

Mint API

The easiest way to mint compressed NFTs at scale.

Overview

The Mint API enables you mint compressed NFTs without the additional headaches. We achieve best-in-class performance by running the API at the edge, alongside our geo-distributed RPC network. Here's a quick overview of the benefits:

1. No need to pay Solana fees. It's on us!
2. No need to create a Merkle tree. We manage them on your behalf, saving you both time and money.
3. We ensure the transaction was submitted successfully and confirmed by the network. No need to poll yourself.
4. We parse the asset ID from the transaction on your behalf, allowing you to immediately use the [DAS API](#).
5. without any extra friction.
6. We upload your off-chain metadata (traits, image, etc) to Arweave. Again, it's on us!
7. Want to take it for a spin? You can find illustrative examples [here](#).

Pricing

100 credits per NFT mint. That's it!

Core Concepts

Merkle Tree Management

Compressed NFTs must be minted to a Merkle tree account. The Merkle tree is a regular Solana account that must be initialized ahead of time with specific configuration values. Developers need to keep track of their Merkle tree accounts and decide which NFTs they want to mint to which tree. At the end of the day, Merkle trees are nothing more than storage layer and have zero impact on your consumer applications. Developers shouldn't have to worry about this! Instead, Helius manages the Merkle trees on your behalf. We ensure there is always a tree available to store your NFT. We cover the cost of creating the Merkle trees. For advanced users: Our trees by default have a tree depth and canopy of 20 and 12, respectively. This means that the minimum proof path length is 8 nodes. If you need a smaller proof path size for composability, please reach out to us on Discord. Want to learn more about Merkle trees? Read our deep dive [here](#).

Collection Delegation

Helius needs permission to mint the NFT into your collection via the collection authority. This process is called "delegation". The authority must be delegated to the Helius minting account:

- * Devnet: `2LbAtCJSaHqTnP9M5QSjvAMXk79RNLusFspFN5Ew67TC`
- * Mainnet: `HnT5KVAywGgQDh6Usk4bxRg4RwKxCK4jmECyaDth5R`

You can learn more about collection delegation [here](#). The Helius SDK allows you to delegate/revoke the collection authority. Learn more [here](#). With Helius as the delegated authority, anyone can mint NFTs to your collection before you revoke this authority. We advise creators to verify their collections for any unauthorized NFTs after revoking the collection authority.

Creator Verification

Unfortunately, Metaplex does not support delegation for creator verification. This means that creators cannot be verified during the mint. However, since you own the NFT you will be able to verify the creator yourself afterwards. If delegated creator verification is important to you, please let us know on Discord or Telegram.

Arweave Uploads

You can choose to provide your own off-chain metadata via the uri field. If no uri is specified, we will build a JSON file and upload to Arweave on your behalf. The file will adhere to the [v1.0 Metaplex JSON standard](#). We upload to Arweave via [Irys](#) (formerly known as Bundlr). The API does not support image uploading at this time. Image uploads are handled differently than standard API requests. The Helius SDK offers image support. Learn more [here](#).

Update Metadata

Users can use the Update instruction to update a compressed NFT's metadata. This must be done by the user, and not via the Mint API. An example of how to update a compressed NFT's metadata can be found [here](#). Compressed NFTs with a verified collection can have the collection authority update metadata, whereas compressed NFTs without a collection cannot update the metadata. The latter is true because Helius has tree authority and manages the compressed NFT. The following fields can be updated: * Name * Symbol * Uri * Creators * SellerFeeBasisPoints * PrimarySaleHappened * IsMutable The IsMutable field can only change from mutable to immutable.

