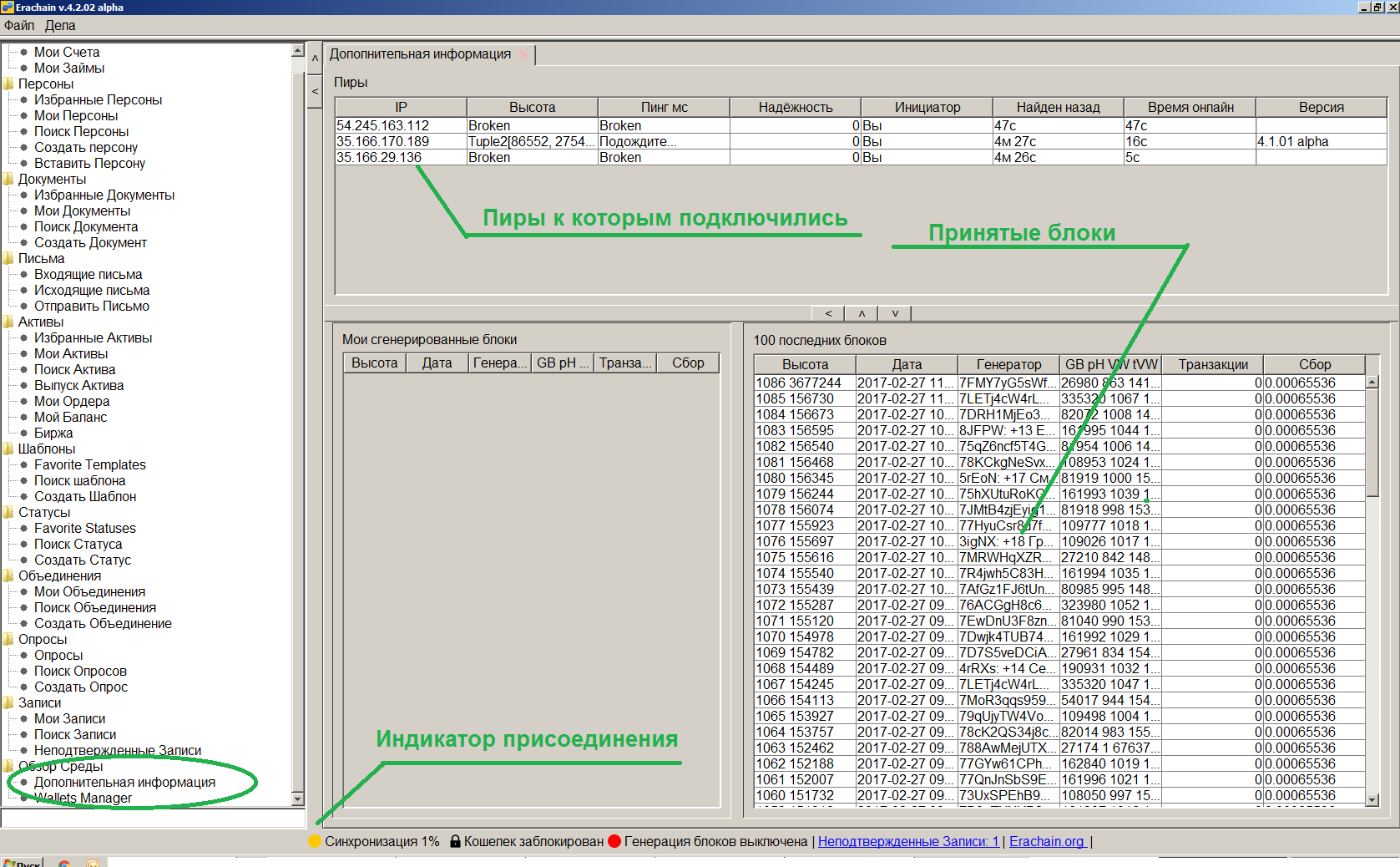
**Очень важно**: Сохраните Seed в секретном месте. Кто обладает Seed, тот имеет полный доступ к кошельку. Даже если вы уничтожите кошелек или компьютер, при вводе Seed на другом компьютере вся информация о вашем кошельке восстановится. В случае утраты Seed информация кошелька будет утеряна навсегда.

В следующем окне вставляем Seed (Ctrl+V) и указываем пароль для кошелька.

Немного ждем, пока нода создаст кошелек и выведет главное окно.

В главном Окне в левой панели выбираем пункт «Дополнительная информация».

На открывшейся вкладке можно наблюдать за состоянием ноды.

****

При подключении к системе индикатор присоединения изменится с красного цвета на желтый и начнется синхронизация (чтение блоков).

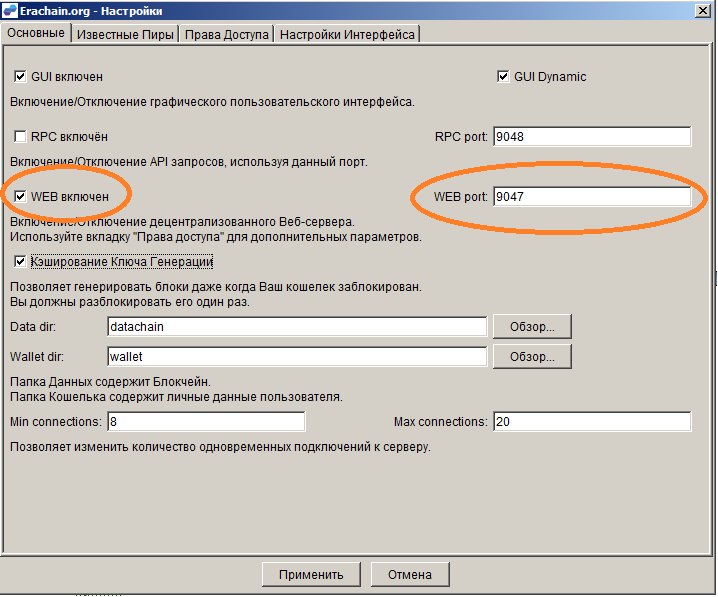
**Замечание: Для подключения к системе потребуется некоторое время(иногда до 10 минут).**

Синхронизация происходит в параллельном потоке, поэтому вы можете выполнять любые действия.

**Настройка ноды**

Для настройки ноды в качестве моста между Ерачейн и внешней программой необходимо настроить доступ WEB.

Для этого заходим в меню «Файл» «Настройки», и устанавливаем галочку «WEB включен»



Нажимаем кнопку «Применить».

После перезапуска Нода готова к работе с внешней программой.

**Выполнение GET запроса**

Открываем браузер, в адресной строке пишем : 127.0.0.1:9047/api

На экран выведутся все запросы, которые вы можете отправлять ноде.

Дополнительно вы можете вызвать: 127.0.0.1:9047/apirecords

Давайте запросим у ноды последний блок. Для этого выполним GET запрос: 127.0.0.1:9047/api/lastblock

Получим ответ JSON

"reference":"3ttojQuUTHsD85MpUf...eiDguu2ixgSTQpff2Shk4SErVXcjoMUNTX8bxVywcgJ",

"blockATs":"",

"creator":"75hXUtuRoKGCyhzps7LenhWnNtj9BeAF12",

"signature":"5HePJJyoo7nw6Xm64...ir6b3sybm4ypurg1DKMTJVufd3LQAow7Jo2AnD7neyQ5t",

"fee":"0.00008192",

"transactions":[],

"version":1,

"generatingBalance":124885,

"winValueTargeted":2870,

"transactionsHash":"HuPQsudxJC41ZFS7dRsyYp18RjNZMRJaVjnGv8HHo3Hr",

"timestamp":1512707833333,

"height":86330

Описание JSON:

reference: подпись предыдущего блока,

blockATs: зарезервировано,

creator: адрес создателя блока,

signature: подпись блока,

fee: сбор за блок,

transactions: список транзакций,

version: версия блока,

generatingBalance: баланс на счету создателя блока,

winValueTargeted: величина победы,

transactionsHash: ХЭШ транзакций,

timestamp: время в формате UNIX,

height: высота(номер) блока

Получить информацию о 20-м блоке можно по запросу:

127.0.0.1:9047/api/blockbyheight/20

Все остальные команды выполняются аналогичным образом.

Теперь вы получили в свое распоряжение инструмент позволяющий делать много чего интересного, например, используя API, был создан мобильный кошелек для системы ос Android https://github.com/Codeevodevelopment/ERAWallet

Список GET запросов

|  |  |  |
| --- | --- | --- |
| height | высота последнего блока | 127.0.0.1:9047/api/height |
| firstblock | Genesis блок | 127.0.0.1:9047/api/ |
| lastblock | последний блок | 127.0.0.1:9047/api/ |
| block/{signature} | получить блок по его сигнатуре |  |
| blockbyheight/{height} | получить блок по его высоте (номеру) |  |
| childblocksignature/{signature} | Получить блок следующий за блоком |  |
| blocksfromheight/{height}/{limit} | Получить limit блоков начиная с height блока |  |
| blockssignaturesfromheight/{height}/{limit} | Получить сигнатуры limit блоков начиная с height блока | http://127.0.0.1:9047/api/blockssignaturesfromheight/5/5 |
| recordparse/{raw} | Создать транзакцию из {raw} массива байт |  |
| record/{signature} | Получить транзакцию по ее сигнатуре |  |
| recordbynumber/{height-sequence} | Получить транзакцию по номеру блока { height} и ее номере в блоке { sequence} | http://127.0.0.1:9047/api/recordbynumber/27115-1 |
| addressvalidate/{address} | Является ли строка {address} адресом | http://127.0.0.1:9047/api/addressvalidate/7R4jwh5C83HLj7C1FiSbsGptMHqfAirr8R |
| addressassetbalance/{address}/{assetid} | Баланс адреса {address} для номера актива {assetid} | http://127.0.0.1:9047/api/addressassetbalance/7R4jwh5C83HLj7C1FiSbsGptMHqfAirr8R/1 |
| addressassets/{address} | Балансы по всем активам адреса | http://127.0.0.1:9047/api/addressassets/7R4jwh5C83HLj7C1FiSbsGptMHqfAirr8R |
| addresspublickey/{address} | Открытый ключ адреса |  |
| assetheight | Максимальный номер актива | http://127.0.0.1:9047/api/assetheight |
| asset/{key} | Информация о активе по его номеру | http://127.0.0.1:9047/api/asset/1 |
| assetdata/{key} | Получить изображения и иконку актива в формате BASE64 |  |
| assetimage/{key} | Получить изображение актива в бинарном виде |  |
| asseticon/{key} | Получить иконку актива в бинарном виде |  |
| assets | Получить список всех активов системы |  |
| assetsfilter/{filter\_name\_string} | Получить список активов по части его названия |  |
| personheight | Количество персон в системе |  |
| person/{key} | Информация о персоне по ее номеру {key} | http://127.0.0.1:9047/api/person/2 |
| persondata/{key} | Получить изображения и иконку персоны в формате BASE64 |  |
| personimage/{key} | Получить картинку персоны в бинарном виде |  |
| personkeybyaddress/{address} | Получить номер персоны по адресу |  |
| personbyaddress/{address} | Получить информацию о персоне по ее адресу |  |
| personsfilter/{filter\_name\_string} | Получить список персон для имя которых включает {filter\_name\_string} |  |

В следующей статье мы научимся создавать адреса кошелька, что приблизит нас к самому интересному – сбору транзакции и отправке её в блокчейн.