Аккаунт NEAR – Экспресс-гайд

Знакомство с учетной записью NEAR

1. NEAR использует имена (домены) аккаунтов, которые легко читаются человеком в формате name.near. Например, maria.near или jane.near.

2. Система учетных записей NEAR похожа на систему доменов веб-сайтов в том смысле, что учетная запись может создавать столько суб-учетов, сколько необходимо. Например, аккаунт с именем maria.near может создать аккаунт типа sub.maria.near, а он, в свою очередь, может создать аккаунты first.sub.masha.near, second.sub.maria.near и так далее.

3. NEAR Wallet (https://wallet.near.org/) (кошелек протокола NEAR), NEAR Faucet (https://faucet.paras.id/) (кран для пользователей Ethereum и Metamask) или near-cli (https://github.com/near/near-cli) (интерфейс командной строки, предоставляющий функциональность для интеграции NEAR) могут быть использованы для создания аккаунта.

4. В NEAR можно создать аккаунт и отправить его другу или подписчику в подарок с помощью сервиса https://nearnames.com.

5. Информацию об аккаунте можно проверить в NEAR Explorer (https://explorer.near.org/), а также в NEAR Wallet.

6. Помимо видимых аккаунтоа (тип name.near), экосистема NEAR также поддерживает создание невидимых счетов с помощью near-cli (они похожи на адреса Bitcoin и Ethereum). Подробное руководство на английском языке вы можете найти здесь.

7. Каждый аккаунт в системе может иметь только 1 смарт-контракт. Для приложений, которые требуют от пользователя использования нескольких смарт-контрактов, можно использовать дочерние аккаунты. Например, contract_1.maria.near, contract_2.maria.near и т.д.

8. В экосистеме NEAR существуют аккаунты разработчиков (https://docs.near.org/docs/concepts/account#dev-accounts). Их особенность заключается в том, что они созданы для тестирования и отладки смарт-контрактов.

УЧЕТНАЯ ЗАПИСЬ NEAR - Ключи

1. NEAR, как и большинство других блокчейнов, основан на криптографии с открытым ключом. Она опирается на пары ключей, каждая из которых состоит из открытого ключа (публичный ключ) и закрытого ключа (приватный ключ).

2. Открытый ключ используется в NEAR для идентификации, а закрытый - для подписания транзакций (подтверждение принадлежности счета при создании транзакции).

3. В NEAR существует 3 типа ключей. Ключи доступа предназначены для подписания транзакций со счета, ключи валидатора позволяют выполнять операции, связанные с проверкой сети, ключи ноды (сетевой ноды) позволяют осуществлять низкоуровневое взаимодействие между нодам сети.

4. Ключи могут храниться в 3 различных хранилищах. InMemoryKeyStore - хранилище в памяти, используется для временных сценариев. BrowserLocalStorageKeyStore - незашифрованное локальное хранилище браузера, используется для работы с приложениями в браузере. UnencryptedFileSystemKeyStore - незашифрованное хранилище в файловой системе, используется при работе с near-cli.

5. Учетная запись может иметь несколько ключей доступа или не иметь ни одного.

6. Ключи могут иметь различные уровни доступа - FullAccess (полный доступ) или FunctionCall (только возможность вызова методов контракта).

7. Все ключи уникальны в рамках одного аккаунта, но открытый ключ может быть назначен разным аккаунтам с разными уровнями доступа. Уровень доступа определяет, какие действия в учетной записи могут быть выполнены с помощью данного ключа.

8. Для уровня доступа FullAccess доступны все 8 типов действий: CreateAccountAction (создать аккаунт), DeployContractAction (развернуть контракт), FunctionCallAction (вызвать методы контракта), TransferAction (отправить токены на другой аккаунт), StakeAction (закрепить токены), AddKeyAction (добавить ключ в аккаунт), DeleteKeyAction (удалить ключ аккаунта), DeleteAccountAction (удалить аккаунт).

9. Для уровня доступа FunctionCall доступно только действие FunctionCallAction (вызов методов контракта). Кроме того, для такого ключа можно указать, какие методы контракта он может вызывать.