

Selene Network User Manual

The Selene Network core code is designed to work well when browsing NFTs from either a PC or mobile device. In either case, the URL used to browse the site allows the user to override some of the default settings.

For instance, if a Selene Network compatible NFT smart contract has been launched to the blockchain but the project owner does not have the core code installed, they can view their contract on any Selene Network server by providing the contract address.

The following sections show the most useful URL parameters and describe them through example. If you use multiple options at the same time, separate them with the ampersand ('&') symbol.

Main Level Selene Network Overrides

?contract=

If you know the contract address of a Selene Network compatible smart contract, you can add it to the URL and any server will be able to decode its metadata.

This is useful when linking to a specific project on social media. For instance, you might see this used when adding a link to the description of a video where the project was discussed.

Example

<https://amorstyle.com/dsn/?contract=0xd49c2e6d97722B48Ddd3A26cA0F9D6f6B6f7FDEF>

?agent=

When the Selene Network core code serves a page to the viewer, it displays the agent id that will be credited for the mint in the footer of the page. By default, the agent id is set by the server install.

The agent URL parameter allows the caller to override the default agent.

This functionality should be thoroughly understood by both Founder and Partner Agents in the network. When Partner Agents drive traffic to any Selene Network enabled server, they will need to provide this option in order to be credited for the mint.

Note that the normal browsing functionality that a visitor might do may clear this parameter off the URL. Thus, combining it with the specific contract may help reduce this disconnect.

Ultimately, if a Partner Agent doesn't want to take this commission loss risk, they can always install the core code on their own server and drive the customer traffic to their own site. In which case, they will have set the default Partner Agent id for all minting thus not needing the agent id on the URL.

Example

<https://amorstyle.com/dsn/?agent=5000>

?gallery=

When the Selene Network is loaded, it reads from a ‘gallery’ smart contract that is stored in the server’s configuration file. If a user wants to override the default gallery, they can provide the address on the URL using the ‘gallery’ parameter.

This functionality can be used by both the website designer and agents alike. A website designer may want to have a default gallery that shows the latest offerings from any creator on the network. Or, maybe, the website designer could setup a gallery for each creator they work with. Thus, the creator has a gallery that is a collection of just their NFTs.

Similarly, if a creator has their own gallery and they get interviewed by some social media influencer, that influencer could link to their gallery rather than a single project. In this case, they may also add their agent id.

Example

<https://amorstyle.com/dsn/?gallery=0x9d039e337e7edf45cef6772cb9eaf702cc169069>

?index=

The index field is used so the visitor can provide the index into the gallery that is being requested. The default functionality for the Selene Network is to show the most recent recorded index (highest number). This number is most commonly used with the gallery URL shown above.

Example

<https://amorstyle.com/dsn/?gallery=0x9d039e337e7edf45cef6772cb9eaf702cc169069&index=5>

?display=mobile

There are two supported page layouts, mobile and PC. The standard horizontal layout is assumed. When the code detects that it’s running on a mobile device, it sets the ‘mobile’ keyword the output is formatted narrow.

This functionality is useful when developing custom NFT code.

Example

<https://amorstyle.com/dsn/?display=mobile>

Wallet specific functionality

?account=

The default display for the wallet functionality is to display the ‘limit’ amount of NFTs in all the accounts that are connected to the website. To override this, the visitor can provide a specific wallet address to be viewed by using the ‘account’ parameter.

This override, combined with ‘limit’ (below). can be very useful for providing a clickable NFT resource for a friend. This allows you to share your wallet to someone looking for a specific project link – just click the NFT in my wallet and it will take you to the mint location for that project.

Example

<https://amorstyle.com/dsn/wallet/?account=0xa043F049e084c0a66bcEcdB2Dc0a31Aa93EF7c37>

?limit=

When using the NFT browsing functionality provided by the built in wallet, the Selene Network configuration file allows the server to set any number of NFTs that can be viewed by default. The limit parameter allows the caller to override the default.

This override can be useful when you share your wallet with a friend that wants a link to a particular Selene Network compatible NFT project.

Example

<https://amorstyle.com/dsn/wallet/?limit=5>

Video Specific Functionality

?id=

The Selene Network has a built-in page for viewing Theta Video API DRM protected videos. If you know the video_id, this option allows you to trigger the token-gating process for that video.

Example

https://amorstyle.com/dsn/video/?id=video_8yyib7gg8aldke8m06kxg440c

Reel Specific Functionality

?reel=

The core code also supports parsing a JSON file that represents a collection of videos. A reel can be thought of as a playlist that can be made up of videos either on YouTube or the Theta EdgeCloud Video API. Each video has a name, image and description which

get build into the playlist. Every video also either includes a youtube video id or a Theta EdgeCloud Video Id. (youtube or video_id.)

The reel parameter is a fully qualified URL that is expected to list in the same place as the project.

Note that reel functionality requires a little bit more. The following project is a working model.

Example

<https://amorstyle.com/dsn/?contract=0x1D1aE042083adCDE9D417b38B3b0eE0a9F68f50B>

Developer API functionality

The main Selene Network code also provides some developer API functionality that allows any website to extract information about a particular install with regards to it's gallery and offered products.

Because the Selene Network uses the blockchain as the database and it validates the integrity of the data using the registered hash, these endpoints allow the callers to not have to deal with the web3 coding, but simply extract the data as JSON files.

The Gallery Endpoint

This endpoint returns a JSON file that represents the contents of the gallery file. The Selene Network interacts with the gallery smart contract in order to get the listed products and validate that the registered gallery file is unaltered (using the hash).

Note that when the code enumerates the gallery smart contract for registered files, if the code finds multiple entries with the same 'name', the entries will be assumed to be updates to the same file. For instance, the gallery owner updates the gallery to include new products and then writes the hash to the blockchain.

Older 'overwritten' versions are ignored and galleries are displayed from the most recent to the oldest.

Also, it's worth noting that the URLs found in the gallery will generally reference the Selene Network install location.

Example

<https://amorstyle.com/dsn/api/gallery/>

The Project Endpoint

The project endpoint provides information about an individual Selene Network compatible smart contract as a JSON file.

Logically, the website looking for product information would perform the gallery query first which gives them the list of contracts and then present those contract addresses to this endpoint in order to get the raw information about that project.

It's worth noting that the information returned with the project endpoint may point to servers that are not the main gallery install location. The reason for this is that anyone can offer projects from anyone else in their gallery. They are just Selene Network compatible smart contracts.

Example

<https://amorstyle.com/dsn/api/project/?project=0x161e88623e775e814aC269b0B9dF5EA943a36360>