

Webhook APIs

API's to help programmatically allow our Paymasters to sponsor gas when specified conditions are met.

Auth Token

To obtain an authToken required in the header, go to your account settings page on the dashboard and generate an authToken: <https://dashboard.biconomy.io/account>

1. Create a Webhook:

POST Request URL: <https://paymaster-dashboard-backend.prod.biconomy.io/api/v2/public/sdk/webhook>

Parameters

Header

Param Type Description Required authToken string Token unique to every user account Required apiKey string API Key Associated with dApp Required Body

Param Type Description Required webhookUrl string URL where webhook is available Required requestType string Must be "GET" or "POST" Required Responses

200 OK { "statusCode" :

200 , "message" :

"Webhook created" , "data" :

{ "webhookId" :

"55564719-da8e-4f52-84b1-57da63097b9f" , "webhookUrl" :

"https://little-socks-yawn.locat.it" , "requestType" :

"POST" } } 401 Unauthorized { "statusCode" :

401 , "message" :

"Auth token and API key is required in the headers" }

2. Get a Registered Webhook:

GET Request URL: <https://paymaster-dashboard-backend.prod.biconomy.io/api/v2/public/sdk/webhook>

Parameters

Header

Param Type Description Required authToken string Token unique to every user account Required apiKey string API Key Associated with dApp Required Body

Param Type Description Required webhookUrl string empty string should be supplied Required Query Params

Param Type Description Required webhookId string Webhook ID which was returned on creation Required 200 OK { "statusCode" :

200 , "message" :

"Webhook list fetched" , "data" :

[{ "webhookId" :

"1704bab9-2173-4416-b897-71d41942014f" , "webhookUrl" :

"https://twitter.com/home" , "requestType" :

"POST" } } 401 Unauthorized { "statusCode" :

401 , "message" :

"Auth token is required in the headers" }

3. Update Webhook Status:

PATCH Request URL: <https://paymaster-dashboard-backend.prod.biconomy.io/api/v2/public/sdk/webhook>

Parameters

Header

Param Type Description Required authToken string Token unique to every user account Required apiKey string API Key Associated with dApp Required Body

Param Type Description Required webhookId string Webhook Id recieved when first created Required active boolean Address of smart contract Required Responses

200 OK { "statusCode" :

200 , "message" :

"Webhook updated" } 400 Bad Request Active must be a boolean value

{ "statusCode" :

400 , "message" :

"active must be a boolean value" } 401 Unauthorized { "statusCode" :

401 , "message" :

"Auth token and API key is required in the headers" } 404 Not Found { "statusCode" :

404 , "message" :

"Webhook not found" }

4. Delete Webhook:

DELETE Request URL: <https://paymaster-dashboard-backend.prod.biconomy.io/api/v2/public/sdk/webhook>

Parameters

Header

Param Type Description Required authToken string Token unique to every user account Required apiKey string API Key Associated with dApp Required Body

Param Type Description Required webhookId string Webhook Id recieved when first created Required 200 OK { "statusCode" :

200 , "message" :

"Webhook deleted" } 400 Bad Request Webhook ID must be a string

{ "statusCode" :

400 , "message" :

"webhookId must be a string" } 401 Unauthorized { "statusCode" :

401 , "message" :

"Auth token and API key is required in the headers" } 404 Not Found { "statusCode" :

404 , "message" :

"Webhook not found" } Using Webhooks with the SDK:

When building out the `paymasterServiceData` object you can optionally pass your `webhookData` to it. In the example below we pass a num value of 2 into the webhook data. Our webhook will check this data and verify if the number passed is an even or odd number.

```
let paymasterServiceData : SponsorUserOperationDto =
```

```
{ mode : PaymasterMode . SPONSORED , smartAccountInfo :
```

```
{ name :
```

```
"BICONOMY" , version :
```

```
"2.0.0" , } , calculateGasLimits :
```

```
true , webhookData :
```

```
{ num :
```

```
2 , } , } ; The webhookData gets passed to your webhook from our backend like this:
```

```
import axios from
```

```
"axios" ;
```

```
// POST const response =
```

```
await axios . post ( webhookUrl ,
```

```
{ data : webhookData , } ) ;
```

```
// GET const response =
```

```
await axios . get ( webhookUrl , webhookData ) ; Our backend expects a response in this format:
```

```
const webhookResponseData = response . data ; this . logger . log (webhookResponseData: { JSON . stringify ( webhookResponseData , ) } for dappld: { dappld } , ) ; const
```

```
{ arePoliciesVerified }
```

= webhookResponseData ; arePoliciesVerified should either be true or false based on which it gets determined if the webhook conditions are passed or not.

A sample webhook implementation that checks if the num data passed to is even:

```
const express =
```

```
require ( "express" ) ; const app =
```

```
express ( ) ;
```

```
app . use ( express . json ( ) ) ;
```

```
app . post ( "/" ,
```

```
( req , res )
```

```
=>
```

```
{ const data = req . body ; console . log ( "data" , data ) ; const
```

```
{ num }
```

```
= data . data ;
```

```
if
```

```
( num %
```

```
2
```

```
===
```

```
0 )
```

```
{ res . json ( { arePoliciesVerified :
```

```
true , } ) ; }
```

```
else
```

```
{ res . json ( { arePoliciesVerified :
```

```
false , } ) ; } } ) ;
```

```
app . listen ( 8080 ,
```

```
( )
```

```
=>
```

```
console . log ( "Server listening on port 8080!" ) ) ;Previous Sponsor UserOperation End PointNext Explorer
```