Pool

Pool

Thepool.sol contract is the main user facing contract of the protocol. It exposes the liquidity management methods that can be invoked using eitherSolidity orWeb3 libraries.

Write Methods

supply

function supply(address asset, uint256 amount, address onBehalfOf, uint16 referralCode)

ThereferralCode is emitted in Supply event and can be for third party referral integrations. To activate referral feature and obtain a unique referral code, integrators need to submit proposal to Aave Governance.

When supplying, thePool contract must have**allowance() to spend funds on behalf of msg.sender ** for at-least**amount ** for theasset being supplied. This can be done via the standard ERC20approve() method on the underlying token contract Referral supply is currently inactive, you can pass0 asreferralCode . This program may be activated in the future through an Aave governance proposal Param Name Type Description asset address address of the asset being supplied to the pool. amount uint256 amount of asset being supplied. onBehalfOf address address that will receive the corresponding aTokens.

Note: only the onBehalfOf address will be able to withdraw asset from the pool. referralCode uint16 unique code for 3rd party referral program integration. Use 0 for no referral.

supplyWithPermit

function supplyWithPermit(address asset, uint256 amount, address onBehalfOf, uint16 referralCode, uint256 deadline, uint8 permitV, permitR, bytes32 permitS)

Supply with transfer approval of supplied asset via permit function. This method removes the need for separate approval tx before supplying asset to the pool.

Permit signature must be signed bymsg.sender with spender as Pool address. Referral program is currently inactive, you can pass0 asreferralCode . This program may be activated in the future through an Aave governance proposal Call Params

Name Type Description asset address Address of underlying asset being supplied. Same asset as used in permit s,v,r amount uint256 Amount of asset to be supplied and signed for approval. Same amount as used in permit s,v,r onBehalfOf address Address that will receive the aTokens. referralCode uint16 unique code for 3rd party referral program integration.

Use 0 for no referral. deadline uint256 unix timestamp up-till which signature will be valid permitV uint8 Signature parameter v permitR bytes32 Signature parameter r permitS bytes32 Signature parameter s

withdraw

function withdraw(address asset, uint256 amount, address to)

Withdrawsamount of the underlyingasset , i.e. redeems the underlying token and burns the aTokens.

If user has any existing debt backed by the underlying token, then the maxamount available to withdraw is theamount that will not leave user health factor < 1 after withdrawal.

When withdrawingto another address,msg.sender should haveaToken that will be burned by Pool . Call Params

Name Type Description asset address address of the underlying asset, not the aToken amount uint256 amount deposited, expressed in wei units. Usetype(uint).max to withdraw the entire balance. to address address that will receive theasset

borrow

function borrow(address asset, uint256 amount, uint256 interestRateMode, uint16 referralCode, address onBehalfOf)

Borrowsamount ofasset withinterestRateMode , sending theamount tomsg.sender , with the debt being incurred byonBehalfOf .

Note: IfonBehalfOf is not same asmsg.sender , thenonBehalfOf must have supplied enough collateral viasupply() and have delegated credit tomsg.sender viaapproveDelegation() . Referral program is currently inactive, you can pass0 asreferralCode . This program may be activated in the future through an Aave governance proposal Call Params

Name Type Description asset address address of the underlying asset amount uint256 amount to be borrowed, expressed in wei units interestRateMode uint256 the type of borrow debt.

Stable: 1, Variable: 2 referralCode uint16 referral code for our referral program. Use 0 for no referral code. onBehalfOf address address of user who will incur the debt.

Use msg.sender when not calling on behalf of a different user.

repay

function repay(address asset, uint256 amount, uint256 rateMode, address onBehalfOf)

RepaysonBehalfOf 's debtamount ofasset which has arateMode .

When repaying, thePool contract must have**allowance() to spend funds on behalf of msg.sender ** for at-least**amount ** for theasset you are repaying with. This can be done via the standard ERC20approve() method on the underlying token contract. Referral program is currently inactive, you can pass0 asreferralCode . This program may be activated in the future through an Aave governance proposal Call Params

Name Type Description asset address address of the underlying asset amount uint256 Amount of underlying asset being repaid. Use uint(-1) to repay the entire debt, ONLY when the repayment is not executed on behalf of a 3rd party. In case of repayments on behalf of another user, it's recommended to send an _amount slightly higher than the current borrowed amount. rateMode uint256 the type of debt being repaid. Stable: 1, Variable: 2 onBehalfOf address address of user who will incur the debt. Use msg.sender when not calling on behalf of a different user.

repayWithPermit

function repayWithPermit(address asset, uint256 amount, uint256 interestRateMode, address onBehalfOf, uint256 deadline, uint8 permitV, permitR, bytes32 permitS)

Repay with transfer approval of borrowed asset via permit function. This method removes the need for separate approval tx before repaying asset to the pool.

Permit signature must be signed bymsg.sender with spender value asPool address. Call Params

Name Type Description asset address Address of underlying asset being supplied. Same asset as used in permit s,v,r amount uint256 Amount of asset to be supplied and signed for approval. Same amount as used in permit s,v,r interestRateMode uint256 the type of debt being repaid. Stable: 1, Variable: 2 onBehalfOf address Address that will receive the aTokens. deadline uint256 unix timestamp up-till which signature will be valid permitV uint8 Signature parameter v permitR bytes32 Signature parameter r permitS bytes32 Signature parameter s

repayWithATokens

function repayWithATokens(address asset,uint256 amount,uint256 interestRateMode)

Allows user to repay withaTokens of the underlying debt asset without any approvals eg. Pay DAI debt using aDAI tokens.

Call Params

Param Name Type Description asset address Address of the underlying asset to be repaid amount uint256 Amount of underlying asset being repaid. Use uint256(-1) to pay without leaving aToken dust interestRateMode uint256 Interest rate mode of the debt position 1 - stable 2 - variable

swapBorrowRateMode

function swapBorrowRateMode(address asset, uint256 rateMode)

Swapsmsg.sender 's borrow rate mode between stable and variable.

Call Params

Name Type Description asset address address of the underlying asset rateMode uint256 the rate mode the user is swapping from. Stable: 1, Variable: 2

rebalanceStableBorrowRate

function rebalanceStableBorrowRate(address asset, address user)

Rebalances stable borrow rate of theuser for givenasset . In certain conditions, the protocol allows for stable rates to be rebalanced to avoid a large percentage of liquidity being borrowed at a stable rate below market variable rate. In V3, The condition for rebalance is if the current supply rate <= supply rate if all borrows are variable * 0.9,contract reference .

Call Params

Name Type Description asset address Address of the underlying token that has been borrowed for which the position is being rebalanced. user address Address of the user being rebalanced.

setUserUseReserveAsCollateral

function setUserUseReserveAsCollateral(address asset, bool useAsCollateral)

Sets theasset ofmsg.sender to be used as collateral or not.

An asset inIsolation Mode can be enabled to use as collateral only if no other asset is already enabled to use as collateral. User won’t be able to disable an asset as collateral if they have an outstanding debt position which could be left with HF <HEALTH_FACTOR_LIQUIDATION_THRESHOLD on disabling the given asset as collateral. Call Params

Name Type Description asset address address of the underlying asset to be used as collateral useAsCollateral bool true if the asset should be used as collateral

liquidationCall

function liquidationCall(address collateral, address debt, address user, uint256 debtToCover, bool receiveAToken)

Liquidate positions with ahealth factor below 1.

When the health factor of a position is below 1, liquidators repay part or all of the outstanding borrowed amount on behalf of the borrower, whilereceiving a discounted amount of collateral in return (also known as a liquidation 'bonus"). Liquidators can decide if they want to receive an equivalent amount of collateralaTokens instead of the underlying asset. When the liquidation is completed successfully, the health factor of the position is increased, bringing the health factor above 1.

Liquidators can only close a certain amount of collateral defined by a close factor. Currently theclose factor is 0.5 . In other words, liquidators can only liquidate a maximum of 50% of the amount pending to be repaid in a position. The liquidation discount applies to this amount.

Liquidators must approve() the Pool contract to use debtToCover of the underlying ERC20 of theasset used for the liquidation. NOTES

- In most scenarios
- , profitable liquidators will choose to liquidate as much as they can (50% of theuser
- position).
- debtToCover
- parameter can be set touint(-1)
- and the protocol will proceed with the highest possible liquidation allowed by the close factor.
- To check a user's health factor, usegetUserAccountData() .
-

Call Params

Name Type Description collateral address address of the collateral reserve debt address address of the debt reserve user address address of the borrower debtToCover uint256 amount of asset debt that the liquidator will repay receiveAToken bool if true, the user receives the aTokens equivalent of the purchased collateral. If false, the user receives the underlying asset directly.

flashLoan

function flashLoan(address receiverAddress, address[] calldata assets, uint256[] calldata amounts, uint256[] interestRateModes, address onBehalfOf, bytes calldata params, uint16 referralCode)

Allows users to access liquidity of the pool for givenlist of assets for one transaction as long as the amount taken plus fee is returned or debt position is opened by the end of transaction.

If no debt position is opened, receiver must approve thePool contract for at least theamount borrowed + fee , else transaction will revert. Flash loan fee is waived for the approvedflashBorrowers Referral program is currently inactive, you can pass0 asreferralCode . This program may be activated in the future through an Aave governance proposal Call Params

Name Type Description receiverAddress address Address of the contract that will receive the flash borrowed funds. Must implement the IFlashLoanReceiver interface. assets address[] Addresses of the underlying assets that will be flash borrowed amounts uint256[] Amounts of assets being requested for flash borrow. This needs to contain the same number of entries as assets. modes uint256[] the types of debt position to open if the flashloan is not returned. 0: no open debt. (amount+fee must be paid in this case or revert) 1: stable mode debt 2: variable mode debt onBehalfOf address if the associated mode is not0 then the incurred debt will be applied to the onBehalfOfaddress. Note: onBehalfOf must already have approved sufficient borrow allowance of the associated asset to msg.sender params bytes Arbitrary bytes-encoded params that will be passed to executeOperation() method of the receiver contract. referralCode uint16 Referral Code used for 3rd party integration referral. The unique referral id is can be requested via governance proposal flashLoanSimple

function flashLoanSimple(address receiverAddress, address asset, uint256 amount, bytes calldata params, uint16 referralCode)

Allows users to access liquidity ofone reserve or one transaction as long as the amount taken plus fee is returned.

Receiver must approve thePool contract for at least theamount borrowed + fee, else transaction will revert. Does not waive few for approvedflashBorrowers nor allow opening a debt position instead of repaying. Referral program is currently inactive, you can pass0 asreferralCode . This program may be activated in the future through an Aave governance proposal Call Params

Name Type Description receiverAddress address Address of the contract that will receive the flash borrowed funds. Must implement the IFlashLoanReceiver interface. asset address Address of the underlying asset that will be flash borrowed. amount uint256 Amount of asset being requested for flash borrow params bytes Arbitrary bytes-encoded params that will be passed to executeOperation() method of the receiver contract. referralCode uint16 Referral Code used for 3rd party integration referral. The unique referral id is can be requested via governance proposal

mintToTreasury

function mintToTreasury(address[] calldata assets)

Mints reserve income accrued to treasury (as per the reserve factor) for the given list of assets.

Call Params

Name Type Description assets address[] List of assets for which accrued income is minted

setUserEMode

function setUserEMode(uint8 categoryId)

Updates the user efficiency mode category. The category id must be a valid id already defined byPool or Risk Admins

Will revert if user is borrowing non-compatible asset or change will drop HF <HEALTH_FACTOR_LIQUIDATION_THRESHOLD Call Params

Name Type Description categoryId uint8 eMode category id (0 - 255) defined by Risk or Pool Admins. categoryId == 0 ⇒ non E-mode category.

mintUnbacked

mintUnbacked (asset, amount, onBehalfOf, referralCode)

Allows contracts, withBRIDGE role permission, to mint unbackedaTokens to theonBehalfOf address. This method is part of the V3Portal feature.

Only available to the addresses withBRIDGE role. Bridge addresses can be whitelisted by the governance. Param Type Description asset address address of the underlying asset token amount uint256 the amount to be minted onBehalfOf address the address which will receive the aTokens referralCode uint16 Code used to register the integrator originating the operation, for potential rewards

0 if the action is executed directly by the user, without any middle-man

backUnbacked

backUnbacked (asset, amount, fee)

Allows contracts, withBRIDGE role permission, to back the currently unbacked aTokens withamount of underlying asset and payfee . This method is part of the V3Portal feature.

Only available to the addresses withBRIDGE role. Bridge addresses can be whitelisted by the governance. Param Type Description asset address address of the underlying asset to repay amount uint256 amount of asset supplied to back the unbacked tokens fee uint256 amount paid in fee

rescueTokens

function rescueTokens(address token, address to, uint256 amount)

Rescue and transfer tokens locked in this contract.

Only available toPOOL_ADMIN role. Pool admin is selected by the governance.

View Methods

getReserveData

function getReserveData(address asset)

Returns the state and configuration of the reserve.

Call Params

Name Type Description asset address address of the underlying reserve asset. Return Values

Name Type Description configuration uint256 Reserve configuration. bit 0-15: LTV bit 16-31: Liquidation threshold bit 32-47: Liquidation bonus bit 48-55: Decimals bit 56: reserve is active bit 57: reserve is frozen bit 58: borrowing is enabled bit 59: stable rate borrowing enabled bit 60: asset is paused bit 61: borrowing in isolation mode is enabled bit 62-63: reserved bit 64-79: reserve factor bit 80-115: borrow cap in whole tokens, 0 ⇒ no cap bit 116-151: supply cap in whole tokens, 0 ⇒ no cap bit 152-167: liquidation protocol fee bit 168-175: eMode category bit 176-211: unbacked mint cap in whole tokens, 0 ⇒ no cap bit 212-251: debt ceiling for isolation mode with decimals bit 252-255: unused liquidityIndex uint128 yield generated by reserve during time interval since lastUpdatedTimestamp. Expressed in ray currentLiquidityRate uint128 current supply rate. Expressed in ray variableBorrowIndex uint128 yield accrued by reserve during time interval since lastUpdatedTimestamp. Expressed in ray currentVariableBorrowRate uint128 current variable borrow rate. Expressed in ray currentStableBorrowRate uint128 current stable borrow rate. Expressed in ray lastUpdateTimestamp uint40 timestamp of when reserve data was last updated. Used for yield calculation. id uint16 reserve's position in the list of active reserves. aTokenAddress address address of associated aToken stableDebtTokenAddress address address of associated stable debt token variableDebtTokenAddress address address of associated variable debt token interestRateStrategyAddress address address of interest rate strategy. accruedToTreasury uint128 the current treasury balance (scaled) unbacked uint128 the outstanding unbacked aTokens minted through the bridging feature isolationModeTotalDebt uint128 the outstanding debt borrowed against this asset in isolation mode

getUserAccountData

function getUserAccountData(address user)

Returns the user account data across all the reserves

Call params

Name Type Description user address address of the user Return Values

Name Type Description totalCollateralBase uint256 total collateral of the user, in market's base currency totalDebtBase uint256 total debt of the user, in market's base currency availableBorrowsBase uint256 borrowing power left of the user, in market's base currency currentLiquidationThreshold uint256 liquidation threshold of the user - weighted average of liquidtion threshold of collateral reserves ltv uint256 maximum Loan To Value of the user - weighted average of max ltv of collateral reserves healthFactor uint256 current health factor of the user

getConfiguration

function getConfiguration(address asset)

Returns the configuration of the reserve.

Call Params

Name Type Description asset address address of the underlying reserve asset. Return Values

Name Type Description configuration uint256 Reserve configuration. bit 0-15: LTV bit 16-31: Liquidation threshold bit 32-47: Liquidation bonus bit 48-55: Decimals bit 56: reserve is active bit 57: reserve is frozen bit 58: borrowing is enabled bit 59: stable rate borrowing enabled bit 60: asset is paused bit 61: borrowing in isolation mode is enabled bit 62-63: reserved bit 64-79: reserve factor bit 80-115: borrow cap in whole tokens, 0 ⇒ no cap bit 116-151: supply cap in whole tokens, 0 ⇒ no cap bit 152-167: liquidation protocol fee bit 168-175: eMode category bit 176-211: unbacked mint cap in whole tokens, 0 ⇒ no cap bit 212-251: debt ceiling for isolation mode with decimals bit 252-255: unused

getUserConfiguration

function getUserConfiguration(address user)

Returns the configuration of the user across all the reserves.

Call Params

Name Type Description user address address of the user Return Values

Type Description uint256 Bitmap of the users collaterals and borrows. Its divided into pairs of bits, one pair for each asset. The first bit of the pair indicates if it is being used as collateral by the user, the second bit indicates if it is being borrowed. The corresponding assets are in the same position as getReservesList() For example, if the hex value returned is 0x40020, which represents a decimal value of 262176, then in binary it is 1000000000000100000. If we format the binary value into pairs, starting from the right, we get 1 00 00 00 00 00 00 10 00 00. If we start from the right and move left in the above binary pairs, the third pair is 10. Therefore the 1 indicates that third asset from the reserveList is used as collateral, and 0 indicates it has not been borrowed by this user.

getReserveNormalizedIncome

function getReserveNormalizedIncome(address asset)

Returns the ongoing normalized income for the reserve.

A value of 1e27 means there is no income. As time passes, the yield is accrued. A value of 2*1e27 means for each unit of asset one unit of income has been accrued.

Return Value

Type Description uint256 Normalized income, expressed in ray

getReserveNormalizedDebt

function getReserveNormalizedVariableDebt(address asset)

Returns the ongoing normalized variable debt for the reserve.

A value of 1e27 means there is no debt. As time passes, the debt is accrued. A value of 2*1e27 means that for each unit of debt, one unit worth of interest has been accumulated.

Return Value

Type Description uint256 Normalized debt, expressed in ray

getReservesList

function getReservesList()

Returns the list of initialized reserves.

getEModeCategoryData

function getEModeCategoryData(uint8 id)

Returns category data for the given eModeCategory id.

Return Values

Type Description uint16 loan to value (ltv) for the given eModeCategory id uint16 liquidationThreshold for the given eModeCategory id uint16 liquidationBonus for the given eModeCategory id address custom price source for the eMode category string custom label describing the eMode category

getUserEMode

function getUserEMode(address user)

Returns eModeCategory Id of the user's eMode. 0 ⇒ no eMode.

FLASHLOAN_PREMIUM_TOTAL

function FLASHLOAN_PREMIUM_TOTAL() public view returns (uint128)

Returns the percent of total flashloan premium paid by the borrower. A part of this premium is added to reserve's liquidity index i.e. paid to the liquidity provider and the other part is paid to the protocol i.e. accrued to the treasury.

FLASHLOAN_PREMIUM_TO_PROTOCOL

function FLASHLOAN_PREMIUM_TO_PROTOCOL() public view returns (uint128)

Returns the percent of flashloan premium that is accrued to the treasury.

ABI

Pool ABI ```

Copy [{ "inputs": [{ "internalType": "contract IPoolAddressesProvider", "name": "provider", "type": "address" }],
"stateMutability": "nonpayable", "type": "constructor" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType":
"address", "name": "reserve", "type": "address" }, { "indexed": true, "internalType": "address", "name": "backer", "type":
"address" }, { "indexed": false, "internalType": "uint256", "name": "amount", "type": "uint256" }, { "indexed": false,
"internalType": "uint256", "name": "fee", "type": "uint256" }], "name": "BackUnbacked", "type": "event" }, { "anonymous":
false, "inputs": [{ "indexed": true, "internalType": "address", "name": "reserve", "type": "address" }, { "indexed": false,
"internalType": "address", "name": "user", "type": "address" }, { "indexed": true, "internalType": "address", "name":
"onBehalfOf", "type": "address" }, { "indexed": false, "internalType": "uint256", "name": "amount", "type": "uint256" }, {
"indexed": false, "internalType": "enum DataTypes.InterestRateMode", "name": "interestRateMode", "type": "uint8" }, {
"indexed": false, "internalType": "uint256", "name": "borrowRate", "type": "uint256" }, { "indexed": true, "internalType":
"uint16", "name": "referralCode", "type": "uint16" }], "name": "Borrow", "type": "event" }, { "anonymous": false, "inputs": [{
"indexed": true, "internalType": "address", "name": "target", "type": "address" }, { "indexed": false, "internalType": "address",
"name": "initiator", "type": "address" }, { "indexed": true, "internalType": "address", "name": "asset", "type": "address" }, {
"indexed": false, "internalType": "uint256", "name": "amount", "type": "uint256" }, { "indexed": false, "internalType": "enum
DataTypes.InterestRateMode", "name": "interestRateMode", "type": "uint8" }, { "indexed": false, "internalType": "uint256",
"name": "premium", "type": "uint256" }, { "indexed": true, "internalType": "uint16", "name": "referralCode", "type": "uint16" }],
"name": "FlashLoan", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name":
"asset", "type": "address" }, { "indexed": false, "internalType": "uint256", "name": "totalDebt", "type": "uint256" }], "name":
"IsolationModeTotalDebtUpdated", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType":
"address", "name": "collateralAsset", "type": "address" }, { "indexed": true, "internalType": "address", "name": "debtAsset",
"type": "address" }, { "indexed": true, "internalType": "address", "name": "user", "type": "address" }, { "indexed": false,
"internalType": "uint256", "name": "debtToCover", "type": "uint256" }, { "indexed": false, "internalType": "uint256", "name":
"liquidatedCollateralAmount", "type": "uint256" }, { "indexed": false, "internalType": "address", "name": "liquidator", "type":
"address" }, { "indexed": false, "internalType": "bool", "name": "receiveAToken", "type": "bool" }], "name": "LiquidationCall",
"type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name": "reserve", "type":
"address" }, { "indexed": false, "internalType": "address", "name": "user", "type": "address" }, { "indexed": true, "internalType":
"address", "name": "onBehalfOf", "type": "address" }, { "indexed": false, "internalType": "uint256", "name": "amount", "type":
"uint256" }, { "indexed": true, "internalType": "uint16", "name": "referralCode", "type": "uint16" }], "name": "MintUnbacked",
"type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name": "reserve", "type":
"address" }, { "indexed": false, "internalType": "uint256", "name": "amountMinted", "type": "uint256" }], "name":
"MintedToTreasury", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name":
"reserve", "type": "address" }, { "indexed": true, "internalType": "address", "name": "user", "type": "address" }], "name":
"RebalanceStableBorrowRate", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address",
"name": "reserve", "type": "address" }, { "indexed": true, "internalType": "address", "name": "user", "type": "address" }, {
"indexed": true, "internalType": "address", "name": "repayer", "type": "address" }, { "indexed": false, "internalType": "uint256",
"name": "amount", "type": "uint256" }, { "indexed": false, "internalType": "bool", "name": "useATokens", "type": "bool" }],
"name": "Repay", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name":
"reserve", "type": "address" }, { "indexed": false, "internalType": "uint256", "name": "liquidityRate", "type": "uint256" }, {
"indexed": false, "internalType": "uint256", "name": "stableBorrowRate", "type": "uint256" }, { "indexed": false, "internalType":
"uint256", "name": "variableBorrowRate", "type": "uint256" }, { "indexed": false, "internalType": "uint256", "name":
"liquidityIndex", "type": "uint256" }, { "indexed": false, "internalType": "uint256", "name": "variableBorrowIndex", "type":
"uint256" }], "name": "ReserveDataUpdated", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true,
"internalType": "address", "name": "reserve", "type": "address" }, { "indexed": true, "internalType": "address", "name": "user",
"type": "address" }], "name": "ReserveUsedAsCollateralDisabled", "type": "event" }, { "anonymous": false, "inputs": [{
"indexed": true, "internalType": "address", "name": "reserve", "type": "address" }, { "indexed": true, "internalType": "address",
"name": "user", "type": "address" }], "name": "ReserveUsedAsCollateralEnabled", "type": "event" }, { "anonymous": false,
"inputs": [{ "indexed": true, "internalType": "address", "name": "reserve", "type": "address" }, { "indexed": false,
"internalType": "address", "name": "user", "type": "address" }, { "indexed": true, "internalType": "address", "name":
"onBehalfOf", "type": "address" }, { "indexed": false, "internalType": "uint256", "name": "amount", "type": "uint256" }, {
"indexed": true, "internalType": "uint16", "name": "referralCode", "type": "uint16" }], "name": "Supply", "type": "event" }, {
"anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name": "reserve", "type": "address" }, { "indexed":
true, "internalType": "address", "name": "user", "type": "address" }, { "indexed": false, "internalType": "enum
DataTypes.InterestRateMode", "name": "interestRateMode", "type": "uint8" }], "name": "SwapBorrowRateMode", "type":
"event" }, { "anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name": "user", "type": "address" }, {
"indexed": false, "internalType": "uint8", "name": "categoryId", "type": "uint8" }], "name": "UserEModeSet", "type": "event" }, {
"anonymous": false, "inputs": [{ "indexed": true, "internalType": "address", "name": "reserve", "type": "address" }, { "indexed":
true, "internalType": "address", "name": "user", "type": "address" }, { "indexed": true, "internalType": "address", "name": "to",
"type": "address" }, { "indexed": false, "internalType": "uint256", "name": "amount", "type": "uint256" }], "name": "Withdraw",
"type": "event" }, { "inputs": [], "name": "ADDRESSES_PROVIDER", "outputs": [{ "internalType": "contract
IPoolAddressesProvider", "name": "", "type": "address" }], "stateMutability": "view", "type": "function" }, { "inputs": [], "name":
"BRIDGE_PROTOCOL_FEE", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }], "stateMutability":
"view", "type": "function" }, { "inputs": [], "name": "FLASHLOAN_PREMIUM_TOTAL", "outputs": [{ "internalType": "uint128",
"name": "", "type": "uint128" }], "stateMutability": "view", "type": "function" }, { "inputs": [], "name":
"FLASHLOAN_PREMIUM_TO_PROTOCOL", "outputs": [{ "internalType": "uint128", "name": "", "type": "uint128" }],
"stateMutability": "view", "type": "function" }, { "inputs": [], "name": "MAX_NUMBER_RESERVES", "outputs": [{
"internalType": "uint16", "name": "", "type": "uint16" }], "stateMutability": "view", "type": "function" }, { "inputs": [], "name":
"MAX_STABLE_RATE_BORROW_SIZE_PERCENT", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }],

"stateMutability": "view", "type": "function" }, { "inputs": [], "name": "POOL_REVISION", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "uint256", "name": "fee", "type": "uint256" }], "name": "backUnbacked", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "uint256", "name": "interestRateMode", "type": "uint256" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" }, { "internalType": "address", "name": "onBehalfOf", "type": "address" }], "name": "borrow", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "uint8", "name": "id", "type": "uint8" }, { "components": [{ "internalType": "uint16", "name": "ltv", "type": "uint16" }, { "internalType": "uint16", "name": "liquidationThreshold", "type": "uint16" }, { "internalType": "uint16", "name": "liquidationBonus", "type": "uint16" }, { "internalType": "address", "name": "priceSource", "type": "address" }, { "internalType": "string", "name": "label", "type": "string" }], "internalType": "struct DataTypes.EModeCategory", "name": "category", "type": "tuple" }], "name": "configureEModeCategory", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "address", "name": "onBehalfOf", "type": "address" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" }], "name": "deposit", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }], "name": "dropReserve", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }, { "internalType": "address", "name": "from", "type": "address" }, { "internalType": "address", "name": "to", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "uint256", "name": "balanceFromBefore", "type": "uint256" }, { "internalType": "uint256", "name": "balanceToBefore", "type": "uint256" }], "name": "finalizeTransfer", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "receiverAddress", "type": "address" }, { "internalType": "address[]", "name": "assets", "type": "address[]" }, { "internalType": "uint256[]", "name": "amounts", "type": "uint256[]" }, { "internalType": "uint256[]", "name": "interestRateModes", "type": "uint256[]" }, { "internalType": "address", "name": "onBehalfOf", "type": "address" }, { "internalType": "bytes", "name": "params", "type": "bytes" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" }], "name": "flashLoan", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "receiverAddress", "type": "address" }, { "internalType": "address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "bytes", "name": "params", "type": "bytes" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" }], "name": "flashLoanSimple", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }], "name": "getConfiguration", "outputs": [{ "components": [{ "internalType": "uint256", "name": "data", "type": "uint256" }], "internalType": "struct DataTypes.ReserveConfigurationMap", "name": "", "type": "tuple" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "uint8", "name": "id", "type": "uint8" }], "name": "getEModeCategoryData", "outputs": [{ "components": [{ "internalType": "uint16", "name": "ltv", "type": "uint16" }, { "internalType": "uint16", "name": "liquidationThreshold", "type": "uint16" }, { "internalType": "uint16", "name": "liquidationBonus", "type": "uint16" }, { "internalType": "address", "name": "priceSource", "type": "address" }, { "internalType": "string", "name": "label", "type": "string" }], "internalType": "struct DataTypes.EModeCategory", "name": "", "type": "tuple" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "uint16", "name": "id", "type": "uint16" }], "name": "getReserveAddressById", "outputs": [{ "internalType": "address", "name": "", "type": "address" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }], "name": "getReserveData", "outputs": [{ "components": [{ "components": [{ "internalType": "uint256", "name": "data", "type": "uint256" }], "internalType": "struct DataTypes.ReserveConfigurationMap", "name": "configuration", "type": "tuple" }, { "internalType": "uint128", "name": "liquidityIndex", "type": "uint128" }, { "internalType": "uint128", "name": "currentLiquidityRate", "type": "uint128" }, { "internalType": "uint128", "name": "variableBorrowIndex", "type": "uint128" }, { "internalType": "uint128", "name": "currentVariableBorrowRate", "type": "uint128" }, { "internalType": "uint128", "name": "currentStableBorrowRate", "type": "uint128" }, { "internalType": "uint40", "name": "lastUpdateTimestamp", "type": "uint40" }, { "internalType": "uint16", "name": "id", "type": "uint16" }, { "internalType": "address", "name": "aTokenAddress", "type": "address" }, { "internalType": "address", "name": "stableDebtTokenAddress", "type": "address" }, { "internalType": "address", "name": "variableDebtTokenAddress", "type": "address" }, { "internalType": "address", "name": "interestRateStrategyAddress", "type": "address" }, { "internalType": "uint128", "name": "accruedToTreasury", "type": "uint128" }, { "internalType": "uint128", "name": "unbacked", "type": "uint128" }, { "internalType": "uint128", "name": "isolationModeTotalDebt", "type": "uint128" }], "internalType": "struct DataTypes.ReserveData", "name": "", "type": "tuple" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }], "name": "getReserveNormalizedIncome", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }], "name": "getReserveNormalizedVariableDebt", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }], "stateMutability": "view", "type": "function" }, { "inputs": [], "name": "getReservesList", "outputs": [{ "internalType": "address[]", "name": "", "type": "address[]" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "user", "type": "address" }], "name": "getUserAccountData", "outputs": [{ "internalType": "uint256", "name": "totalCollateralBase", "type": "uint256" }, { "internalType": "uint256", "name": "totalDebtBase", "type": "uint256" }, { "internalType": "uint256", "name": "availableBorrowsBase", "type": "uint256" }, { "internalType": "uint256", "name": "currentLiquidationThreshold", "type": "uint256" }, { "internalType": "uint256", "name": "ltv", "type": "uint256" }, { "internalType": "uint256", "name": "healthFactor", "type": "uint256" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "user", "type": "address" }], "name": "getUserConfiguration", "outputs": [{ "components": [{ "internalType": "uint256", "name": "data", "type": "uint256" }], "internalType": "struct DataTypes.UserConfigurationMap", "name": "", "type": "tuple" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "user", "type": "address" }], "name": "getUserEMode", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "address", "name": "asset", "type": "address" }, { "internalType": "address", "name": "aTokenAddress", "type": "address" }, {

```
"internalType": "address", "name": "stableDebtAddress", "type": "address" }, { "internalType": "address", "name":
"variableDebtAddress", "type": "address" }, { "internalType": "address", "name": "interestRateStrategyAddress", "type":
"address" } ], "name": "initReserve", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ {
"internalType": "contract IPoolAddressesProvider", "name": "provider", "type": "address" } ], "name": "initialize", "outputs": [],
"stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "collateralAsset", "type":
"address" }, { "internalType": "address", "name": "debtAsset", "type": "address" }, { "internalType": "address", "name": "user",
"type": "address" }, { "internalType": "uint256", "name": "debtToCover", "type": "uint256" }, { "internalType": "bool", "name":
"receiveAToken", "type": "bool" } ], "name": "liquidationCall", "outputs": [], "stateMutability": "nonpayable", "type": "function" },
{ "inputs": [ { "internalType": "address[]", "name": "assets", "type": "address[]" } ], "name": "mintToTreasury", "outputs": [],
"stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address"
}, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "address", "name": "onBehalfOf", "type":
"address" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" } ], "name": "mintUnbacked", "outputs": [],
"stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address"
}, { "internalType": "address", "name": "user", "type": "address" } ], "name": "rebalanceStableBorrowRate", "outputs": [],
"stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address"
}, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "uint256", "name": "interestRateMode",
"type": "uint256" }, { "internalType": "address", "name": "onBehalfOf", "type": "address" } ], "name": "repay", "outputs": [ {
"internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ {
"internalType": "address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256"
}, { "internalType": "uint256", "name": "interestRateMode", "type": "uint256" } ], "name": "repayWithATokens", "outputs": [ {
"internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ {
"internalType": "address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256"
}, { "internalType": "uint256", "name": "interestRateMode", "type": "uint256" }, { "internalType": "address", "name":
"onBehalfOf", "type": "address" }, { "internalType": "uint256", "name": "deadline", "type": "uint256" }, { "internalType": "uint8",
"name": "permitV", "type": "uint8" }, { "internalType": "bytes32", "name": "permitR", "type": "bytes32" }, { "internalType":
"bytes32", "name": "permitS", "type": "bytes32" } ], "name": "repayWithPermit", "outputs": [ { "internalType": "uint256",
"name": "", "type": "uint256" } ], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address",
"name": "token", "type": "address" }, { "internalType": "address", "name": "to", "type": "address" }, { "internalType": "uint256",
"name": "amount", "type": "uint256" } ], "name": "rescueTokens", "outputs": [], "stateMutability": "nonpayable", "type":
"function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address" } ], "name":
"resetIsolationModeTotalDebt", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType":
"address", "name": "asset", "type": "address" }, { "components": [ { "internalType": "uint256", "name": "data", "type": "uint256"
} ], "internalType": "struct DataTypes.ReserveConfigurationMap", "name": "configuration", "type": "tuple" } ], "name":
"setConfiguration", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address",
"name": "asset", "type": "address" }, { "internalType": "address", "name": "rateStrategyAddress", "type": "address" } ], "name":
"setReserveInterestRateStrategyAddress", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ {
"internalType": "uint8", "name": "categoryId", "type": "uint8" } ], "name": "setUserEMode", "outputs": [], "stateMutability":
"nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address" }, {
"internalType": "bool", "name": "useAsCollateral", "type": "bool" } ], "name": "setUserUseReserveAsCollateral", "outputs": [],
"stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address"
}, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "address", "name": "onBehalfOf", "type":
"address" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" } ], "name": "supply", "outputs": [],
"stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address", "name": "asset", "type": "address"
}, { "internalType": "uint256", "name": "amount", "type": "uint256" }, { "internalType": "address", "name": "onBehalfOf", "type":
"address" }, { "internalType": "uint16", "name": "referralCode", "type": "uint16" }, { "internalType": "uint256", "name":
"deadline", "type": "uint256" }, { "internalType": "uint8", "name": "permitV", "type": "uint8" }, { "internalType": "bytes32",
"name": "permitR", "type": "bytes32" }, { "internalType": "bytes32", "name": "permitS", "type": "bytes32" } ], "name":
"supplyWithPermit", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType": "address",
"name": "asset", "type": "address" }, { "internalType": "uint256", "name": "interestRateMode", "type": "uint256" } ], "name":
"swapBorrowRateMode", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType":
"uint256", "name": "protocolFee", "type": "uint256" } ], "name": "updateBridgeProtocolFee", "outputs": [], "stateMutability":
"nonpayable", "type": "function" }, { "inputs": [ { "internalType": "uint128", "name": "flashLoanPremiumTotal", "type": "uint128"
}, { "internalType": "uint128", "name": "flashLoanPremiumToProtocol", "type": "uint128" } ], "name":
"updateFlashloanPremiums", "outputs": [], "stateMutability": "nonpayable", "type": "function" }, { "inputs": [ { "internalType":
"address", "name": "asset", "type": "address" }, { "internalType": "uint256", "name": "amount", "type": "uint256" }, {
"internalType": "address", "name": "to", "type": "address" } ], "name": "withdraw", "outputs": [ { "internalType": "uint256",
"name": "", "type": "uint256" } ], "stateMutability": "nonpayable", "type": "function" } ]
```

Previous PriceOracleSentinel Next PoolConfigurator Last updated6 months ago On this page *Pool * Write Methods * supply * supplyWithPermit * withdraw * borrow * repay * repayWithPermit * repayWithATokens * swapBorrowRateMode * rebalanceStableBorrowRate * setUserUseReserveAsCollateral * liquidationCall * flashLoan * flashLoanSimple * mintToTreasury * setUserEMode * mintUnbacked * backUnbacked * rescueTokens * View Methods * getReserveData * getUserAccountData * getConfiguration * getUserConfiguration * getReserveNormalizedIncome * getReserveNormalizedDebt * getReservesList * getEModeCategoryData * getUserEMode * FLASHLOAN_PREMIUM_TOTAL * FLASHLOAN_PREMIUM_TO_PROTOCOL * ABI

Was this helpful?