

Airdrop

[Suggest Edits](#)

The Airdrop template is an audited, ready-to-deploy, airdrop smart contract for ERC-20, ERC-721, ERC-1155, or native tokens. The airdrop is performed by “pushing tokens,” which happens when the contract owner transfers tokens to the receivers’ addresses. Receivers do not need to take any action and do not incur a gas fee.

The following are common use cases for the Airdrop template:

- Bulk token distribution
- : Automatically distribute tokens to users in bulk. For example, project teams or token issuers can reward their community members, early adopters, and participants in specific campaigns by distributing tokens directly to their blockchain addresses.
- Marketing and promotion
- : Increase awareness and engagement with a project or token. When a project distributes tokens to a large number of users, this attracts attention and incentivizes users to explore and participate in the project's ecosystem.
- Community building and engagement
- : Foster a strong and active community around a project. Distribute tokens to community members to incentivize them to stay engaged, participate in discussions, provide feedback, and contribute to the overall growth and success of the project.

We recommend setting a maximum of 500 recipients per airdrop transaction. More than 500 recipients can cause the blockchain fees to spike or the transaction to fail.

Deployment parameters

The Airdrop template creates a customized airdrop smart contract to distribute ERC-20, ERC-721, ERC-1155, or native tokens to multiple users. To create a contract using this template, provide the following parameter values when deploying a smart contract template. To deploy a template, send a POST request to the `/templates/{id}/deploy` endpoint.

Template ID : 13e322f2-18dc-4f57-8eed-4bddfc50f85e

Template deployment parameters

Parameter	Type	Required	Description
defaultAdmin	Add X	Address of the default admin. This address can execute	
		permissible functions on the contract.	

Important: You lose administrative access to the contract if this is not set to an address you control. contractURI

String
URL for the marketplace metadata of your contract.

Common functions

This section lists the most commonly used functions on the Airdrop template, their respective parameters, and potential failure scenarios. These functions include:

- [setOwner \[write\]](#)
- [airdropERC1155 \[write\]](#)
- [airdropERC721 \[write\]](#)
- [airdropERC20 \[write\]](#)
- [owner \[read\]](#)

setOwner [write]

The setOwner function sets the owner of the smart contract or transfers ownership from the existing owner to a new owner.

Parameters

Parameter	Type	Description
_newOwner	address	Address of the new owner.

Failure scenarios

- The
- setOwner
- function fails if it is called by a non-admin.

airdropERC1155 [write]

The airdropERC1155 function allows the owner of this address to send ERC-1155 tokens to a list of recipients.

Parameters

The names of the parameters are not used in the request body. Parameter Type Description _newOwner address Address of the new owner. _contents tuple:

- address
- uint256
- uint256 Array of recipient information:
- Address that is to receive the airdrop
- ID of the token within the ERC-1155 contract to be distributed
- Amount of the token within the ERC-1155 contract to be distributed

Failure scenarios

The airdropERC1155 function fails if:

- It is called by a non-admin
- It contains incorrect token information, such as an invalid token ID
- The Airdrop contract is not approved for what's being transferred, which means it is not authorized to transfer assets on the airdropper's behalf
- Airdropping to a contract without a receive function

Example

The following sample code shows the new owner address and contents when sending a POST request to the /developer/transactions/contractExecution endpoint:

```
JSON "abiParameters": [ "0x2B5AD5c4795c026514f8317c7a215E218DcCD6cF", [
["0x4CCeBa2d7D2B4fdCE4304d3e09a1fea9fbEb1528", 0, 10], ["0xf4e2B0fcbd0DC4b326d8A52B718A7bb43BdBd072", 0,
10], ] ]
```

airdropERC721 [write]

The airdropERC721 function allows the owner of this address to send ERC-20 tokens to a list of recipients.

Parameters

The names of the parameters are not used in the request body. Parameter Type Description _tokenAddress address Address of the new owner. _contents tuple:

- address
- uint256 Array of recipient information:
- Address that is to receive the airdrop
- Amount of the token within the ERC-721 contract to be distributed

Failure scenarios

The airdropERC721 function fails if:

- It is called by a non-admin
- It contains incorrect token information, such as an invalid token ID
- The Airdrop contract is not approved for what's being transferred, which means it is not authorized to transfer assets on the airdropper's behalf
- Airdropping to a contract without a receive function

Example

The following sample code shows the token address and contents when sending a POST request to the /developer/transactions/contractExecution endpoint:

```
JSON "abiParameters": [ "0x2B5AD5c4795c026514f8317c7a215E218DcCD6cF", [
["0x4CCeBa2d7D2B4fdcE4304d3e09a1fea9fbEb1528", 0], ["0xf4e2B0fcbd0DC4b326d8A52B718A7bb43BdBd072", 1], ] ]
```

airdropERC20 [write]

The airdropERC20 function allows the owner of this address to send NFTs to a list of recipients.

Parameters

The names of the parameters are not used in the request body. Parameter Type Description _tokenAddress address Address of the token. _contents tuple:

- address
- uint256 Array of recipient information:
- Address that is to receive the airdrop
- Quantity of tokens to be transferred

Failure scenarios

The airdropERC20 function fails if:

- It is called by a non-admin
- The Airdrop contract is not approved for what's being transferred, which means it is not authorized to transfer assets on the airdropper's behalf
- Airdropping to a contract without a receive function

Example

The following sample code shows the token address and contents when sending a POST request to the /developer/transactions/contractExecution endpoint:

```
JSON "abiParameters": [ "0x2B5AD5c4795c026514f8317c7a215E218DcCD6cF",
"0xd41c057fd1c78805AAC12B0A94a405c0461A6FBb", [ [ "0x4CCeBa2d7D2B4fdcE4304d3e09a1fea9fbEb1528", 100 ] ] ]
```

owner [read]

The owner function retrieves the address of the current owner.

Example

The following sample code shows the owner function when sending a POST request to the /contracts/query endpoint:

```
JSON "abiFunctionSignature": "owner()" Updated 7 days ago * Table of Contents * * Deployment parameters * * Common functions * * * setOwner \[write\] * * * airdropERC1155 \[write\] * * * airdropERC721 \[write\] * * * airdropERC20 \[write\] * * * owner \[read\]
```