

Liquidations

The health of the Aave Protocol is dependent on the 'health' of the collateralised positions within the protocol, also known as the 'health factor'. When the 'health factor' of an account's total loans is below 1, anyone can make `liquidationCall()` to the [Pool](#) or [L2Pool](#) (in case of Arbitrum/Optimism) contract, pay back part of the debt owed and receive discounted collateral in return (also known as the liquidation bonus).

This incentivises third parties to participate in the health of the overall protocol, by acting in their own interest (to receive the discounted collateral) and as a result, ensure borrows are sufficiently collateralize.

There are multiple ways to participate in liquidations:

1. By calling the `liquidationCall()` directly in the [Pool](#)
2. or [L2Pool](#)
3. contract.
4. By creating your own automated bot or system to liquidate loans.
- 5.

For liquidation calls to be profitable, you must take into account the gas cost involved in liquidating the loan. If a high gas price is used, then the liquidation may be unprofitable for you. See the [Calculating profitability](#) section for more details. V3 allows 100% of debt (i.e. `MAX_LIQUIDATION_CLOSE_FACTOR`) to be liquidated in single `liquidationCall()` if: $HF < CLOSE_FACTOR_HF_THRESHOLD$

Prerequisites

When making `liquidationCall()`, you must:

- Know the account (i.e. the ethereum address: `user`)
- whose health factor is below 1.
- Know the valid debt amount and asset (i.e. `debtToCover` & `debtAsset`)
-)
- - If the HF is above `CLOSE_FACTOR_HF_THRESHOLD`
 - , then only a maximum of 50% (i.e. `DEFAULT_LIQUIDATION_CLOSE_FACTOR`)
 -) of the debt can be liquidated per `validliquidationCall()`
 - If the HF is below `CLOSE_FACTOR_HF_THRESHOLD`
 - , then 100% (i.e. `MAX_LIQUIDATION_CLOSE_FACTOR`)
 -) of the debt can be liquidated in single `validliquidationCall()`
 - You can set the `debtToCover`
 - to `uint(-1)`
 - and the protocol will proceed with the highest possible liquidation allowed by the close factor.
 - You must already have sufficient balance of the debt asset, which will be used by the `liquidationCall`
 - to pay back the debt. You can use `flashLoan`
 - for liquidations 😊
- *
- Know the collateral asset `collateralAsset`
- you closing, i.e. the asset that the user has backing
- their outstanding loan that you will receive as a bonus
- .
- Whether you want to receive `aTokens`
- or the underlying asset after a successful `liquidationCall()`
- .
-

Getting accounts to liquidate

"User Account" in the Aave Protocol refer to a single ethereum address that has interacted with the protocol. This can be an externally owned account or contract. Only user accounts that have $HF < 1$ can be liquidated. There are multiple ways you can get the health factor:

On Chain

1. To gather user account data from on-chain data, one way would be to monitor emitted events from the protocol and keep an up to date index of user data locally.
2.
 1. Events are emitted each time a user interacts with the protocol (supply, borrow, repay, withdraw etc.)
3. 2.
4. When you have the user's address, you can simply call [getUserAccountData\(\)](#)
5. , to read the user's current healthFactor. If the $HF < 1$, then the account can be liquidated.
- 6.

GraphQL

1. Similarly to the sections above you will need to gather user account data and keep an index of the user data locally.
2. Since GraphQL does not provide real time calculated user data such as healthFactor,
3. you will need to compute this locally. The easiest way is to use the [Aave-utilities](#)
4. sdk, which has methods to compute user summary data.
- 5.

Executing the liquidation call

Once you have the account(s) to liquidate, you will need to calculate the amount of collateral that can be liquidated:

1. Use [getUserReserveData\(\)](#)
2. Max debt that be cleared by single liquidation call is given by the `DEFAULT_LIQUIDATION_CLOSE_FACTOR`
3. (when $CLOSE_FACTOR_HF_THRESHOLD < HF < 1$
4.) or `MAX_LIQUIDATION_CLOSE_FACTOR`
5. (when $HF < CLOSE_FACTOR_HF_THRESHOLD$
6.)
7.
 1. $debtToCover = (userStableDebt + userVariableDebt) * LiquidationCloseFactor$
8.
 1. You can pass `uint(-1)`
9.
 1. , i.e. `MAX_UINT`
10.
 1. , as the `debtToCover`
11.
 1. to liquidate the maximum amount allowed.
12. 6.
13. Max amount of collateral that can be liquidated to cover debt is given by the current `liquidationBonus`
14. for the reserves that have `usageAsCollateralEnabled`
15. as `true`.
16.
 1. $maxAmountOfCollateralToLiquidate = (debtAssetPrice * debtToCover * liquidationBonus) / collateralPrice$
17. 2.
- 18.

Calculating profitability vs gas cost

One way to calculate the profitability is the following:

1. Store and retrieve each collateral's relevant details such as address, decimals used and liquidation bonus.
2. Get the user's collateral balance (`aTokenBalance`).
3. Get the asset's price according to the Aave's oracle contract using [getAssetPrice\(\)](#)
4. .
5. The maximum collateral bonus received on liquidation is given by $the maxAmountOfCollateralToLiquidate * (1 - liquidationBonus) * collateralAssetPriceEth$
6. The maximum cost of your transaction will be you gas price multiplied by the amount of gas used. You should be able to get a good estimation of the gas amount used by calling `estimateGas`
7. via your web3 provider.
8. Your approximate profit will be the value of the collateral bonus (4) minus the cost of your transaction (5).
- 9.

Appendix

How is health factor calculated?

The health factor is calculated from the user's total collateral, i.e. all reserves for which `usageAsCollateral` is enabled, balance (in ETH) multiplied by the liquidation threshold percentage for all the user's outstanding assets, divided by the user's total borrow balance across all reserves (in ETH).

This can be calculated both off-chain and on-chain, see [Aave-utilities](#) and [GenericLogic Library](#) respectively for reference.

How is liquidation bonus determined?

Liquidation bonuses for all the assets are evaluated and determined based on each asset's liquidity risk and updated via Aave Governance process.

[Previous Flash Loans](#) [Next Testing Guide](#) Last updated 1 year ago On this page * [Prerequisites](#) * [Getting accounts to liquidate](#) * [On Chain](#) * [GraphQL](#) * [Executing the liquidation call](#) * [Calculating profitability vs gas cost](#) * [Appendix](#) * [How is health factor calculated?](#) * [How is liquidation bonus determined?](#)

Was this helpful?