AToken

AToken

aTokens are tokens minted and burnt uponsupply andwithdraw of assets to an Aave market, which denote the amount of crypto assets supplied and the yield earned on those assets. The aTokens' value is pegged to the value of the corresponding supplied asset at a 1:1 ratio and can be safely stored, transferred or traded. All yield collected by the aTokens' reserves are distributed to aToken holders directly by continuously increasing their wallet balance.

EIP20 Methods

All standard EIP20 methods are implemented for aTokens, such asbalanceOf ,transfer ,transferFrom ,approve ,totalSupply etc.

balanceOf will always return the most up to date balance of the user, which includes their principal balance + the yield generated by the principal balance.

EIP712 Methods

DOMAIN_SEPARATOR

function DOMAIN_SEPARATOR()

Get the domain separator for the token at current chain.

nonces

function nonces(address owner)

Returns the nonce value for address specified as parameter. This is the nonce used when callingpermit()

```

Copy consttoken=newContract(aTokenAddress,aToken.abi,provider); awaittoken.nonces(user);

```

Aave Protocol View Methods

scaledBalanceOf

function scaledBalanceOf(address user)

Returns the scaled supply balance ofuser . The scaled balance is the sum of all the updated stored balance divided by the reserve's liquidity index at the moment of the update.

getScaledUserBalanceAndSupply

function getScaledUserBalanceAndSupply(address user)

Returns the scaled balance of the user and the scaled total supply.

scaledTotalSupply

function scaledTotalSupply()

Returns the scaled total supply of theaToken .

getPreviousIndex

function getPreviousIndex(address user)

Returns last index interest that was accrued to the user's balance (expressed in ray).

getIncentivesController

function getIncentivesController()

Returns the address of the Incentives Controller contract

POOL

function POOL()

Returns the address of the associated Pool for theaToken.

UNDERLYING_ASSET_ADDRESS

function UNDERLYING_ASSET_ADDRESS()

Returns address of the underlying reserve asset.

RESERVE_TREASURY_ADDRESS

function RESERVE_TREASURY_ADDRESS()

Returns address of the Aave Treasury, controlled by governance, receiving the fee on thisaToken.

Aave Protocol Write Methods

setIncentivesController

function setIncentivesController(IAaveIncentivesController controller)

Sets a new Incentives Controller.

Only Pool Admin can call this methods. To update Incentives Controller on main Aave market, Governance Proposal must be submitted.

permit

Allows a user to permit another account (or contract) to use their funds using a signed message. This enables gas-less transactions and single approval/transfer transactions.

Parameter Type Description owner address The owner of the funds spender address The spender for the funds value uint256 The amount the spender is permitted to use deadline uint256 The deadline timestamp that the permit is valid. Use type(uint).max for no deadline. v uint8 Signature parameter r bytes32 Signature parameter s bytes32 Signature parameter ```

Copy import{ signTypedData_v4 }from'eth-sig-util' import{ fromRpcSig }from'ethereumjs-util' // ... other imports importaTokenAbifrom"./aTokenAbi.json" // ... setup your web3 provider constaTokenAddress="ATOKEN_ADDRESS" constaTokenContract=newweb3.eth.Contract(aTokenAbi,aTokenAddress) constprivateKey="YOUR_PRIVATE_KEY_WITHOUT_0x" constchainId=1 constowner="OWNER_ADDRESS" constspender="SPENDER_ADDRESS" constvalue=100// Amount the spender is permitted constnonce=1// The next valid nonce, use _nonces() constdeadline=1600093162 constpermitParams={ types:{ EIP712Domain:[{ name:"name",type:"string"}, { name:"version",type:"string"}, { name:"chainId",type:"uint256"}, { name:"verifyingContract",type:"address"},], Permit:[{ name:"owner",type:"address"}, { name:"spender",type:"address"}, { name:"value",type:"uint256"}, { name:"nonce",type:"uint256"}, { name:"deadline",type:"uint256"},], }, primaryType:"Permit", domain:{ name:"aTOKEN_NAME", version:"1", chainId:chainId, verifyingContract:aTokenAddress, }, message:{ owner, spender, value, nonce, deadline, }, } constsignature=signTypedData_v4(Buffer.from(privateKey,"hex"), { data:permitParams }) // The signature can now be used to execute the transaction const{v,r,s}=fromRpcSig(signature) awaitaTokenContract.methods .permit({ owner, spender, value, deadline, v, r, s }) .send() .catch((e)=>{ throwError(Error permitting:{e.message}) })

```

FAQ

- How aToken earn interest? / How aToken balance increases?
- LendingPool
- methods (deposit, withdraw, borrow, repay, liquidationCall) updates the state and cumulated liquidity index of the reserve once every block. AToken'sbalanceOf
- method returns the balance computed based onblock.timestamp
- andliquidityIndex
- of the underlying reserve and hence, returns the most up to date balance of account, which includesprincipal + interest.
- LiquidityRate vs LiquidityIndex
- Can I transfer aTokens?
- Yes! with few caveat to keep in mind
-
    - By transferring aTokens, you’re transferring your balance of the underlying asset. Only the account holding the aTokens canwithdraw
-
    - the deposited asset.
-
```

    - AToken transfer will fail if the resulting Health Factor ofuser
-
    - will end up being below 1.
- *
- If I transfer aToken does my pending liquidity rewards get transferred?
- No, liquidity rewards earned prior to the transfer of aToken are accrued by the user/address holding the aTokens originally. Though, all future liquidity rewards will be earned by the new recipient.
- What is the difference between ScaledBalance and Balance?
- Example please!
- Let’s say you supply 1,000 DAI to the AaveLendingPool
- , you will receive 1,000 aDAI (at 1:1 exchange rate).
- You can see your aDAI balance increasing right away.
- Now, say a month later your aDAI balance is 1,050. You could withdraw 1,050 DAI from LendingPool by burning 1050 aDAI.
-

Previous WETHGateway Next DebtToken Last updated6 months ago On this page *AToken * EIP20 Methods * EIP712 Methods * DOMAIN_SEPARATOR * nonces * Aave Protocol View Methods * scaledBalanceOf * getScaledUserBalanceAndSupply * scaledTotalSupply * getPreviousIndex * getIncentivesController * POOL * UNDERLYING_ASSET_ADDRESS * RESERVE_TREASURY_ADDRESS * Aave Protocol Write Methods * setIncentivesController * permit * FAQ

Was this helpful?