

# Multiple Rewards and Claim

## Multiple Rewards and Claim

In Aave Protocol V2, Aave Governance activated liquidity mining rewards through a community proposal. While emissions of rewards differ per asset, the rewards were all denominated in stAave tokens. Aave Protocol V3 offers option to have multiple rewards per token. Now, it is possible for an asset listing to enable additional incentive rewards denominated in their protocol tokens.

V3 also allows user to claim rewards to another account as well as self and to claim multiple types of rewards per asset in single tx.

The [RewardsController](#) contract is the main contract where the user interacts to claim the rewards of their positions. The users can claim all the rewards or an individual reward per transaction, with a variety of functions that allow more granularity at claim.

## Integration Guide

### Prerequisites

User(s) should already have a position in the protocol. Depending on the market and incentives that are enabled, they should already have a supply, borrow or both in one of the incentivised assets.

### Check rewards enabled

To get list of rewards available for the given asset, use the `getRewardsByAsset()` method, passing in the associated a/s/vToken address of the incentivised asset. Use `getRewardsData()` to get details per reward per asset.

```

```
Copy const daiRewardsList = await rewardsController.getRewardsByAsset(aDaiAddress);
```

```
const daiRewardOneData = await rewardsController.getRewardsData(aDaiAddress, rewardOneAddress);
```

```

### Check User(s) reward balance

You can get user reward balance for a given reward token or all reward tokens for a given list of assets using `getUserRewardsBalance()` or `getAllUserRewardsBalance()` respectively.

```

```
Copy const unclaimedUserRewards = await rewardsController.getUserRewardsBalance([aDai.address, aWeth.address], userAddress, stkAave.address);
```

```
const [, allUnclaimedRewards] = await rewardsController.getAllUserRewardsBalance([aDai.address, aWeth.address], userAddress);
```

```

### Claim Rewards

Claim one or all type of rewards to self, another address or on behalf of an address (only authorised claimers) for a given assets list.

#### Self

User can claim one or all type of rewards using `claimRewardsToSelf()` and `claimAllRewardsToSelf()` respectively for the list of assets passed to the method.

The `msg.sender` must match the user's address that has accrued the rewards. ```

```
Copy // claims only stAave for asset list [aDai, aWETH, ] const claimStkAave = await rewardsController.claimRewardsToSelf([aDai.address, aWETH.address,], amountToClaim, stkAave.address);
```

```
// claims all reward types for asset list [aDai, aWETH, ] const claimAllRewards = await rewardsController.claimAllRewardsToSelf([aDai.address, aWeth.address,]);
```

```

## onBehalfOf An Address

Authorised user/contract addresses can claim one or all type of rewardsonBehalfOf, the user accruing rewards, usingclaimRewardsOnBehalf() andclaimAllRewardsOnBehalf() respectively for the list of assets passed to the method.

Themsg.sender must be an authorised claimer set using setClaimer() method, via Governance Vote. ``

```
Copy // claims only stkAave for asset list [aDai, aWETH, ]
constclaimStkAaveOnBehalfOf=awaitrewardsController.claimRewardsOnBehalf( [aDai.address], MAX_UINT_AMOUNT,
userWithRewards.address, claimer.address, stkAave.address )
```

```
// claims all reward types for asset list [aDai, aWETH, ]
constclaimAllRewardsOnBehalfOf=awaitrewardsController.claimAllRewardsOnBehalf( [aDai.address],
userWithRewards.address, claimer.address )
```

...

## To Another Address

User can claim one or all type of rewards, to an external account (i.e. an account which may or may not be accruing the rewards being claimed), usingclaimRewards() andclaimAllRewards() respectively for the list of assets passed to the method.

Themsg.sender must match the user's address that has accrued the rewards ``

```
Copy // claims only stkAave for asset list [aDai, aWETH, ] to receivingAccount
constclaimStkAave=awaitrewardsController.claimRewards( [aDai.address,aWETH.address,], amountToClaim,
receivingAccount.address stkAave.address );
```

```
// claims all reward types for asset list [aDai, aWETH, ] to receivingAccount
constclaimAllRewards=awaitrewardsController.claimAllRewards( [aDai.address,aWeth.address,], receivingAccount.address
);
```

...

[Previous Repay With aTokens Next Credit Delegation](#) Last updated1 year ago On this page \*[Multiple Rewards and Claim Integration Guide](#) \* [Prerequisites](#) \* [Check rewards enabled](#) \* [Check User\(s\) reward balance](#) \* [Claim Rewards](#)

Was this helpful?