Solana vs EVM Chains

Overview for developers Suggest Edits

Solana-specific features

- Account type upon wallet creation
- : Solana only supports Externally Owned Accounts (EOAs) when creating a wallet. Smart account accounts (SCAs) cannot be created on Solana. This is in contrast to EVM chains, which allow you to specify an EOA or SCA account type during wallet creation.
- Transaction acceleration/cancellation
- : Solana does not offer the feature of transaction acceleration or cancellation, which are available in EVM chains.
- Signing message and typed data
- : Solana does not offer transaction acceleration or cancellation, which are capabilities in EVM chains.
- Associated Token Account (ATA)
- : When transferring SPLs in Solana, an ATA is automatically created for the recipient, which ensures that they can receive tokens. Since extra network fees are associated with the first transaction of a token, you must have enough fees available in your originating wallet.
- Gas Station
- : Gas Station is a feature under Programmable Wallets that lets developers sponsor network fees on behalf of their end users. Solana natively supports the Gas Station feature with
- feePayer
- , while EVM chains require using SCAs to utilize the gas station feature. We've integrated this Solana support into our Gas Station so that you can sponsor transaction fees for EOA wallets on Solana.
- Gas specification
- : The gas fee mechanism on Solana is similar to EIP-1559: you can specify a priority fee to incentivize validators to include your transaction in a block. Note that on Solano there is no concept of a "max fee" and
- gasLimit
- is optional. Solana defaults to a gas limit of 200,000 micro-lamport (10^-15 SOL).

Migrate from EVM chains to Solana

To transition from EVM chains to Solana, set the blockchains parameter value to SOL on mainnet, or SOL-DEVNET on testnet.

When creating wallets for end-users, follow these steps to allow them to transfer assets:

Step 1. Create Solana wallets

Fordeveloper-controlled wallets , use existing wallet sets to generate Solana wallets and change the following body params:

- accountType
- : Set to
- EOA
- blockchains
- : Set to
- SOL
- or
- SOL-DEVNET
- walletSetId
- : Set to the existing wallet set ID

Foruser-controlled wallets , use the same X-User-Token header parameter for the same end-user. If it expirescreate a new user token . Change the following body parameters:

- accountType
- : Set to
- EOA
- blockchains
- : Set to
- SOL
- or
- SOL-DEVNET

Next, for both wallets, specify the wallet metadata associated with an end-user during or after wallet creation:

- name

- : Set to the wallet name, for example:
- User Wallet1
- refId
- : Set to the unique identifier of the reference, for example:
- UUID1

Step 2. Request test token

To fund Solanatestnet wallets, update the following body params:

- blockchain
- :
- SOL-DEVNET
- address
- : Set to the newly created Solana wallet address

Step 3. Set up the Gas Station policy for Solana

Set up theGas Station policy for Solana in mainnet. Testnet environments come with pre-set default policies.

Step 4. Execute transactions

Specify the transaction detail in the following API endpoints:

- POST /transactions/transfer
- POST /transactions/transfer/estimateFee

Ensure the blockchains body parameter is set to SOL or SOL-DEVNET

Currently, the response to a POST request to the /transactions/transfer/estimateFee endpoint is the same for each fee level (high , medium , low). This will be updated in a future release. Updated14 days ago * Table of Contents * * Solana-specific features * * Migrate from EVM chains to Solana * * * Step 1. Create Solana wallets * * * Step 2. Request test token * * * Step 3. Set up the Gas Station policy for Solana * * * Step 4. Execute transactions