



Blockchain School

Урок 1. Введение

Разбираемся, что такое блокчейн сеть Ethereum и как в ней устроены транзакции, создаем свои кошельки и учимся с их помощью взаимодействовать с умными контрактами в сети.

План

1. Создание кошелька, прием и отправка эфира
2. Что такое приватный ключ?
3. Что такое публичный ключ и адрес?
4. Адрес с чек-суммой
5. Что такое транзакция?
6. Комиссия за транзакцию. Что такое Gas и GasPrice?
7. Что такое попсе?
8. Что такое блок?
9. Задания

Важно:

Предлагаю ознакомиться с видео, в котором мой коллега рассказывает о блокчейне: <https://www.youtube.com/watch?v=5tY56DkVPsI>

Также, почитайте любые материалы по Ethereum, которые вам понравятся. Например вот: <https://ethereum.org/ether>
Технические детали можно пропустить.

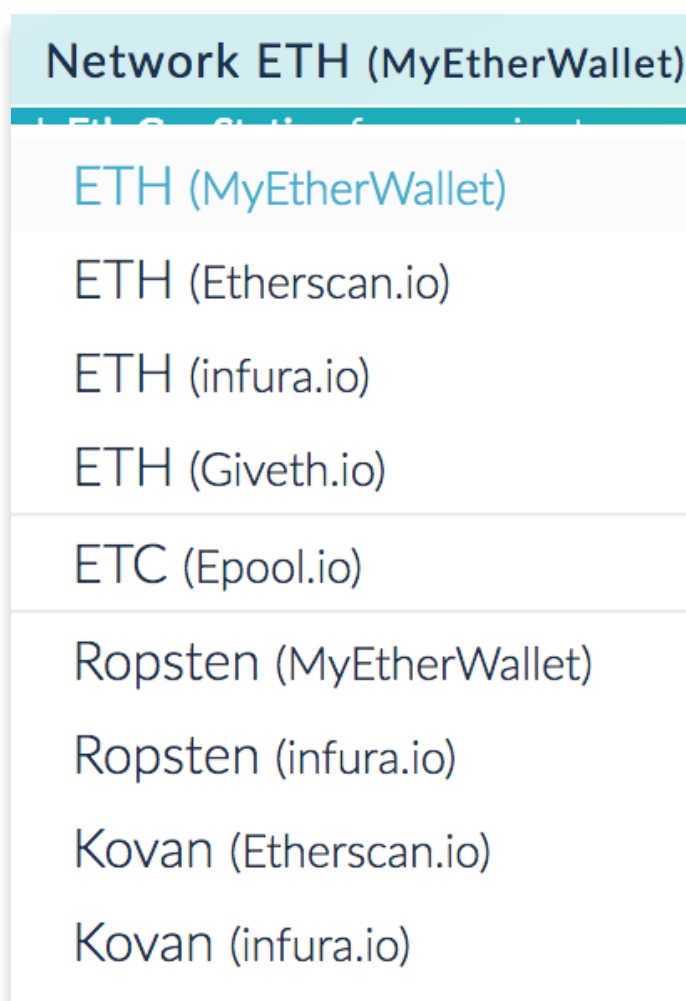
1. Создание кошелька, прием и отправка эфира

С помощью [MyEtherWallet](#) мы создаем Ethereum кошелек для хранения эфирной валюты ETH.

[MyEtherWallet](#) – это дефолтный эфирный кошелек, который работает в браузере.

Во время курса мы будем работать не в главной сети Ethereum, а в тестовой сети Kovan. Транзакции в этой сети оплачиваются не обычным эфиром, а Kovan Ether – тестовой валютой. Так мы не будем тратить реальные деньги на деплой умных контрактов и сможем учиться столько, сколько нужно.

Чтобы поменять сеть, на сайте [MyEtherWallet](#), в правом верхнем углу, выберите Network Kovan (Etherscan.io).



1. Создание кошелька, прием и отправка эфира


Создать кошелек очень просто:

1. Вводите пароль;
2. Сохраняете контейнер с информацией о вашем кошельке;
3. Сохраняете свой приватный ключ.

Create New Wallet

Enter a password


Do NOT forget to save this!



Create New Wallet

Done. Теперь вы можете увидеть адрес своего кошелька, его баланс и другие подробности.

Account Address



0xC83a62b764133773bc56149f4

C6BD23a8b59551b

Account Balance

0 ETH

Transaction History

[ETH \(https://etherscan.io\)](https://etherscan.io)
[Tokens \(Ethplorer.io\)](https://ethplorer.io)

2. Что такое приватный ключ?

Приватный ключ дает вам доступ к кошельку – конкретному денежному счету в сети Ethereum. В случае с эфиром, приватный ключ – это 32 байта информации.

[MyEtherWallet](#) позволяет безопасно генерировать приватные ключи. Ключ создается при помощи генератора случайных чисел, сразу шифруется и помещается в специальный контейнер – текстовый документ, в котором прописаны данные о кошельке. Достать ключ из контейнера может только тот, кто знает пароль, который вы вводите при создании кошелька. Если у вас надежный пароль, никто не сможет достать ваш ключ из контейнера, даже если он был скомпрометирован.

3. Что такое публичный ключ и адрес?

Каждому приватному ключу соответствует публичный ключ. Он генерируется путем хеширования приватного ключа. Публичный ключ – это и есть адрес кошелька, идентификатор аккаунта в блокчейне Ethereum. Вы сообщаете публичный ключ другим участникам сети, чтобы они могли с вами взаимодействовать, например, отправлять вам деньги или включать вас в список доверенных адресов для контракта.

С помощью этого публичного ключа вы можете с любого устройства получить доступ к соответствующему адресу кошелька.

Подробнее о публичном ключе: https://en.wikipedia.org/wiki/Public-key_cryptography

Для шифрования используется криптография эллиптических кривых. Здесь вы можете почитать об этом подробнее: <https://habrahabr.ru/post/188958>

4. Адрес с чек-суммой

Чтобы добавить к адресу чек-сумму, нужно указать некоторые буквенные символы адреса в верхнем регистре (upper case). По этим символам можно понять, нет ли в адресе опечатки.

Подробнее: <https://github.com/ethereum/EIPs/blob/master/EIPS/eip-55.md>

Или на вики: <https://en.wikipedia.org/wiki/Checksum>

5. Что такое транзакция?

Транзакция – это сообщение, отправленное одним из аккаунтов (кошельков) и подписанное приватным ключом, который соответствует этому аккаунту (кошельку). Если вы подписали какое-то сообщение, вашу подпись никто не сможет подделать и все получатели сообщения могут быть уверены, что его подписали именно вы.

Создание кошельков, обмен валютой между ними – это самые простые транзакции, которые можно выполнить в сети. Блокчейн Ethereum позволяет также проводить более сложные транзакции, связанные с умными контрактами – вы можете деплоить контракты, взаимодействовать с ними. Например, отправлять в контракты деньги, просматривать публичные данные контракта, менять значения данных в этом контракте, пользоваться его функциями в своих контрактах.

Каждая транзакция имеет множество характеристик, детали которых можно посмотреть на etherscan.io:

TxHash: Транзакция, прохешированная с помощью алгоритма кэска256.

TxReceipt Status: Статус транзакции – подтверждена она сетью или нет.

Block Height: Количество блоков, которые подтвердили транзакцию.

TimeStamp: Время, когда была смайнена транзакция = Block TimeStamp, время когда блок, включающий транзакцию, был найден.

From: Адрес отправителя.

To: Адрес получателя (в данном случае – контракт).

Value: Количество денег (ETH), отправленных вместе с транзакцией. Это значение может быть нулем.

Gas Limit: Количество газа¹, которое был готов потратить отправитель.

Gas Used By Txn: Количество газа, которое было потрачено за транзакцию. В данном случае это число меньше, чем Gas Limit. Это значит, что отправитель на всякий случай “налил” газ с запасом. Остатки ему вернулись.

Gas Price: Цена в ETH, которую отправитель готов заплатить за единицу газа. Если газ лимит у нас 250000, а цена одной единицы газа 4 Gwei, то за транзакцию отправитель готов был заплатить 1 млн Gwei или же 0,001 ETH².

Actual Tx Cost/Fee: Фактическая цена транзакции с учетом количества использованного газа и установленной на него цены

Actual Tx Cost = Gas Used By Txn * Gas Price

Cumulative Gas Used: Количество газа, которое было потрачено на выполнение всех предыдущих транзакций в этом же блоке.

Nonce: Порядковый номер транзакции³.

¹ - Подробнее в пункте №5

² - 1 Gwei = 0.000000001 ETH

³ - Подробнее в пункте №6

100

Input Data: Когда мы отправляем деньги с одного кошелька на другой, это поле остается пустым. Когда мы взаимодействуем с контрактом и передаем в него какие-то значения (деньги, имя, числа и т.д.), тогда их шестнадцатеричные эквиваленты мы можем видеть в этом поле.

[illegible]

Convert To Ascii

Etherscan не показывает, что есть еще поле **chainID** – идентификатор блокчейн сети. Например, 42 – это ID сети Kovan. Если мы попробуем взять транзакцию с этим ID и выполнить ее в другой сети, то ничего не выйдет, потому что chainID для сети Ethereum – это единица.

ChainID можно увидеть, когда вы подписываете транзакцию.

6. Комиссия за транзакцию.

Что такое Gas и что такое GasPrice?

Основная цель сети Ethereum – обработка транзакций. Выполнение каждой из них требует определенного количества ресурсов, таких как процессорное время и место для хранения данных, поэтому необходимо платить комиссию и с ее помощью поощрять исполнителей.

Чтобы понять, что такое Газ (Gas), рассмотрим простой пример.

Представь, что есть рынок, который объединяет службы доставки денег (эфира). На этом рынке ты можешь заказывать пересылку денег (эфира), но тебе надо за это платить такими же деньгами. Ты публично озвучиваешь задание: доставить 1000 единиц денег (эфира) по такому-то адресу. Расстояние между тобой и указанным адресом – 10 км. Ты говоришь, что готов заплатить 2 единицы денег за каждый километр. Получается, в этом случае $Gas = 10$ (нужно проехать 10 км), $GasPrice = 2$ (за каждый километр ты платишь 2 единицы денег). Итого, комиссия за пересылку составит 20 единиц денег. Чтобы заказать выполнение этого задания, тебе нужно иметь на своем счету 1020 денег (1000 единиц отправить и 20 единиц заплатить за доставку). Дальше исполнители (службы) сами решают, хотят ли они браться за такое задание за такую оплату.

Допустим какой-то исполнитель согласился, но во время выполнения оказалось, что расстояние всего 9 км. Оплату за 10-й километр, который не нужно было проезжать, он вернет заказчику.

Или наоборот, если расстояние оказалось 15 км, а не 10 км, то исполнитель забирает себе оплату за доставку (20 единиц денег), а заказчику возвращает ту сумму, которую он не смог доставить (1000 единиц).

За выполнение транзакции платит ее отправитель. Когда он создает транзакцию, то определяет, сколько необходимо ресурсов для ее выполнения (Gas) и сколько эфира он готов заплатить за единицу газа (gasPrice). Например, отправитель знает, что для выполнения его транзакции понадобится 30000 газа, и он готов заплатить за каждую единицу газа 0.00000002 ETH (gasPrice).

6. Комиссия за транзакцию. Что такое Gas и что такое GasPrice?

Цена транзакции

$\text{gas} * \text{gasPrice} = 30000 * 0.00000002 \text{ ETH} = 0.0006 \text{ ETH}.$

Если в результате выполнения транзакции было использовано только 21000 газа, то отправитель получит обратно $9000 * 0.00000002 \text{ ETH}$.

Таким образом отправитель может 'заказывать' выполнение транзакции за определенную цену. Потенциальные исполнители транзакции (майнеры) видят ее цену и сами решают, соглашаться или нет на ее выполнение. Если они готовы продавать свои ресурсы, вычислительные мощности их компьютеров, за такую цену, то после выполнения транзакции получают указанную оплату.

Важно поставить правильную цену на газ, чтобы транзакция быстро выполнялась.

На сайте Etherscan доступна информация о состоянии сети Ethereum. Вы можете просмотреть транзакции, которые находятся в очереди, или выполненные транзакции.

etherscan.io => Blockchain => View Pending Transactions

Вы можете отсортировать Pending Transactions по цене на газ и просто поставить в своей транзакции цену чуть выше, тогда она будет первой в очереди.

Если вы хотите, чтобы вашу транзакцию быстрее выполнили, лучше увеличивать цену за газ, а не количество газа. У каждого блока в сети Ethereum ограниченный размер. Все транзакции, которые входят в блок, не должны превышать суммарный лимит по газу. Лимит газа в блоке постоянно разный. Сейчас, он составляет примерно 8 миллионов газа, но со временем это количество по чуть-чуть растет.

Если в вашей транзакции будет указано очень большое количество газа, то ее не сразу возьмут в блок.

7. Что такое nonce?

О Nonce уже шла речь, когда мы рассматривали детали транзакций. Теперь разберемся, зачем он вообще нужен.

Все транзакции выполняются в строгом порядке. Вашу транзакцию с порядковым номером 2 никто не выполнит (не смайнит) раньше, чем транзакцию с номером 1.

Почему это важно?

Допустим, в первой транзакции вы регистрируетесь, а во второй – отправляете деньги. Эти транзакции могут идти только в таком порядке, иначе в них не будет логики.

Чтобы нельзя было одну и ту же транзакцию использовать несколько раз. Например, вы дали разрешение одному адресу снять у вас 100 токенов. Это должна быть разовая транзакция, ведь вы не хотите, чтобы у вас несколько раз снимали 100 токенов.

Каждый раз, когда мы меняем какие-либо данные в транзакции, ее подпись (хеш) меняется. Если мы второй раз отправляем ту же сумму денег на тот же адрес, нужно, чтобы электронная подпись была другая. В этом помогает nonce – порядковый номер транзакции для конкретного отправителя.

Сеть Ethereum не принимает повторно транзакции с одной и той же подписью. Nonce меняется каждый раз, соответственно подпись меняется каждый раз. Таким образом мы контролируем все, что происходит с нашим кошельком.

Если отправить транзакцию с nonce 5, когда еще нет транзакций с nonce 3 и 4, то она будет висеть в очереди и не будет выполняться до тех пор, пока не будут созданы и не выполнены транзакции с nonce 3 и 4.

Если мы отправим две разные транзакции с одинаковым nonce, но с разными данными, то мы не можем быть уверены, какая из них выполнится (смайнится) первой. Но после того как смайнилась одна из них, вторая уже не пройдет, потому что у нее тот же nonce, который уже вписан в блокчейн. Такие транзакции удаляются автоматически.

8. Что такое блок?

Блок – это единица изменения состояния блокчейна. Состояние сети изменяется блоками транзакций.

Представим, что у вас есть новая сеть, в которой пока что нет ни одного кошелька и ни одной транзакции. Допустим, в вашей сети в блок помещается 8 миллионов газа. Это значит, что вы можете включить в блок такое количество транзакций, чтобы их суммарная стоимость в газе не превышала 8 миллионов.

Допустим в вашей сети есть 2 адреса с деньгами на счету: 'А' и 'Б'. В блок попадает транзакция перевода всех денег с адреса 'А' на 'Б'. Теперь состояние вашей сети – это два адреса: 'А' без денег, и 'Б' с деньгами.

9. Задание №1

Мы начинаем учиться общаться с умным контрактом на эфирном тестнете Kovan. Для этого вам необходимо:

1. Создать себе Ethereum кошелек с помощью [MyEtherWallet](#)
2. Сообщить мне адрес вашего кошелька в Телеграм. Мой username – @lastperson
3. Я отправлю на адрес вашего кошелька немного эфира из тестовой сети Kovan (KEther, kether, KETH)
4. Вам надо будет вернуть 0.005 эфира на тот же адрес, с которого он придет.

Я написал и задеплоил в тестовый блокчейн Kovan маленький контракт (там видно адрес, исходный код, и ABI⁴): <https://kovan.etherscan.io/address/0xc6d68c96964839215694d85ff63e05690e02fc64#code>

Внимание!

В домене перед Etherscan написано Kovan – это значит, что [Ethereum Block Explorer](#) будет брать информацию именно из этой тестовой сети. Мы будем пользоваться тестнетом Kovan, потому что в нем эфир бесплатный.

Вы можете читать информацию из контракта прямо из эксплорера: <https://kovan.etherscan.io/address/0xc6d68c96964839215694d85ff63e05690e02fc64#readContract>

Он вам позволяет вызывать read-only функции из контракта в реальном времени. Для того, чтобы взаимодействовать с контрактом (а не только читать его), нам надо отправлять на него транзакции, которые будут вызывать функции, меняющие его состояние.

В данном контракте есть только одна такая функция – **function vote(string _answer)**.

Инструкция по вызову этой функции с помощью MEW:

1. Для начала надо переключиться на сеть Kovan (в правом верхнем углу);
2. Выбрать закладку Contracts;
3. Ввести адрес контракта <https://kovan.etherscan.io/address/0xc6d68c96964839215694d85ff63e05690e02fc64#code>
4. Скопировать ABI контракта из Etherscan и вставить в поле ABI;
5. Нажать Access – и вперед. Вы можете читать read-only функции и слать транзакции для вызова write-функций.

⁴ - ABI – это описание интерфейса к контракту. Оно содержит в себе названия функций, параметры, которые они принимают (read или write), информацию о том, что они возвращают, и информацию о том, могут ли они принимать эфир на вход.

9. Задание №2

Вам нужно проголосовать в опросе. О том, что это за опрос и какие варианты ответов, можно прочитать в исходном коде контракта, или же с помощью read-only функций.

Я отправил на ваши адреса по 0.05 KETH (Kovan ETH), чтобы было чем за газ заплатить (ставьте gasPrice = 1 gwei).

Задавайте вопросы в нашем [Телеграм-чате!](#)