

[illegible]

...

The best practice is to check if the wallet is an Argent wallet using the wallet detector contract and then batch the transactions:

...

```
Copy constwalletDetectorAddress="0xeca4B0bDBf7c55E9b7925919d03CbF8Dc82537E8"; constwalletDetectorABI=[ "function isArgentWallet(address _wallet) external view returns (bool)" ];
```

```
constwalletDetector=newethers.Contract(walletDetectorAddress,walletDetectorABI,provider); constisArgent=awaitwalletDetector.isArgentWallet(address);
```

```
if(isArgent) { // do a multicall }else{ // do single calls }
```

...

Meta transactions and relayers

Argent wallet uses the concept of meta transactions. These are a particular type of transaction which are signed by one or more key pairs (in our case the wallet's owner and potentially their guardians) but are submitted to the Ethereum network by a relayer. The relayer pays the gas fee (in ETH) and the wallet will refund the relayer (in ETH or ERC20 tokens) up to an amount signed by the wallet's owner.

From the dapp's perspective, this is managed by the Argent mobile application. The dapp will submit a regular{ to, value, data } transaction to the web3 provider. This transaction will be transmitted to the Argent mobile application through WalletConnect. The mobile wallet will transform the data into a meta transaction:

- to
- will be the ArgentRelayerManager
- contract address
- data
- will be the encoded data of the call to theexecute()
- method with the relevant parameters
-

The dapp will receive the transaction hash in order to monitor the status of the transaction and events will be emitted as usual.

Our relayer has the ability to replay a transaction with a higher gas price due to fluctuating network conditions. The transaction hash is modified and the dapp will not be aware of the new transaction hash.

One solution could be for the dapp to observe a specific event being emitted instead of transaction status. We are working on defining a standard for a dapp to be notified when a transaction is replaced and the transaction hash changes (contact us if you are interested in this).

Login UX best practice

To improve the login experience for Argent users, we recommend showing an Argent branded login button at the top level. This ensures people less familiar with WalletConnect can log in to your app. When clicking on the Argent login, you then present the WalletConnect QR code.

You can find an example of this implementation on <https://app.flexa.network/connect>

?

FAQ

Can I use the ERC20 permit function?

It depends on how the ERC20 smart contract implements thepermit() method. If you look below (source [Etherscan](#)) at the Maker DAI implementation, you will notice at line 24 the use ofecrecover() and not EIP-1271 to verify the signature. Here, the call will fail with Argent wallets.

...

```
Copy functionpermit( address holder, address spender, uint256 nonce, uint256 expiry, bool allowed, uint8 v, bytes32 r, bytes32 s )external{ bytes32 digest= keccak256(abi.encodePacked( "\x19\x01", DOMAIN_SEPARATOR, keccak256(abi.encode(PERMIT_TYPEHASH, holder, spender, nonce, expiry, allowed)) ));
```

```
require(holder!=address(0),"Dai/invalid-address-0"); require(holder==ecrecover(digest,v,r,s),"Dai/invalid-permit"); require(expiry==0||now<=expiry,"Dai/permit-expired"); require(nonce==nonces[holder]++, "Dai/invalid-nonce"); uint wad=allowed?uint(-1):0; allowance[holder][spender]=wad; emitApproval(holder,spender,wad); }
```

...

I'm asking for ERC20 approval of an amount of! 1 0 53 10^{53} but it doesn't seem to work

Argent may override the "infinite value" and ask a user to input a smaller amount due to the additional security risk of allowing large persistent approvals. When checking the allowance in your dapp you need to check for the exact amount the user wants to spend.

Can I use 3rd party relayers with Argent?

Yes, but the relayer needs to implement a call to theexecute() method of ourRelayerManager smart contract. You will have to get the wallet owner signing the correct payload which is not straightforward. Please contact us if you're interested in this topic.

Do you estimate gas?

Yes, we useeth_estimateGas before sending any transaction.

Do you support WalletConnect mobile deeplinks?

Yes, WalletConnect will work on mobile browsers (or native mobile apps) and will deep link to the Argent wallet.

How can I test the integration?

You can test Argent on Ropsten on both iOS and Android. Contact dapps@argent.xyz to receive a test build.

[Previous Ethereum Next Dappland](#)

Last updated11 months ago