Verifying Paymaster Endpoints

All calls are in JSON-RPC format and have to be made to the following URL:https://api.pimlico.io/v2/{chain}/rpc?apikey=[YOUR_API_KEY_HERE]

Where{chain} is the chain variable (such assepolia orpolygon) as found in thesupported chains page

pm_sponsorUserOperation (v2)

pm_sponsorUserOperation is the main endpoint for verifying paymasters. It takes in a User Operation, simulates it and estimates the gas limits, and then signs the User Operation with the verifying paymaster's key, sponsoring it when it is submitted on-chain. If successful, we deduct from your off-chain Pimlico balance.

You can optionally pass in a third parameter,sponsorshipPolicyId , which is a string that you can use to identify the sponsorship policy you want to use. To learn more about sponsorship policies, seeSponsorship Policies .

To use this endpoint, you must have an API key with Pimlico. Use ourQuick Start guide to generate one.

User Operations sponsored using pm_sponsorUserOperation have a 10 minute time window during which they must be included. After this time window elapses, all unused gas will be refunded to your Pimlico balance.

This time limit is necessary in order to avoid DoS attacks, as leaving an infinite time window would allow potential attackers to accumulate User Operations and drain Pimlico's paymaster all in one go, requiring us to maintain enough balance to cover all possible User Operations we ever signed up to sponsor in the entire history of the paymaster.

If you require a longer time window for your User Operations, please get in touch!

Request:

```

{ "jsonrpc":"2.0", "method":"pm_sponsorUserOperation", "params": [ { "sender":"0x1234567890123456789012345678901234567890", "nonce":"0x1", "initCode":"0x", "callData":"0x", "callGasLimit":"0x100000", "verificationGasLimit":"0x20000", "preVerificationGas":"0x10000", "maxFeePerGas":"0x3b9aca00", "maxPriorityFeePerGas":"0x3b9aca00", "paymasterAndData":"0x", "signature":"0x" }, "0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789", {"sponsorshipPolicyId":"sp_example_policy_id"}// optional ], "id":1 }

```

```

import{ JsonRpcProvider }from"@ethersproject/providers";

constchain="sepolia" constapiKey="YOUR_API_KEY_HERE" constentryPoint="0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789"

constuserOperation=...// generate your User Operation here

constprovider=newJsonRpcProvider(https://api.pimlico.io/v2/${chain}/rpc?apikey=${apiKey})

constresult=awaitprovider.send("pm_sponsorUserOperation", [userOperation, {entryPoint: entryPoint}])

```

Response:

```

{ "jsonrpc":"2.0", "result": { "paymasterAndData":"0xbcd12340a21098765432109876543010987654321987654321a21098765432109876543 0a210987654321098765430", "callGasLimit":"0x13210", "verificationGasLimit":"0x237328", "preVerificationGas":"0xa6d30" }, "id":1 }

```

pm_validateSponsorshipPolicies

This method validates a User Operation against an array ofsponsorship policies , and returns an array of sponsorship policies (alongside additional data for each policy) that are willing to sponsor the user operation.

Request:

```

{ "jsonrpc":"2.0", "method":"pm_validateSponsorshipPolicies", "params": [ { "sender":"0x1234567890123456789012345678901234567890", "nonce":"0x1", "initCode":"0x", "callData":"0x", "callGasLimit":"0x100000", "verificationGasLimit":"0x20000", "preVerificationGas":"0x10000", "maxFeePerGas":"0x3b9aca00", "maxPriorityFeePerGas":"0x3b9aca00", "paymasterAndData":"0x", "signature":"0x" }, "0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789", ["sp_crazy_kangaroo","sp_talented_turtle"] ], "id":1 }

```

Response:

```

{ "jsonrpc":"2.0", "result": [ { "sponsorshipPolicyId":"sp_crazy_kangaroo", "data": { "name":"Linea Christmas Week",// optional "author":"Linea",// optional "icon":"",// optional "description":"Linea is sponsoring the first 10 transactions for existing users between Christmas and New Year's Eve.",// optional } } ], "id":1 }

```

pm_supportedEntryPoints

This function returns the list of EntryPoint contracts that are supported on that chain.

Want to use an entryPoint that is not currently supported? Please contact us we can see if we can support it.

Request:

```

{ "jsonrpc":"2.0", "method":"pm_supportedEntryPoints", "params": [], "id":1 }

```

Response:

```

{ "jsonrpc":"2.0", "result": ["0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789"], "id":1 }

```

pm_sponsorUserOperation (v1)

v1 of the API is deprecated and will be removed in the future. Please use v2 instead.

User Operations sponsored using pm_sponsorUserOperation have a 10 minute time window during which they must be included. After this time window elapses, all unused gas will be refunded to your Pimlico balance.

This time limit is necessary in order to avoid DoS attacks, as leaving an infinite time window would allow potential attackers to accumulate User Operations and drain Pimlico's paymaster all in one go, requiring us to maintain enough balance to cover all possible User Operations we ever signed up to sponsor in the entire history of the paymaster.

If you require a longer time window for your User Operations, please get in touch!

Request:

```

{ "jsonrpc":"2.0", "method":"pm_sponsorUserOperation", "params": [ { "sender":"0x1234567890123456789012345678901234567890", "nonce":"0x1", "initCode":"0x", "callData":"0x", "callGasLimit":"0x100000", "verificationGasLimit":"0x20000", "preVerificationGas":"0x10000", "maxFeePerGas":"0x3b9aca00", "maxPriorityFeePerGas":"0x3b9aca00", "paymasterAndData":"0x", "signature":"0x" }, { "entryPoint":"0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789" } ], "id":1 }

```

```

import{ JsonRpcProvider }from"@ethersproject/providers";

constchain="sepolia" constapiKey="YOUR_API_KEY_HERE" constentryPoint="0x5FF137D4b0FDCD49DcA30c7CF57E578a026d2789"

constuserOperation=.../// generate your User Operation here

constprovider=newJsonRpcProvider(https://api.pimlico.io/v1/${chain}/rpc?apikey=${apiKey})

constpaymasterAndData=awaitprovider.send("pm_sponsorUserOperation", [userOperation, {entryPoint: entryPoint}])

```

Response:

```

{ "jsonrpc":"2.0", "result": { "paymasterAndData":"0xbcd12340a21098765432109876543010987654321987654321a2109876543210987654321a2109876543210987654321" }, "id":1 }

```