

DRM Audio Smart Contract

This document outlines how to use the DRM Audio Smart Contract as demonstrated with the `drmaudiov1.zip` project.

Prerequisites

Read the other two documents included within this sample: `readme_hosting.pdf` and `readme_smart_contracts.pdf`.

Requirements

Webhosting service

Many people use really simple templates to build and maintain their website which do not include webhosting for their domain. In order to serve projects to the Selene Network, you'll need to have the ability to manage the files that are mapped to by your domain name. In other words, you need to be able to create folders and upload files to your server. This is generally a function of webhosting via hosting services.

Selene Network install

Because this demo involves the use of custom code which includes the core Selene Network software, the server hosting the project will also need to have a version of the Selene Network software installed.

Fortunately, it's relatively easy to download and install the latest version of the Selene Network from <https://amorstyle.com/download> as either `dsn.zip` or the latest versioned file. Inside that ZIP file you'll find a `docs` folder with the install documentation.

Text editor for modifying files

You will need to modify JSON files and a few PHP files.

Background understanding

This project is built to use the 'custom code' functionality of the Selene Network core code. The custom code actually loads the Selene Network software in order to give the look and feel of the Selene Network while also giving the project it's own location to display the custom functionality that the visitor will experience.

The overall project is hosted on your server and it loads the Selene Network from your own server.

Because this example requires the validation of ownership in order to get the content, the project creator will need to provide a little information in the authentication file along with the project file that describes the contract address and content location on the server. This will be detailed below.

Three custom files

At the root level of this project there are three short files that need to be modified slightly in order to present correctly to the webpage visitor.

- `drmaudiov1.php`
- `drmaudiov1.js`
- `drmaudiov1auth.php`

The 'drmaudiov1' in each of these files should be renamed to match your project.

The basic purpose of each file is outlined here.

`drmaudiov1.php`

This is considered the 'home' file for the project. When someone clicks on the Selene Network UI's home button for the project, this is the file that gets referenced.

The top of the file loads the main Selene Network core software, at the bottom of the file it registers itself as a custom NFT to the `Init()` function.

In the `data()` function, `gName` should be set to your project name rather than 'drmaudiov1.'

In the same function, the `customContract` needs to be set to the address of the version of the `drmaudiov1` smart contract that you deploy to the blockchain. Your smart contract address.

The `customTitle` is going to be used to name the page.

`drmaudiov1.js`

When the home page (`drmaudiov1.php`) loads, the Selene Network will load the JavaScript referenced in the 'javascriptName' data item. The main routine in that file is run after all the php code has run.

The job of the `drmaudiov1.js` file is to enable the 'signature download' button(s) and handle a click on any of those buttons.

In the handling of the click on the download button, the javascript code will call Metamask to sign a message that it in turn sends to the `drmaudiov1auth.php` file on the server.

If the signature is decoded successfully, the JavaScript file handles receiving the downloaded file in the browser.

`drmaudiov1auth.php`

This is the file that handles validating the signature is valid. To do this, the message that was signed needs to decode to the smart contract address of your project smart contract. Thus, make sure that the 'DRMContract' element in the `data()` function matches your smart contract.

Also, the 'root' path is the file location where you place the private data on your server. Because the calling code will provide a 'loc' and 'filename' the content that is stored for download on the server should be placed in a subdirectory and have the given name.

In the case of this sample, the 9 m4a files are placed in a 'audio_demo' folder and are each named 'audio_demo_X.m4a' where X is a number between 1 and 9.

Note

It is recommended that you create a directory on the server of the name 'private' and set the attributes so only you can read and write to the file. When you do this, no one else should be able to get to the content other than through the use of the PHP code that downloads it.

Last part is in the run_the_code() function. If you are authoring something other than a 'audio/m4a' file, you need to modify this to specify the type of audio file you're be downloading. If you change this, you must also modify the 'Accept' value in the JavaScript code to reflect the same type.

Editing the project JSON file

The JSON for the sample project looks like this:

```
✓ object {12}
  name : "DRM Audio Smart Contract"
  description : "Smart contract that is used by the Selene Network to tokengate audio files."
  external_url : "https://amorstyle.com/drmaudiov1/"
  whitepaper : "https://amorstyle.com/nfts/drmaudiov1/whitepaper.pdf"
  image : "https://amorstyle.com/nfts/drmaudiov1/images/drmaudiov1.jpg"
  artist : "Amor's AI helper"
  creator : "AmorStyle"
  collection : "DSN compatible NFT"
  home : "drmaudiov1"
  type : "music"
✓ audio {3}
  title : "Under the Full Moon"
  artist : "Joe Spangler and the Ragtips"
✓ tracks [9]
  ✓ 0 {6}
    title : "This is track one"
    artist : "Joe Spangler and Ragtips"
    file : "audio_demo_1.m4a"
    loc : "audio_demo"
    image : "images/drmaudiov1.jpg"
    track : 1
  ✓ 1 {6}
    title : "This is track two"
    artist : "Joe Spangler and Ragtips"
    file : "audio_demo_2.m4a"
    loc : "audio_demo"
```

Note that as you edit this, all the URL paths should point to your server and the appropriate fields should be created to the correct sources.

There are two sections that are worth noting: type and audio.

Type

The type field is used by the Selene Network core code to provide an overlay on purchased tokens. If this is set to 'reel', the code will overlay a film strip with a reel. If it's set to 'video', the high level code will display a film strip. If it's set to 'music', as in this example, the code will display musical notes.

When the overlay image is clicked, the user is hyperlinked to the server location where the project is hosted so the 'prize' can be downloaded.

Note that this is important if some other Selene Network install mints from your smart contract. In that case, the other server doesn't hold the prize which can only be downloaded from the 'home' location – your server.

And, the type value is used in both the project file and the token metadata file. When the Selene Network reads the token metadata file (located in yourproject\metadata\yourproject.json) when the visitor is on the wallet page, that token will be clickable to the home location too.

Special note: You can share an account using the wallet functionality and if the token in metadata uses the 'type' field as music, the wallet will provide a clickable link back to the project home location. For example:

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

audio

The audio field is the location that holds the information that the drmaudiov1.php displays as the custom information to the visitor. If you add or remove entries, you should also edit the php file that renders this information to the visitor.

In this section you'll find the collection name and artist at the top. Then, the 'tracks' section explicitly lists each track that is offered. It's worth noting that you can create unique images for each track.

The loc and file items are used when downloading the files, thus the correct location and filenames should be used.

The track number is used so that when each button is pressed, the visitor gets the correct index.

hashTable

Lower in the project JSON file you'll find the hashTable. Each file that is downloadable should have it's SHA1 hash recorded along with it's name.

Here is an example of the drmaudiov1.json data:

```

      value : "08140473700591e7487c377e042308e011004e0"
    }
  },
  private : [9]
    {
      0 : {2}
        {
          name : "audio_demo_1.m4a"
          value : "1a3a62eed7bfb6fc53757663a9e6c76b3e556623"
        }
      1 : {2}
        {
          name : "audio_demo_2.m4a"
          value : "e6ed17899c0213fa08eb1d0080a0a66c1e7dfe9d"
        }
      2 : {2}
        {
          name : "audio_demo_3.m4a"
          value : "d07eb01755ae522a0d64f19d5632d23cf604ab22"
        }
      3 : {2}

```

Configuring your smart contract

It is assumed that you have already read the other two documents included in this same that outline the hosting process and the smart contract activation.

Editing

When editing the project JSON file, you'll want to get the assistance from the Selene Network tools with your install. For example, you can view your file like this:

<https://amorstyle.com/dsn/tools/?project=https://amorstyle.com/nfts/drmaudiov1/drmaudiov1.json>

And once you have something that you think will work, you can register it on your project smart contract as instructed in the smart contract activation document. The information used is the URL, Name and SHA1 at the top of the tools page.

Testing

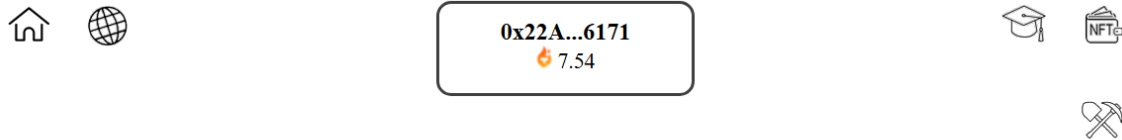
You don't know if what you created is going to work until you register it the first time on the smart contract. Once you do, you can view the contract two different ways.

Using URL for standard layout

If your project is displayed in it's generic capacity on any Selene Network install, you can see what it will look like using a version of the following URL:

<https://amorstyle.com/dsn/?contract=0x44a8e5648334a6685c8425676d7c074e19c9a37f>

Note that the visitor is going to see something similar to this:



Standard DSN Contract

SimpleSwap:



DRM Audio Smart Contract

Whitepaper: <https://amorstyle.com/nfts/drmaudiov1/whitepaper.pdf>

Smart contract that is used by the Selene Network to tokengate audio files.

More Info: <https://amorstyle.com/drmaudiov1/>

Creator: AmorStyle

Artist: Amor's AI helper

[0x44a8e5648334a6685c8425676d7c074e19c9a37f](https://amorstyle.com/drmaudiov1/)

Totals: 1 Soulbound Single

Mint



Minting Price: ~\$4.99

Required Coin: 133.23 tfuel

Please validate the value and receiver in metamask.

Project Owner: [0x158D916596E7D76C029Cf35D027acf849869bB26](https://amorstyle.com/drmaudiov1/)

Project Titled: [0x158D916596E7D76C029Cf35D027acf849869bB26](https://amorstyle.com/drmaudiov1/)



[Participant NFT Project](#)

[Agent NFT Project](#)

[Website NFT Project](#)

This project builds upon a permissionless, decentralized sales network protocol that is built on the [Theta blockchain](#) where participation is encouraged and incentivized.

[Distributed Sales Network v1.21](#) Minting Ids - Agent Id: 1001, Website Id: 1.

Visitor: Consider becoming an Agent for the network.

Brought to you by [Amor, Penny Oracle](#)

© 2024-2025, AmorStyle

You should see a 'home' icon to the right of the contract address.

The ribbon next to the name of the project should most likely be blue. If the hash on the smart contract matches the project JSON file hash, the project will be shown as blue. In this case, I

Using home location for custom layout

<https://amorstyle.com/nfts/drmaudiov1/drmaudiov1.php>