

Topup your Wallet in PnP Web No Modal SDK

For topping up the wallet of a user, you can use the `initiateTopup()` function. This method takes in the parameters of the selected payment provider and accordingly gives you the option to seamlessly add money to your user's wallet address directly.

note This is a paid feature and the minimum [pricing plan](#) to use this SDK in a production environment is the Growth Plan . You can use this feature in the development environment for free. info This uses the [Torus EVM Wallet UI Plugin](#) and [Torus Solana Wallet UI Plugin](#) for the respective chains. Please install and configure the packages before using this function.

`initiateTopup()`

[^](#)

- EVM
- Solana

Shows the TopUp modal to select local currency and amount to top up the wallet.

Example[^](#)

```
import
{
  WalletServicesPlugin
}
from
"@web3auth/wallet-services-plugin" ;
const walletServicesPlugin =
new
WalletServicesPlugin ( ) ; web3auth . addPlugin ( walletServicesPlugin ) ;
// Add the plugin to web3auth
await walletServicesPlugin . showCheckout ( ) ;
// Opens the TopUp modal Initiates the Top Up flow with selected provider.
```

Arguments[^](#)

- Table
- Interface

Parameter	Type	Description	Mandatory	provider	PAYMENT_PROVIDER_TYPE	["moonpay" , "wyre" , "rampnetwork" , "xanpool" , "mercuryo" , "transak"]	Payment Provider name	Yes	params	PaymentParams	Payment	Params	object	Yes
-----------	------	-------------	-----------	----------	-----------------------	--	-----------------------	-----	--------	---------------	---------	--------	--------	-----

PaymentParams

[^](#)

Parameter	Type	Description	Mandatory	selectedAddress?	string	Address to send the funds to	No	selectedCurrency?	string	Default fiat currency for the user to make the payment in	No	fiatValue?	number	Amount to buy in the selectedCurrency	No	selectedCryptoCurrency?	string	Cryptocurrency to buy	No	cryptoAmount?	number	Amount	Cryptocurrency to buy	No	async
-----------	------	-------------	-----------	------------------	--------	------------------------------	----	-------------------	--------	---	----	------------	--------	---------------------------------------	----	-------------------------	--------	-----------------------	----	---------------	--------	--------	-----------------------	----	-------

`initiateTopup (provider :`

`PAYMENT_PROVIDER_TYPE , params :`

`PaymentParams) :`

`Promise < void`

`{ if`

```

( ! this . torusWalletInstance . isLoggedIn )
throw
TorusWalletPluginError . web3AuthNotConnected ( ) ; await
this . torusWalletInstance . initiateTopup ( provider , params ) ; }
export
interface
PaymentParams
{ /* * Address to send the funds to/ selectedAddress ? :
string ; /* * Default fiat currency for the user to make the payment in/ selectedCurrency ? :
string ; /* * Amount to buy in the selectedCurrency/ fiatValue ? :
number ; /* * Cryptocurrency to buy/ selectedCryptoCurrency ? :
string ; /* * Amount Cryptocurrency to buy/ cryptoAmount ? :
number ; }
export
const
PAYMENT_PROVIDER
=
{ MOONPAY :
"moonpay" , WYRE :
"wyre" , RAMPNETWORK :
"rampnetwork" , XANPOOL :
"xanpool" , MERCURYO :
"mercuryo" , TRANSAX :
"transax" , }
as
const ;

```

Example [↗](#)

```

import
{
SolanaWalletConnectorPlugin
}
from
"@web3auth/solana-wallet-connector-plugin" ;
const torusPlugin =
new
SolanaWalletConnectorPlugin ( { torusWalletOpts :
{ } , walletInitOptions :

```

```

{ whiteLabel :
{ theme :
{
isDark :
true ,
colors :
{
primary :
"#00a8ff"
}
} , logoLight :
"https://web3auth.io/images/web3auth-logo.svg" , logoDark :
"https://web3auth.io/images/web3auth-logo---Dark.svg" , } , } , } ) ;
await web3auth . addPlugin ( torusPlugin ) ;
// add plugin to web3auth instance
await torusPlugin . initiateTopup ( "moonpay" ,
{ selectedAddress :
"wallet_address" , selectedCurrency :
"USD" ,
// Fiat currency fiatValue :
100 ,
// Fiat Value selectedCryptocurrency :
"ETH" ,
// Cryptocurrency SOL, MATIC etc. } ) ;

```

Supported Cryptocurrency^â

Supported Cryptocurrency for each Payment Provider by Network

MOONPAY

^â

Network Supported Cryptocurrency mainnet AAVE ,BAT ,DAI ,ETH ,MKR ,MATIC ,USDT ,USDC matic ETH :eth_polygon ,MATIC :matic_polygon ,USDC :usdc_polygon bsc_mainnet BNB :bnb_bsc ,BUSD :busd_bsc avalanche_mainnet AVAX :avax_cchain

WYRE

^â

Network Supported Cryptocurrency mainnet AAVE ,BAT ,BUSD ,DAI ,ETH ,MKR ,UNI ,USDC ,USDT matic USDC :MUSDC avalanche_mainnet AVAX

RAMPNETWORK

^â

Network Supported Cryptocurrency mainnet ETH ,DAI ,USDC ,USDT matic MATIC_DAI ,MATIC_MATIC ,MATIC_USDC

avalanche_mainnet AVAX

XANPOOL

[â](#)

Network Supported Cryptocurrency mainnet ETH ,USDT

MERCURYO

[â](#)

Network Supported Cryptocurrency mainnet ETH ,BAT ,USDT ,DAI bsc_mainnet BNB ,BUSD ,1INCH

TRANSAK

[â](#)

Network Supported Cryptocurrency mainnet AAVE ,DAI ,ETH ,USDC ,USDT ,CHAIN matic AAVE ,DAI ,MATIC ,USDC ,USDT ,WETH ,CHAIN bsc_mainnet BNB ,BUSD avalanche_mainnet AVAX

BANXA

[â](#)

Network Supported Cryptocurrency mainnet ETH ,USDT ,BUSD ,LINK ,USDC ,CHZ ,BAT ,MANA ,AAVE ,COMP ,ENJ matic MATIC [Edit this page](#) [Previous Multi Factor Authentication](#) [Next Show Wallet Connect Scanner](#)