

Contract Secured Revenue (CSR)

Canto implements a fee split model called Contract Secured Revenue, or CSR, by means of the [CSR](#) module. CSR allows you to claim a percentage of all transaction fees paid by users when interacting with your smart contracts.

CSR was added in the Canto v5.0.0 chain upgrade with an initial fee split of 20%, following a successful [governance proposal](#).

Overview

To earn CSR fees, you must register your contracts with the CSRTurnstile smart contract deployed on Canto mainnet at <0xEcf044C5B4b867CFda001101c617eCd347095B44>. The source code for Turnstile can be found [here](#).

Upon registering a contract, a transferrable NFT representing the right to claim that contract's revenue is minted to an address of your choice. Alternatively, you can assign a contract's revenue to an existing CSR NFT.

CSR fees for all registered contracts accrue in the Turnstile contract. To withdraw fees from the Turnstile contract, you must specify the token ID of a CSR NFT you hold.

Registering a Contract

Calling the `register` method from an EOA does not revert and will mint a CSR NFT. However, no fees will be distributed to this NFT when the EOA transacts. Additionally, any contracts later assigned to this NFT will *not* **generate CSR fees**. To register a smart contract for CSR, call the `register` method on the [Turnstile contract](#) from the contract you wish to register.

This method takes one parameter: the address to which the CSR NFT should be minted. This can be an address that does not exist (i.e. an address that has never transacted before).

You can register a contract in the constructor or otherwise; however, CSR fees will only begin accruing once the contract is registered.

...

```
Copy interface Turnstile {
    function register(address) external returns (uint256);
}

contract Example {
    Turnstile turnstile = Turnstile(0xabc...);

    constructor() {
        //Registers the smart contract with Turnstile //Mints the CSR NFT to the contract creator
        turnstile.register(tx.origin);
    }
}
```

...

The `register` method returns the token ID of the CSR NFT that was minted.

Assignment

You can register a smart contract for CSR while assigning its fees to an existing CSR NFT by calling the `assign` method on the [Turnstile contract](#). This method also takes one parameter: the `uint256` token ID of the CSR NFT to which fees should be assigned.

The same constraints apply:

- You must call this method from the contract you wish to register.
- CSR fees will only begin accruing once the contract is registered.
-

...

```
Copy interface Turnstile {
    function assign(uint256) external returns (uint256);
}

contract Example {
    Turnstile turnstile = Turnstile(0xabc...);

    constructor() {
        //Registers the smart contract with Turnstile //Assigns this contract's fee revenue to CSR NFT #1
        turnstile.assign(1);
    }
}
```

...

The `assign` method returns the token ID of the CSR NFT that was passed to it.

Withdrawing Fees

You can check the amount of fees that have accrued for a CSR NFT by reading the value stored in thebalances mapping for the NFT's token ID.

To withdraw CSR fees, call thewithdraw method on the Turnstile contractfrom the address that holds the CSR NFT . This method takes three parameters: the token ID of the CSR NFT, the address that should receive the fees, and the amount of fees to withdraw.

Factory Contracts

You can implement theregister andassign methods in factory contracts, such that child contracts are automatically registered for CSR when they are created.

[Previous NOTE, DEX, and Lending Market](#) [Next Verifying Contracts](#) Last updated24 days ago On this page *[Overview](#) *
[Registering a Contract](#) * [Assignment](#) * [Withdrawing Fees](#) * [Factory Contracts](#)