ERC-20 Paymaster FAQs

What is a paymaster?

A Paymaster is a special smart contract under the ERC-4337 specification that User Operations are able to delegate the responsibility of gas fee payments to. This means that ERC-4337 smart contract wallets no longer need to necessarily be responsible for directly paying gas fees in ETH. The paymaster contracts are able to use custom logic (with certain limitations) to decide whether or not they are willing to sponsor a UserOperation.

What is an ERC20 Paymaster?

An ERC20 Paymaster is a specific type of paymaster that is willing to sponsor the gas fees for a UserOperation if and only if the smart contract wallet pays the paymaster for it in an ERC20 token like USDC, DAI, etc. In effect, this allows smart contract wallets to pay for gas fees purely in ERC20 tokens and means, if designed correctly, they never need to hold any native tokens like ETH.

How do I use Pimlico's ERC20 Paymaster?

Check out our tutorial! We wrote a tutorial that takes you through the whole flow of deploying a SimpleWallet and sending your first UserOperation sponsored with USDC.

While the specifics can vary depending on the specific ERC20 token and chain you're using, the general steps are as follows:

Step 1: Find the appropriate ERC20 Paymaster address for the token and chain you're using.

You can do this either by going to our ERC20 Paymaster Contracts reference page or

Step 2: Approve ERC20 tokens to the ERC20 Paymaster so it can spent it on your behalf

Call the approve function on the ERC20 token you wish to use, specifying the corresponding ERC20 paymaster as the contract you are approving the tokens to.

Step 3: Generate the paymasterAndData to place in your UserOperation

The paymasterAndData for the ERC20 Paymaster can look in one of two ways.

You can either make the paymasterAndData simply the ERC20 Paymaster address. In this case, there is no restriction (beyond the token approval limit you set in the previous step) on how much you allow the ERC20 Paymaster to take. In this case, the paymasterAndData would be 20 bytes long.

Alternatively, you can make the paymasterAndData a concatenation of the ERC20 Paymaster Address, as well as a uint256 value that represents the maximum amount of ERC20 tokens you allow the paymaster to take. This can protect against unexpected price movements in between the time you submit the UserOperation and between the time it gets included, and serves a similar purpose as the amountOutMin or amountInMax parameters found in Uniswap's periphery contracts . If you specify the maximum ERC20 token amount, the paymasterAndData should be 56 bytes (20 for the address + 32 for the uint256) long.

Step 4: Submit the UserOperation

Do any final touches to your UserOperation, such as signing, then submit the UserOperation to a bundler like ours and watch it be included on-chain!

Is there a walkthrough I can use to see how to interact with the paymaster?

Yes!

Check out our Getting Started with the ERC20 Paymaster guide that walks you through deploying your SimpleWallet and getting your first UserOperation sponsored with USDC.

What ERC20 tokens and on what chains can I use Pimlico's ERC20 Paymaster with?

The currently supported tokens are listed here .

Theoretically, we can support any token on any EVM chains that have Chainlink support. If you have a token that you would like supported, pleasereach out to us !

Where can I find Pimlico's ERC20 Paymaster contract?

You can find the contract source codehere .

Does the ERC20 Paymaster use an admin upgradeable proxy?

No, we do not. The contracts we deploy can never be upgraded. However there are two variables that can be changed by the owner: thepriceMarkup and thepriceUpdateThreshold .

Has Pimlico's ERC20 Paymaster been audited?

Yes, the ERC20 Paymaster contract by Pimlico has been audited byNethermind Audits . The full audit report can be foundhere .

However, an audit does not guarantee complete security. Please use the paymaster at your own risk.

Is there an SDK I can use to interact with the ERC20 Paymaster?

You can use permissionless.js to easily interact with the ERC20 Paymaster. Check out thedocumentation for more information.

Is Pimlico's ERC20 Paymaster permissionless?

Yes. You do not need to go through our backend API to interact with the ERC20 Paymaster.

Is Pimlico's ERC20 Paymaster decentralized?

Currently, no. There is an owner in the ERC20 Paymaster contract that can update thepriceMarkup andpriceUpdateThreshold , as well as to withdraw the ERC20 tokens accumulated by the paymaster.

What admin controls does Pimlico have over the ERC20 Paymaster?

The owner of the ERC20 Paymaster has limited admin control. They can withdraw accumulated ERC20 tokens from the contract for the purpose of swapping them back into native tokens and depositing them back into the paymaster. Additionally, the owner can update thepriceMarkup andpriceUpdateThreshold configurations within predefined limits to manage operational costs and risks.

How can I use a token that is not currently supported by Pimlico?

Please get in touch with us. We will do what we can to see if we can add support for the token you are interested in.

What are the tradeoffs versus a non-permissionless paymaster?

A non-permissionless paymaster could potentially offer a slightly smoother user experience as it could handle token approvals during the execution phase of the UserOperation. However, this would require interaction with a hosted API, which could introduce potential points of failure or trust. By contrast, Pimlico's permissionless design avoids these potential issues but requires users to manually handle token approvals.

However, it is possible to bypass this extra step for smart contract wallets if they are able to make an approve call either during the deployment of the smart contract, or during the validation step of the UserOperation.

Does Pimlico take a fee?

Yes, Pimlico takes a fee in the form of apriceMarkup on the ERC20 token price. This markup serves to compensate us for maintaining the infrastructure and covering risks associated with price fluctuations and slippage risk. The defaultpriceMarkup we take is 10%, but that can depend on on the token, so double-check before using the paymaster.

ThepriceMarkup can be changed by the owner of the paymaster, however there is a hardcoded maximum of 20% that is enforced by the smart contract. This means we can never set thepriceMarkup to more than 20%.