# Page HOME : / A propos

* Mon profil (qui je suis),
* CV
* Parcours,
* Compétences,
* Liens divers

# Page Contact :

Formulaire de contact

# Page Portfolio

Liste de tous mes projets (terminés ou en cours)

# Page Rapport de Stage :

## Intro :

* Introduction sur l’entreprise (lieu, histoire, activités…)
* Contexte de travail,
* Objectifs du stage …

## Articles sur le stage :

* Intégration
* Différentes missions
* Difficultés rencontrées
* Nouvelles technos apprises…
* Compétentes utilisés
* Propositions de solutions a des problèmes
* Apprentissage
* Documentation de qualité

Liste des articles :

* + Création d’un NFT (decouverte des métadatas + du smart contact)
  + Création d’une collection de NFTs (Algorithme de création des métadatas + Smart contract dynamique)
  + Création du site de mint (React +bibliotheque etherjs)
  + Création Du générateur d’images pour NFT
  + Réalisation du site pour le voisin (a venir)

## Bilan du stage :

* Vos impressions, votre ressenti ;
* L’expérience acquise ;
* Les points d’amélioration ;

# Liste des Articles

## Introduction :

J’ai été contacté par Samuel Dubreuil, un 3D designer pour l’assister lors de la création de Nomad Gallery. Nomad Gallery est une entreprise qui a pour objectif de digitaliser le travail physique d’artistes en collaborant avec eux. Le process de digitalisation se fait en plusieurs étapes :

* L’œuvre d’art est refaite en 3D par l’équipe de Nomad Gallery,
* Une collection NFT à partir de cette œuvre est créé.

Lorsque que je suis arrivé, l’entreprise venait d’être créé et j’étais là pour m’occuper de la partie dev de l’organisation. Je précise que c’est la première fois que j’ai dû travailler sur des NFTs ou plus généralement sur le développement blockchain. J’ai donc dû apprendre sur le tas. J’étais également le seul développeur de l’entreprise donc j’ai tout appris en autodidacte. L’objectif final du stage est d’arriver à lancer correctement l’entreprise c’est-à-dire vendre la première collection de NFTs et à continuer de faire d’autres collections même après la fin du stage.

L’équipe de Nomad Gallery est composée de trois membres :

* Samuel, le fondateur et un des deux artistes 3D
* Kevin, le second Artiste 3D
* Moi qui m’occupe de la partie dev.

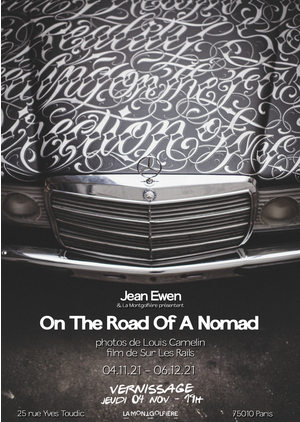
La première mission de Nomad Gallery était de digitaliser la pièce principale de l’exposition de Jean Ewen : « On the road of a nomad ».

Jean Ewen, est un artiste pluridisciplinaire dont le travail se concentre autour de l'univers de la calligraphie déclinée sous plusieurs supports et média. Il mêle des inspirations américaines et françaises pour construire son cadre et ses process de création, qui ne ressemblent à aucun autre.

On The Road of A Nomad est le dernier projet artistique de Jean Ewen.

Pendant plus de six mois, Jean Ewen a gravé entièrement à la main une carcasse de Mercedes D240. Il délivre ici sa vision du nomadisme, une définition de vie d'ici et d'ailleurs, en quête de construction personnelle.

Le photographe Louis Camelin et l'équipe de films Sur Les Rails, l'ont suivi tout au long du processus, comme un travail documentaire.

 Une image contenant route, extérieur, rue, transport

Description générée automatiquement

L’objectif de la mission était de reproduire en 3D la voiture gravée de Jean Ewen puis de réaliser un NFT de la voiture. J’ai été missionné de réaliser la collection de NFTs à partir du fichier 3D réalisé par l’équipe de Nomad Gallery.

## Article 1 Création d’un NFT :

Pour la première mission, je devais réaliser la collection de NFT à partir du fichier 3D de la reproduction de la voiture que l’équipe m’avais fourni.



gif voiture



Pour réaliser la mission qui m’a été donné je commence par rechercher comment faire un NFT. Je suis tombé sur plusieurs solutions :

|  |  |  |
| --- | --- | --- |
| Solution 1 : | Utiliser une solution toute prête (exemple : création via OpenSea) | Créer notre propre NFT Création des metadatas du nft et du smart contract |
| Inconvénient : | Le NFT n’est pas réellement crée avant son achat et on ne peut pas l’exporter sur d’autres plateformes, il ne nous appartient pas réellement.  Le terme utilisé est « Lazy mint » | Les frais de mint sont chers sur la blockchain ether en ce moment (une autre solution serait d’utiliser la blockchain polygone ou d’attendre la sortie de Ethereum 2.0) |
| Avantages : | Les frais de mint (création) sont au frais de l’acheteur donc si le NFT ne se vend pas, pas besoin de payer | Le NFT nous appartient totalement. |

J’ai finalement choisi la deuxième option qui nous permet d’être plus libre avec la création des NFTs et d’automatiser leurs créations, ce que nous verrons dans un second temps.

Il faut commencer par réfléchir aux metadatas (en français métadonnées).

Selon Wikipédia, Une métadonnée est une [donnée](https://fr.wikipedia.org/wiki/Donn%C3%A9e_(informatique)) servant à définir ou décrire une autre donnée. Les métadonnées synthétisent des informations élémentaires sur les données

Si l’on prend l’exemple d’un document Microsoft Word, les métadonnées sont la taille du fichier, sa date de création, le nom de l’auteur, la date des différents changements ou le nombre total de modifications. On peut aussi compter le titre, les étiquettes et les commentaires.

Dans le cadre des NFTs, les métadonnées jouent un rôle très important. En effet les fichier métadonnées résolvent un problème financier et technique : Il est impossible d’héberger des fichiers volumineux sur une blockchain. Par exemple un film en 1080p pèse entre 2 et 4 Go, le média Gemini estime qu’héberger 1 Go de data sur la blockchain Ethereum couterait environ 17,500 ETH. A l’heure où j’écris ces lignes, la valeur de ETH est de 1800 $ ça reviendrait à un cout de 31 000 000$ pour 1 Go.

Pour contrer ce problème, le token NFT qui est inscrit dans la Blockchain va être relié directement au fichier metadata réalisé en Json. Ce fichier va pointer vers l’url de l’artwork que l’on souhaite en NFT mais contient également les autres informations du NFT : Par exemple, si on souhaite faire un NFT d’une image, le fichier metadata va contenir l’url de l’image stockée mais également tous les autres paramètres du NFT (son nom, sa description, son créateur…).

Le standard de réalisation des fichier metadatas est le format JSON (JavaScript Object Notation).

Le fichier metadatas est stocké à l’intérieur du smart contract sous le paramètre URI (Universal Resource Identifier). On ne peut pas héberger directement le fichier Json dans la blockchain, les couts de réalisation du NFT serait trop élevé. Le paramètre URI pointe alors directement vers l’url du fichier metadata.

L’achat d’un NFT n’est donc pas l’achat d’un artwork mais plutôt l’achat d’un certificat d’authenticité de cet artwork.

Voici le fichier que j’ai utilisé pour réaliser les metadatas.

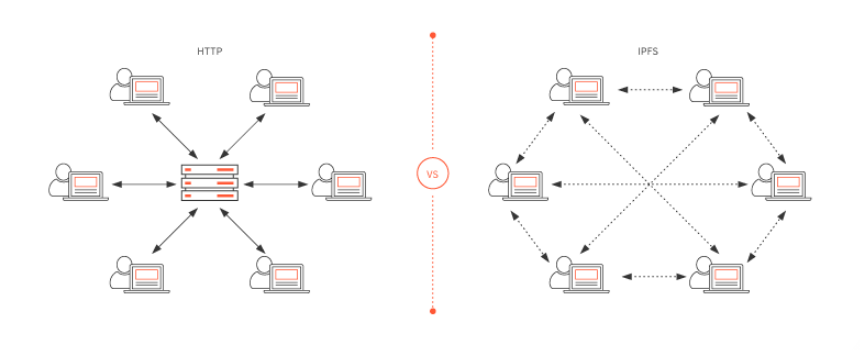
Une image contenant texte

Description générée automatiquement

On peut y retrouver les paramètres de notre NFT :

|  |  |
| --- | --- |
| **Paramètres :** | **Description :** |
| original\_creator | Adresse metamask du créateur du NFT, ici j’ai mis l’adresse de compte de Nomad Gallery. |
| Image | Lien de l’image uploadé sur IPFS, j’expliquerai le fonctionnement d’ipfs juste après. |
| animation\_url | Lien du gif représentant le NFT sur IPFS |
| tokenId  name | Numéro de série du NFT  Nom du NFT |
| description | Description du NFT |
| attributes | Tous les attributs particuliers du NFT. Ici on a juste besoin d’ajouter un paramètre de couleur qui va permettre de différencier la voiture Originale de la version Or. |
| file\_key | Lien du fichier téléchargeable de la voiture en haute qualité sur IPFS. |
| edition\_total | Nombre total de NFTs dans la collection. |
| royalty\_amount | Taux de droits d’auteurs qui nous sont reversés à chaque vente de NFTs. |
| copyright\_transfer | Nous ne voulons pas vendre les droits d’exploitations du NFTs donc on laisse ce paramètre sur false |
| resellable | Le NFT peut effectivement être revendu. |

Les metadatas ainsi que les medias des nft sont hébergés hors de la blochain, sur un service similaire à Google Drive. Pour les NFTs on a tendance à utiliser IPFS

IPFS (InterPlanetary File System) est un système distribué de fichiers pair à pair qui ne dépend pas de serveurs centralisés. L’objectif de ce système est de décentraliser la distribution de fichiers. 

Ce système a également pour objectif de résoudre les problèmes de http (Tous les fichiers sont centralisés dans les serveurs et sont vulnérables aux crashs et aux attaques DDOS. Les conséquences de ces vulnérabilités sont les vols de données, les fuites ou bien la perte de ces données).

Pour ajouter des fichiers sur ipfs, il faut télécharger l’application IPFS puis ajouter les fichiers que l’on souhaite dessus. Ensuite il faut récupérer l’adresses des fichiers uploadés et l’ajouter dans les metadatas

Une image contenant texte

Description générée automatiquement

Vous pouvez retrouver plus d’informations à propos d’ipfs [ici](https://cryptoast.fr/interplanetary-file-system-ipfs-reseau-partage-fichiers-web-3/)

Pour pouvoir créer le NFT à partir du smart contract, il faut installer metamask.

Metamask est un wallet (portefeuille) de [cryptodevises](https://www.comparateurbanque.com/epargne/trading/) basé sur la machine virtuelle Ethereum (Ethereum Virtual Machine). Il est donc compatible avec n’importe quelle Blockchain basé sur ether. Pour installer metamask, il suffit de télécharger l’extension metamask sur son navigateur puis de créer un compte.

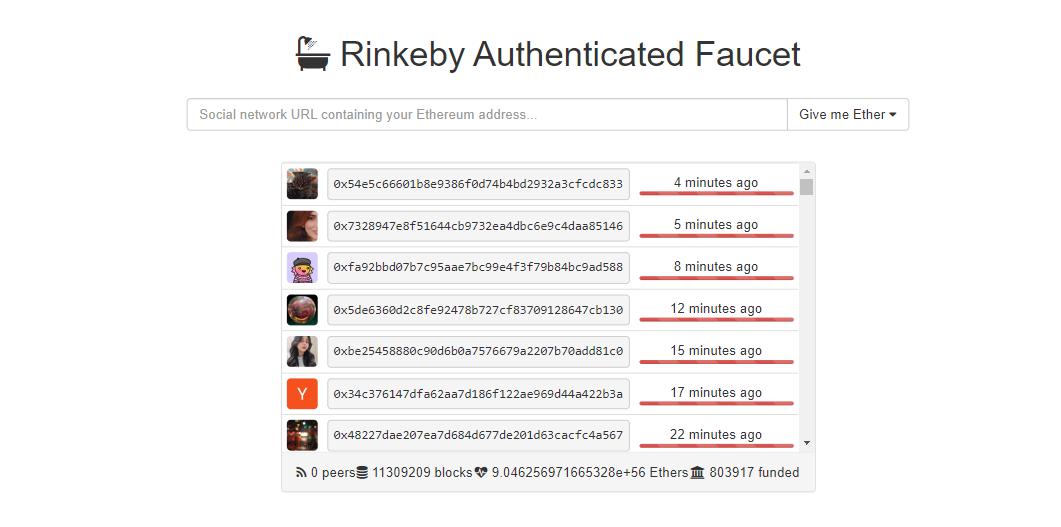
Déploiement du smart contact : On utilise L’IDE Remix pour cela :

Remix est un IDE en ligne qui permet de simplement lancer et déployer des applications sur la blockchain. Ici pour tester nous n’allons pas lancer le smart contract sur la blockchain classique mais plutôt sur une blockchain de test.

Un testnet est un réseau alternatif d'une blockchain servant d'environnement bac à sable pour que les développeurs puissent mettre à l'épreuve leur code avant le déploiement sur le mainnet.

Nous allons utiliser la testnet Rinkeby

Pour utiliser la blockchain de test, il faut récupérer des Ethers de test.



<https://www.rinkeby.io/#faucet>

Il suffit d’entrer l’addresse d’un compte metamask sur le site de rinkeby pour avoir accès à des Ethers sur cette Blockchain. Ces tokens n’ont aucune valeur et permettent juste de tester des smarts contracts avant leur déploiement.

Le token est paramétré lors de la création du smart contract.

Un smart contract est l'équivalent informatique d'un contrat traditionnel. Cependant, à la différence d'un contrat traditionnel, dont l'exécution est régie par un cadre juridique, celle du smart contract ne nécessite l'intervention d'aucun tiers de confiance et est, en tant que protocole informatique, régie par le code informatique.

Le smart contract s'appuie sur la technologie blockchain pour sécuriser et rendre infalsifiables les termes et les conditions de son exécution.

Il faut donc créer le Smart Contract puis le déployer sur la blockchain.

Etant donné que les NFT utilisent la blockchain Ethereum pour fonctionner, le smart contract doit être rédigé en solidity.

Solidity est un langage de programmation, de type orienté objet, tout comme peuvent l'être le Javascript, le Python ou le C++. Il permet le développement de smart-contract sur la blockchain Ethereum.

Pour ce projet j’ai donc dû apprendre les bases de solidity. Il y a un cours gratuit pour solidity très simple à prendre en main, il s’agit de [crypto zombie](https://cryptozombies.io/fr/). J’ai donc appris les bases sur ce site ce qui m’a permis de créer le smart contract du NFT, je me également inspiré des autres contrats de NFT que l’on peut trouver sur internet ou sur le site de Etherscan, en effet la plupart des gens qui réalisent des projet crypto mettent en libre accès leurs smarts contract sur Etherscan.

Pour notre NFT, nous allons utiliser le token ERC-721. Le token ERC-721 a pour caracteristique le fait d’être unique, c’est ce qui fait la valeur du NFT, de plus ce token est impossible à contrefaire. Ce token a également la possibilité d’avoir une valeur différente d’un autre token du même smart contract. Cela permet par de créer une collection de NFTs avec plusieurs variantes d’artworks pour créer de la rareté dans les NFTs par exemple. Cette caractéristique n’existe pas dans le cadre du token ERC-20, c’est pour cela qu’il faut bien choisir le token pour ce type de Projet.

Pour notre smart contract, les deux principales fonctions sont le constructor et la fonction de mint :

* Le constructor est la partie qui va permettre de créer la collection de NFTs à partir du contrat, on y retrouve tous les paramètres du NFT qui vont être entrés lors de la création du déploiement du contrat.
* La fonction de mint permet de créer des NFT. Le lanceur du contrat peut choisir de mint lui-même les NFT ou bien de laisser à l’acheteur le mint et donc de payer les frais.



Une fois le contrat créé, il faut le compiler sur l’IDE Remix, Remix est un IDE en ligne qui permet de compiler des smarts contracts sur la blochain Ethereum.

Ici on va le compiler sur la blockchain de test

Une image contenant texte

Description générée automatiquement

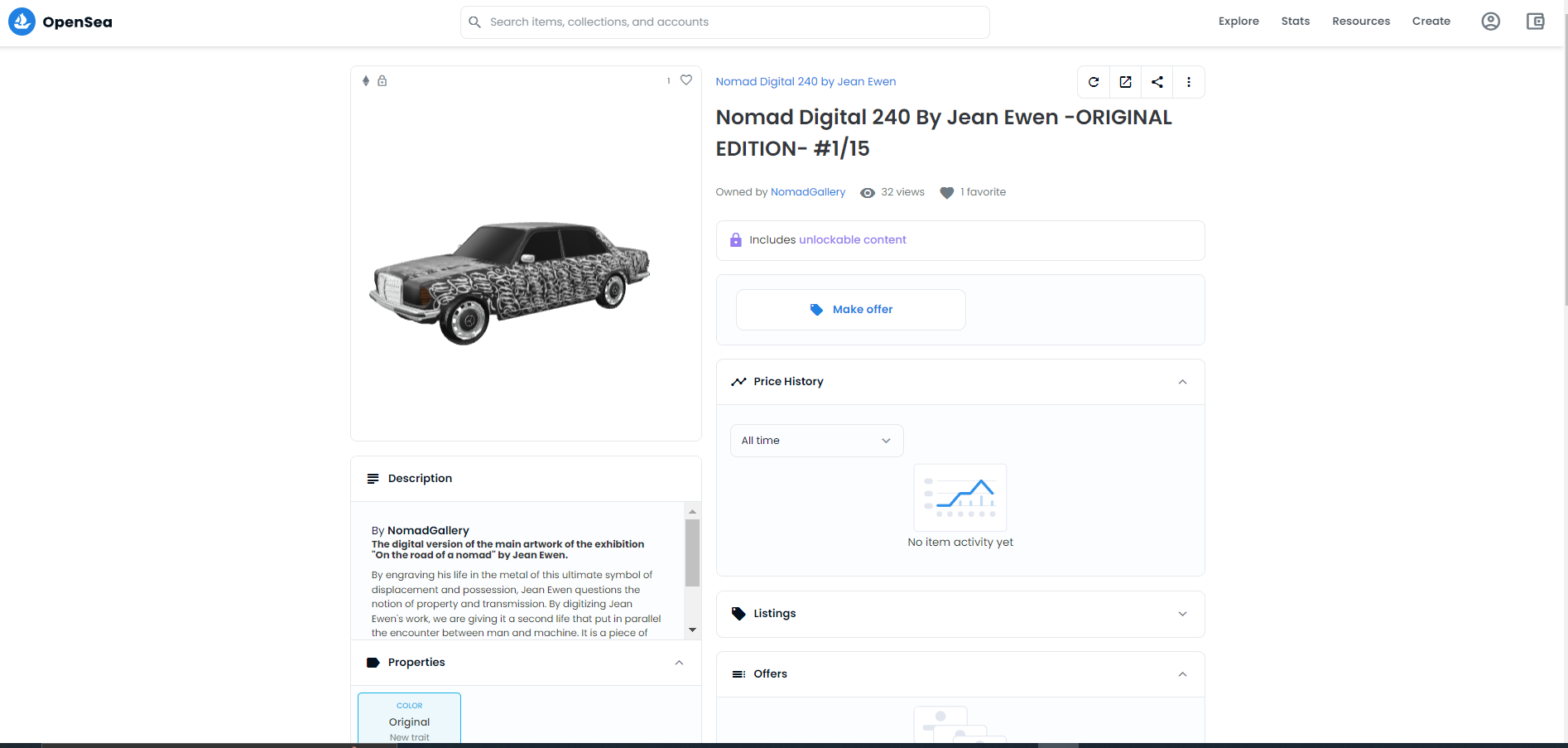
Ensuite il faut déployer les NFTs sur la blockchain en entrant les paramètres du NFT(Le nom de la collection, le lien des fichiers metadatas uploadés sur IPFS et le symbole du NFT).

Une image contenant texte

Description générée automatiquement

Il faut valider la transaction sur metamask et voilà, le NFT est déployé.

On peut ensuite visualiser la transaction sur Etherscan et voir le NFT sur le profil OpenSea lié au metamask



## Article 2 : Création d’une collection de NFT :

Le problème avec la manière de créer des NFT vu dans l’article 1 c’est que si je veux créer un grand nombre de NFTs, c’est trop long et fastidieux, pour une collection de plus de 100 NFTs c’est irréalisable. J’ai donc décidé d’automatiser la création de ces NFT

A partir du précédent projet il faut trouver un moyen d’automatiser la création de plusieurs NFT.

J’ai choisi d’utiliser un script Python qui va générer les metadatas des fichiers NFT à partir d’une image et des paramètres choisit par l’utilisateur.

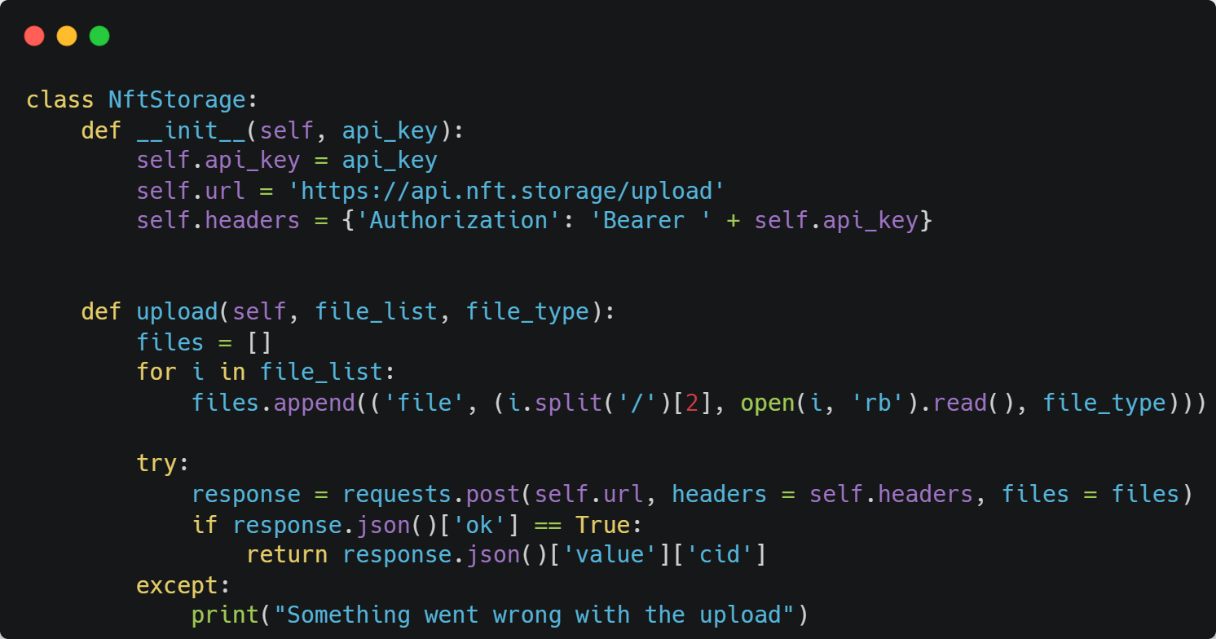
Python est le langage je connais le mieux avec JavaScript, Je l’utilise depuis maintenant 4 ans environ et c’est pour moi le moyen le plus simple et le plus adapté pour réaliser ce genre de projet qui contient de la manipulation de fichiers. Ça me fait gagner beaucoup de temps car je connais maintenant la syntaxe par cœur. Je l’ai par exemple utilisé au fil de l’année dernière pour le projet de YDays auquel j’étais inscrit : MyOwme (l’assistant vocal connecté).

Le script commence à générer les metadatas sous forme de fichier JSON à partir des paramètres choisis par l’utilisateur.

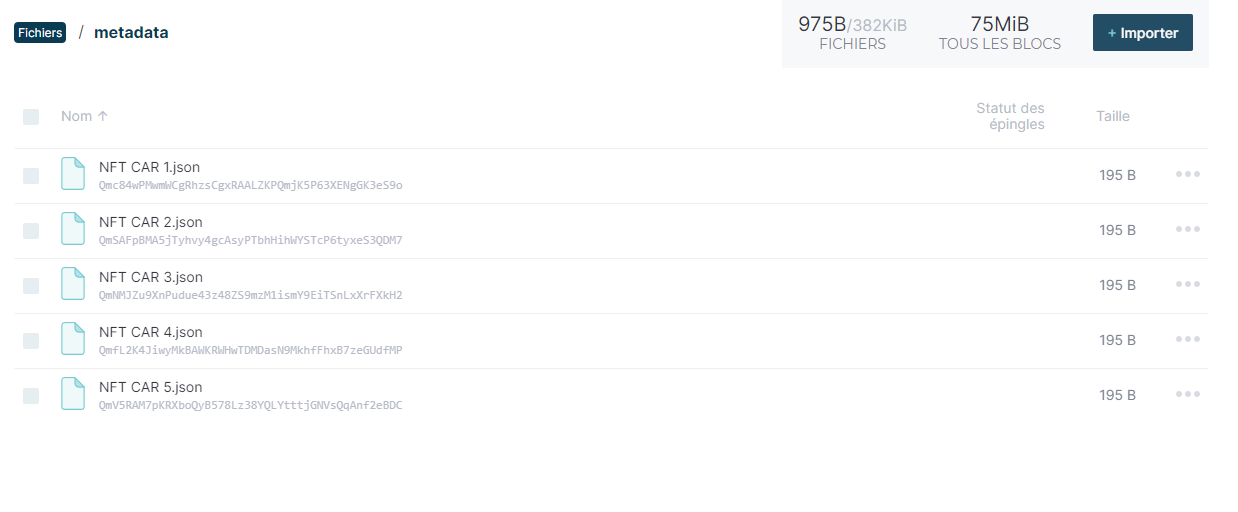
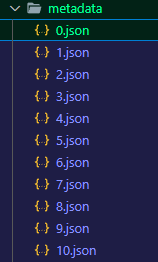


Une fois les fichiers crées j’ajoute au script une partie pour pouvoir upload les fichiers automatiquement sur ipfs. Pour l’upload sur IPFS, j’utilise l’API de NFTstorage qui va se charger de le faire automatiquement.

Plus d’informations sur NFTstorage : [NFT Storage](https://nft.storage/)



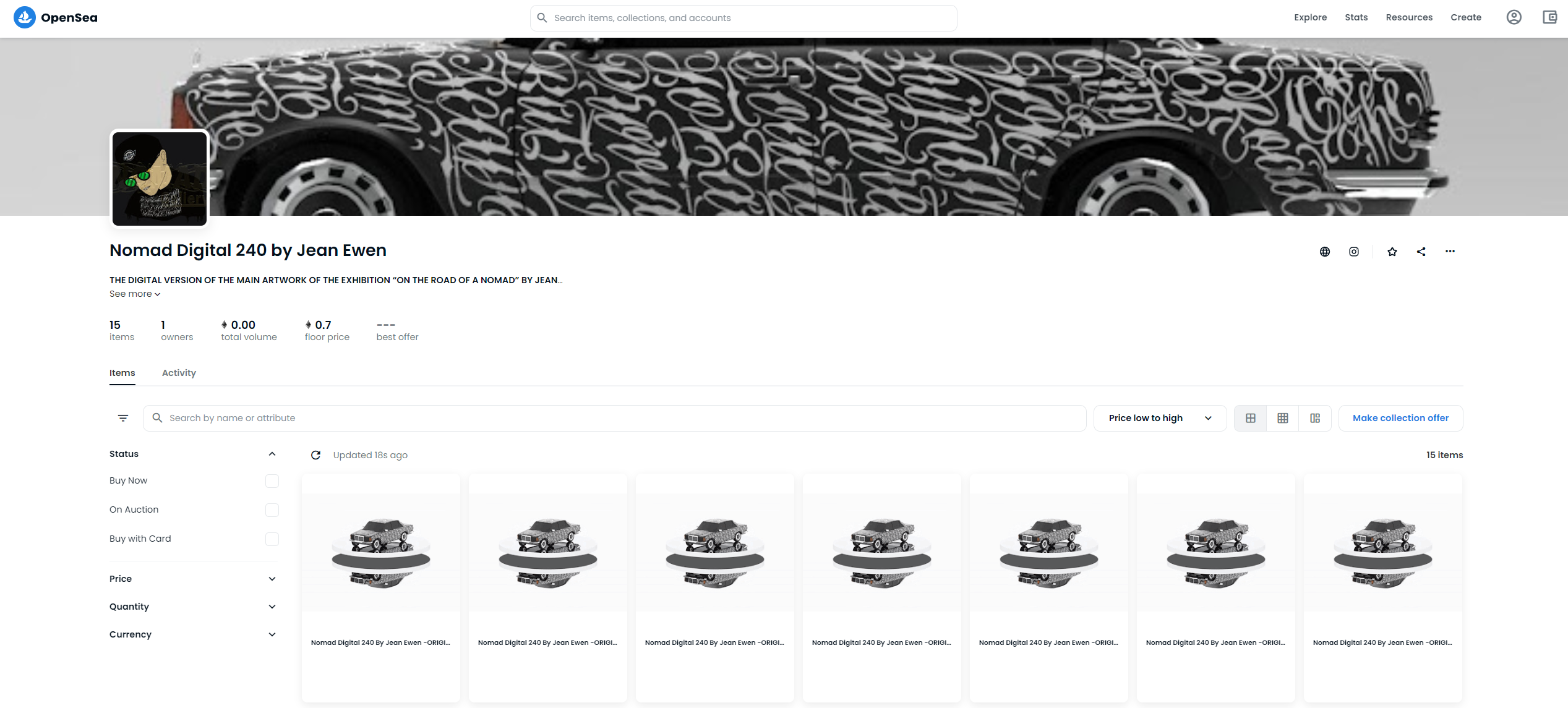
Une fois le script lancé, on peut alors retrouver tous les fichiers uploadés sur ipfs



En ce qui concerne le smart contract, j’ai repris celui réalisé précédemment qui est également fonctionnel avec ce projet

Je déploie tous les NFT avec Remix comme pour l’exemple précèdent.

On peut ensuite visualiser la transaction sur Etherscan et voir le NFT sur le profil OpenSea lié au metamask. Avec cette méthode nous avons pu réaliser l’entièreté de la collection de Jean Ewen sur Opensea.



Je suis réellement content de cette partie car j’ai réussi sur le plan technique à créer toute la collection de NFTs en apprenant tous les usages de la création sur le tas. J’ai également essayé de relier la création des metadatas a la création des NFTs automatiquement, mais j’ai réalisé que c’était une mauvaise idée car en cas de problème, on ne peut pas vérifier le contenu des metadatas et on pourrait alors perdre des frais de gaz. Cette méthode est un peu plus fastidieuse mais on évite les mauvaises surprises. L’autre problème est que l’utilisation des smarts contracts avec python est beaucoup plus compliqué que je ne le pensais, il faut utiliser Django, un framework que je ne connais pas. J’aurais donc dû apprendre en plus l’utilisation de ce Django pour un résultat équivalent.

## Article 3 : Création du site de mint (React +bibliotheque etherjs)

La collection a bien été réalisée et les NFTs de la voiture ont bien été créés. C’est à ce moment que nous affrontons notre plus gros problème : nous n’arrivons pas à créer de l’engouement autour de ce projet. Dans le monde des NFTs, il y a 2 façons de créer de la hype et de la valeur autour d’un projet : Le projet contient une base technique qui permet à celui qui achète le NFT de le rentabiliser. On peut prendre l’exemple des jeux Play To Earn qui permettent de gagner de l’argent en jouant à un jeu, le ticket d’entrée du jeu étant le NFT. L’exemple le plus parlant est axii infinity : ce jeu est un clone de pokemon basé sur la blockchain. Pour jouer a ce jeu il vous faudra acheter un premier monstre en NFT puis vous pourrez gagner de l’argent en gagnant des combats, il y a donc un intérêt financier a acheter le NFT, le jeu rapporte de l’argent et le NFT est rentabilisé. De plus cela permet de financer le jeu.

L’autre manière de créer de l’engouement est de vendre un projet artistique. Le créateur Beeple arrive à vendre des NFTs à plusieurs millions d’euros en étant un artiste connu.



Pour tenter d’attirer la communauté de l’artiste nous avons pensé à un moyen de la faire de la promo sur ses réseaux sociaux, Samuel m’a emmené faire le tour de Paris pour prendre en photo la voiture dans différents endroits grâce à la réalité augmentée. Au début j’ai pensé à utiliser un filtre instagram, l’application d’instagram, spark AR studio est assez simple à utiliser mais le résultat n’était pas celui que j’espérai. Finalement Kevin de l’équipe a découvert qu’avec les derniers iphones, il est possible d’utiliser des fichiers 3D en réalité augmentée très simplement. Nous avons utilisé cette méthode et nous avons posté les images sur instagram.



Malgré la mise en ligne des photos, l’engouement autour du NFT dans la communauté de Jean Ewen ne prends pas

Le problème du projet Nomad Mercedes et que son créateur Jean Ewen n’est pas suffisamment connu et que sa communauté n’est pas réceptive à la mode des NFTs.

Nous avons donc essayé une nouvelle approche : Attirer les gens de la communauté Crypto. En recherchant les projets qui marchent sur Twitter, j’ai vu plusieurs personnes qui mettent en places des plateformes de free mint, c’est-à-dire des plateformes ou les NFTs sont gratuits mais les frais de mint (création) sont à la charge de l’acheteur.

Après discussion avec le reste de l’équipe, j’ai pensé que si nous arrivions à créer de l’engouement autour de ces NFTs gratuits, nous pourrions créer une communauté qui seraient intéressés à l’idée de d’acheter nos NFTs.

Dans l’objectif de créer une communauté, j’ai créé un Discord pour nous permettre de donner les informations sur les projets à la communauté.

Ensuite il faut réaliser un site qui va permettre de mint gratuitement les NFTs.

L’avantage avec ce site est que les acheteurs iront eux même faire le mint du NFT, ils nous achètent le NFT au prix convenu et en plus ils payent les frais de mint. Comme ça si aucun NFT n’est vendu, nous n’aurons payés aucun frais de gaz.

Pour le design, Samuel a imaginé ce logo de Nomad Gallery en feu, c’est ce logo que nous allons mettre à disposition en NFT sur le site de freemint. Pour les metadatas j’utilise le générateur réalisé précédemment.

gif FreeMint

Finalement le smart contract est assez similaire au précedent, j’y ajoute des fonctions utilitaires.

Il faudra simplement appeler la fonction mint() du smart contract a chaque fois qu’un utilisateur souhaite mint un NFT.

Cette fois ci je vais déployer le NFT avec JavaScript en utilisant la bibliothèque HardHat, ça simplifie un peu le processus de Remix.

Il suffit de lancer ce script pour pouvoir déployer le NFT. Ensuite le script nous renvoie l’adresse du contrat.



Ensuite nous allons commencer à créer le site de mint, pour cela j’ai décidé d’utiliser React couplé avec la bibliothèque etherjs qui permet d’interagir avec un smart contract.

J’aurais pu utiliser un autre Framework pour le site de mint mais j’ai choisi ReactJS car c’est le framework le plus abouti en termes d’utilisation pour le web3 et c’est une technologie que je souhaite découvrir. Je précise que lors de ce projet, c’est la première fois que j’utilise React. C’est une totale découverte pour moi et je décide d’apprendre à utiliser ce framework en autodidacte. C’est assez déroutant pour moi car c’est la première fois que j’utilise réellement un framework FrontEnd et avant ce projet j’avais du mal à faire une réelle distinction dans l’usage d’un framework frontend et d’un framework backend (nous avions déjà utilisé Symfony et Golang pour réaliser des sites web statiques).

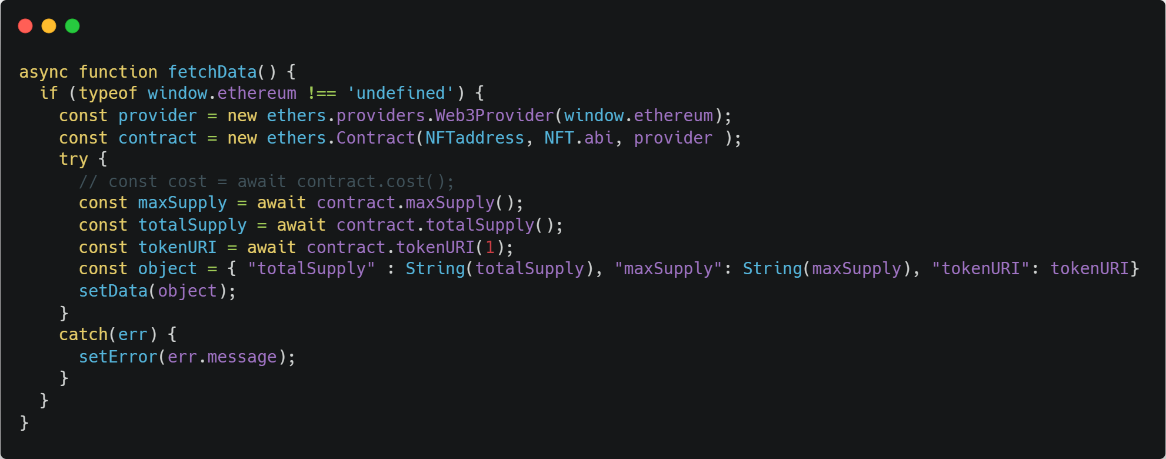
Pour relier le smart contact avec le site, on utilise Etherjs :

En réalité avec react il existe deux bibliothèques pour le web 3, Etherjs et web3.js mais la bibliotheque ether.js est plus complète, c’est pour cela que je vais l’utiliser.

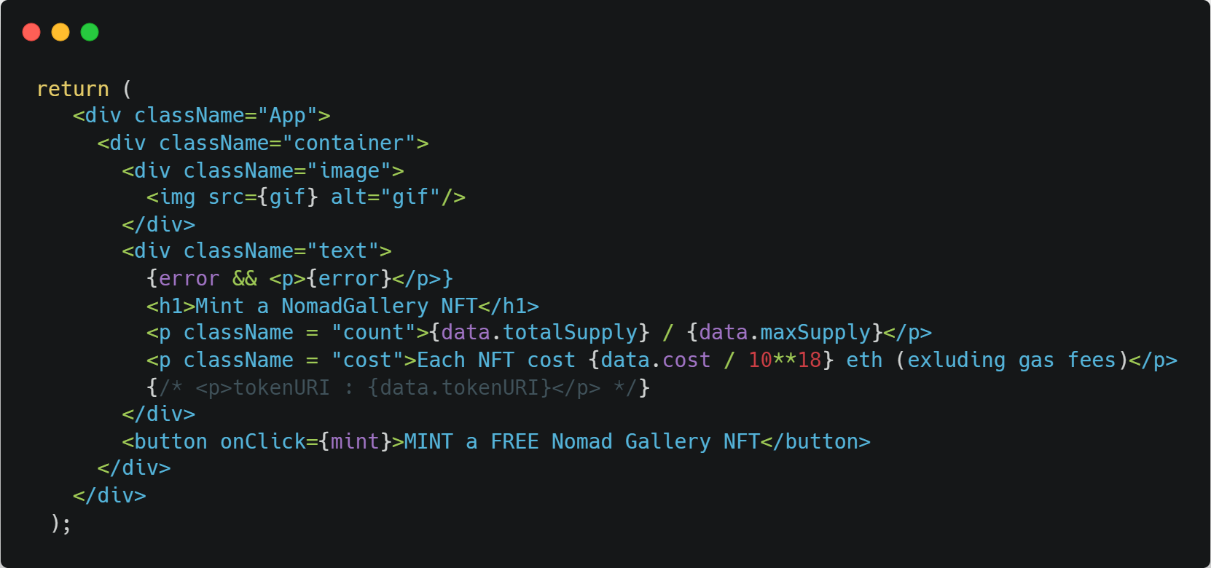
La première fonction Fetch Data permet de récupérer toutes les informations concernant le smart contract que j’ai déployé précédemment. Elle prend en paramètre l’adresse du NFT et retourne plusieurs valeurs :

* Le nombre maximum de NFTs pouvant être mint,
* Le nombre de NFTs ayant été mint
* L’adresse des metadatas uploadés sur IPFS

Ensuite elle retourne un objet contenant ces trois valeurs dans le hook d’état.



On affiche toutes ces valeurs dans le code HTML du composant. Le bouton va déclencher la fonction de mint.

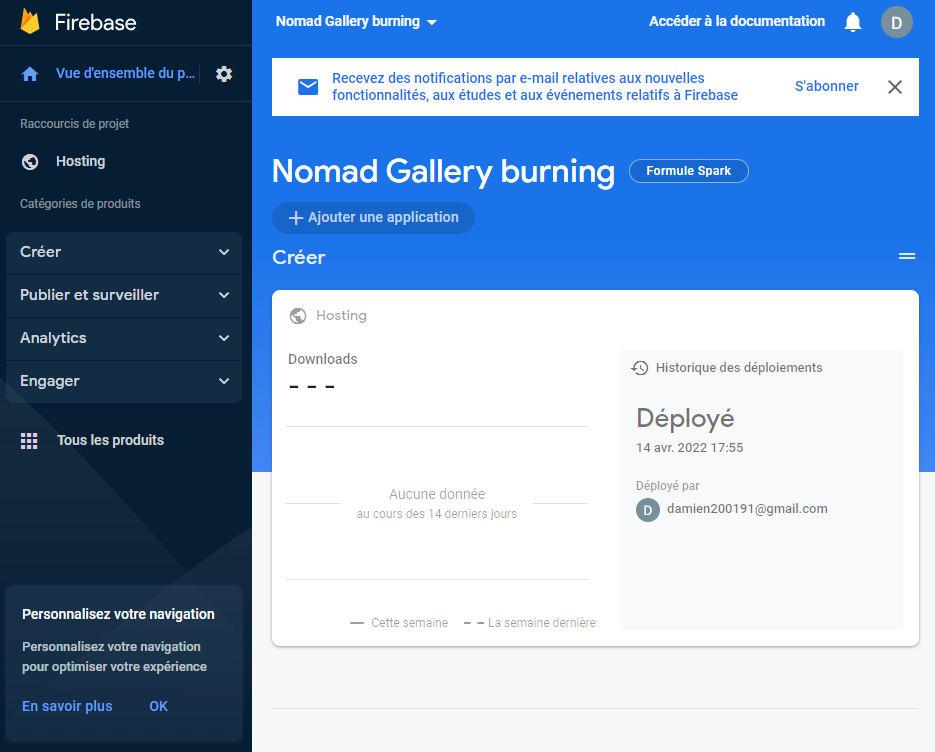


La fonction de mint permet comme son nom l’indique de mint un NFT. Elle va utiliser la bibliothèque Etherjs pour récupérer les informations metamask de l’utilisateur puis lui proposer de mint le NFT, elle fait le lien entre le metamask de l’utilisateur et la fonction mint du smart contract.

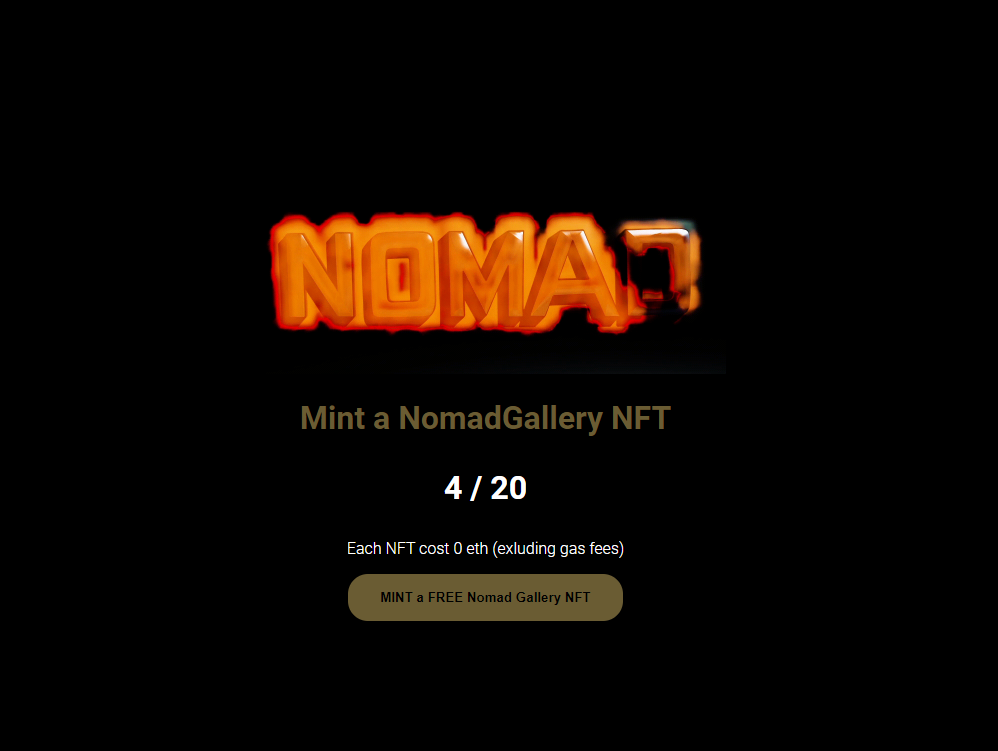


Ensuite j’ai dû trouver un moyen d’héberger le site, si possible gratuitement. J’ai utilisé FireBase pour cela :

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application. Ce service permet d’héberger des bases de données en NoSQL gratuitement mais également des sites web.



Voilà le site de mint est maintenant prêt à être utilisé.



[FreeMint NFT](https://nomad-gallery-burning.firebaseapp.com/)

Après le lancement du site de mint nous avons essayé de le promouvoir sur les différents réseaux sociaux mais rien à faire. Impossible de créer de l’engouement autour du projet. Avec du recul ce fut la partie la plus frustrante du stage, tout le travail était fait mais impossible d’avoir un retour sur investissement. Sans compter le fait que vers la fin du projet, la mode des NFTs a commencé à s’essouffler. Peut-être que si nous étions arrivés plus tôt sur le marché, le projet aurait pu prendre.

## Article 4 Création d’un genérateur d’image NFT :

Après l’échec du site de free mint, nous nous sommes concertés. Le projet a fait perdre beaucoup d’argent à l’entreprise et il n’y a pas eu de retour sur investissement : aucun NFT n’a été vendu. Nous sommes arrivés à la conclusion qu’il fallait changer de modèle économique. Nous avons donc pensé à revoir notre formule. Si un artiste veut collaborer avec nous, alors nous lui fournirons une collection de NFT entière et déployé sur la blockchain mais nous ne nous occuperons pas de la partie vente des NFTs. Cette façon de faire nous permettra alors d’être payé à chaque réalisation peu importe la réussite du projet ou non. De plus, cela va nous permettre de proposer d’autres services plus simples, comme une réalisation de sites web ou bien de la modélisation 3D. Ce genre de services peut être mis en place sur le site web de Nomad Gallery mais également sur Fiverr. Afin de proposer nos services sur ce site, nous avons commencé par créer un compte fiverr puis nous avons réfléchi à toutes les offres que nous pourrions proposer.

* Collection de NFT basée sur un modèle 3D
* Création d’un site de mint
* Création et conception d’un fichier 3D
* Création et déploiement d’une collection de 1000 NFTs uniques

Afin de pouvoir créer le service de collection de NFTs uniques, j’ai dû réaliser un générateur d’images à partir d’assets et la suite adapter le script précédent pour l’upload de NFTs pour l’intégrer dans celui-ci.

Pour cela il faut générer toutes les images puis les metadatas à partir de ces images.

La première étape du générateur d’images est de calculer le nombre d’images maximum pouvant être réalisés avec les assets fournis.

Le script boucle sur le nombre d’attributs et calcul le nombre d’assets par attributs. Il suffit de multiplier le nombre d’assets dans chaque attribut et on obtient le nombre maximum d’images.



Le script va ensuite demander à l’utilisateur le nombres d’images qu’il veut générer puis va lancer la génération.

La création d’images commence par aléatoirement relier les attributs entre eux, les images ne sont pas encore créées mais on ajoute dans un tableau la liste de tous les attributs d’une image stockés dans un dictionnaire Python. Ce tableau servira par la suite à créer les metadatas liés aux images.

Deux images identiques ne peuvent être crées, A chaque création d’images il y a une vérification dans le tableau que l’image est bien unique. Si le nombre d’assets est élevé, la probabilité de créer deux images identiques est faible mais si le nombre d’assets est limité alors ça peut être un problème. Il faut donc bien faire attention à ça.



A partir du tableau et des différents attributs, les images vont être crées.

Le programme boucle alors sur le tableau d’attributs et génère les images une par une. Il va alors chercher le nom des attributs dans les fichiers et fusionner les différents assets entre eux pour créer une image unique



Une fois les images créées, le programme demande à l’utilisateur s’il souhaite ajouter metadatas aux images précédemment crées. Les images vont alors être uploadés sur IPFS et les metadatas vont être générés.



Contrairement aux précédents fichiers de metadatas, celui-ci devra contenir les attributs de l’image (dans l’exemple d’un personnage, on peut imaginer ajouter un attribut pour chaque vêtement).

Le programme réutilise alors le tableau d’attributs pour les ajouter dans les metasdatas.

Ensuite la génération des fichiers est relativement similaire à celle réalisé dans le générateur précédent, il y a une boucle sur le nombre de NFTs crées et les metadatas vont être générés avec tous les attributs sous la forme d’un fichier Json



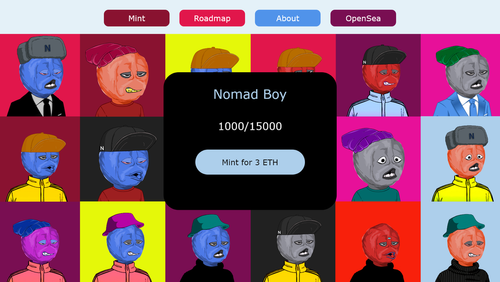
Le script va upload les métadatas sur ipfs, le script est similaire à celui réalisé précédemment dans l’article sur la création de la collection de NFTs. J’intègre donc la partie upload avec NFTstorage du script précédent dans le nouveau script.

Une image contenant texte

Description générée automatiquement

Pour finir je dois recréer un site de mint pour la collection de NFTs. Le processus est identique à celui de l’article sur la création du site de mint. J’adapte donc le smart contract de la DAPP. Il n’y a aucuns paramètres à modifier.

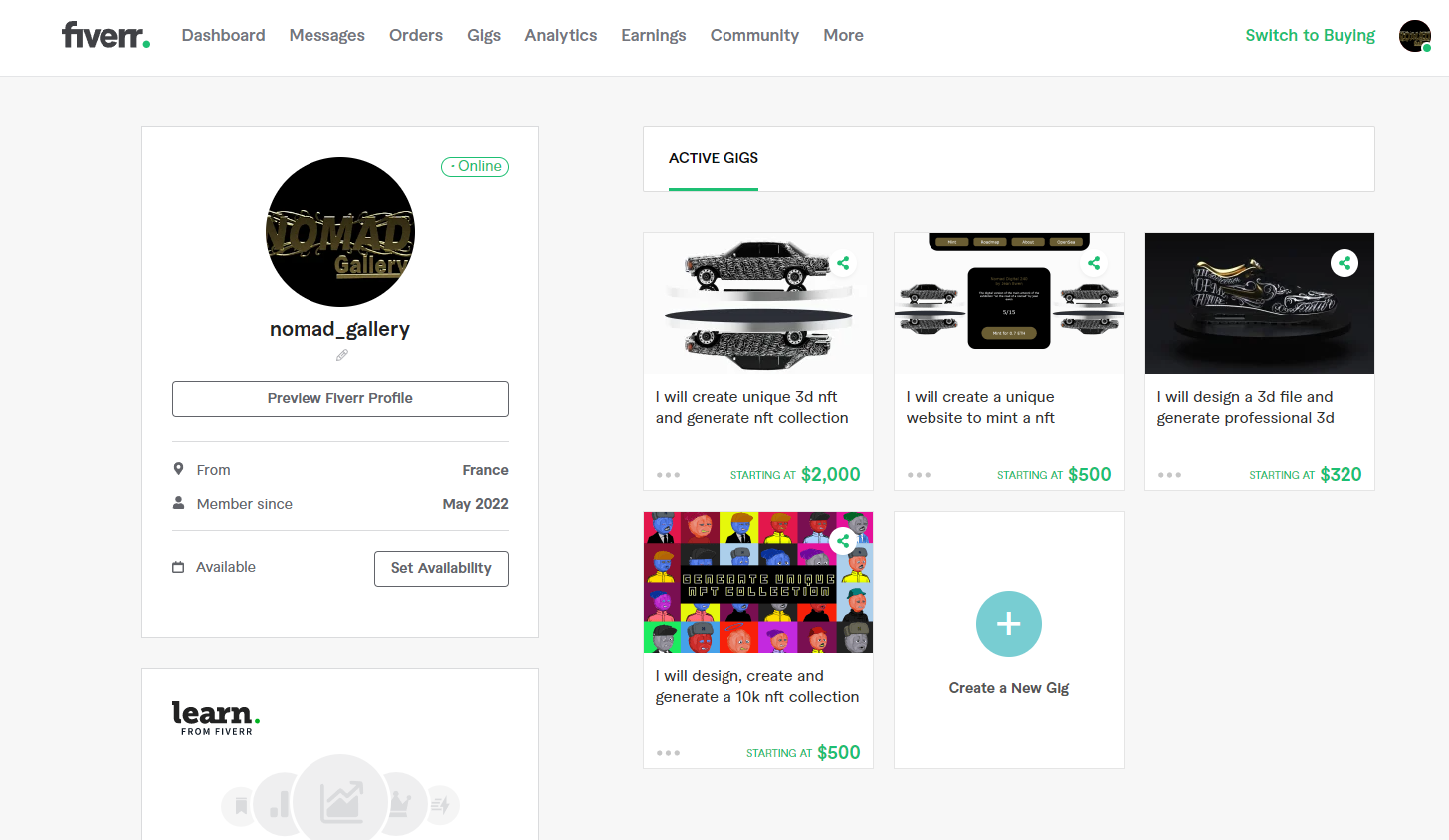
Ensuite pour la DAPP j’ai repris le code précédemment réalisé, puis j’ai revu le design du site pour qu’il colle avec le design de collection NFT.



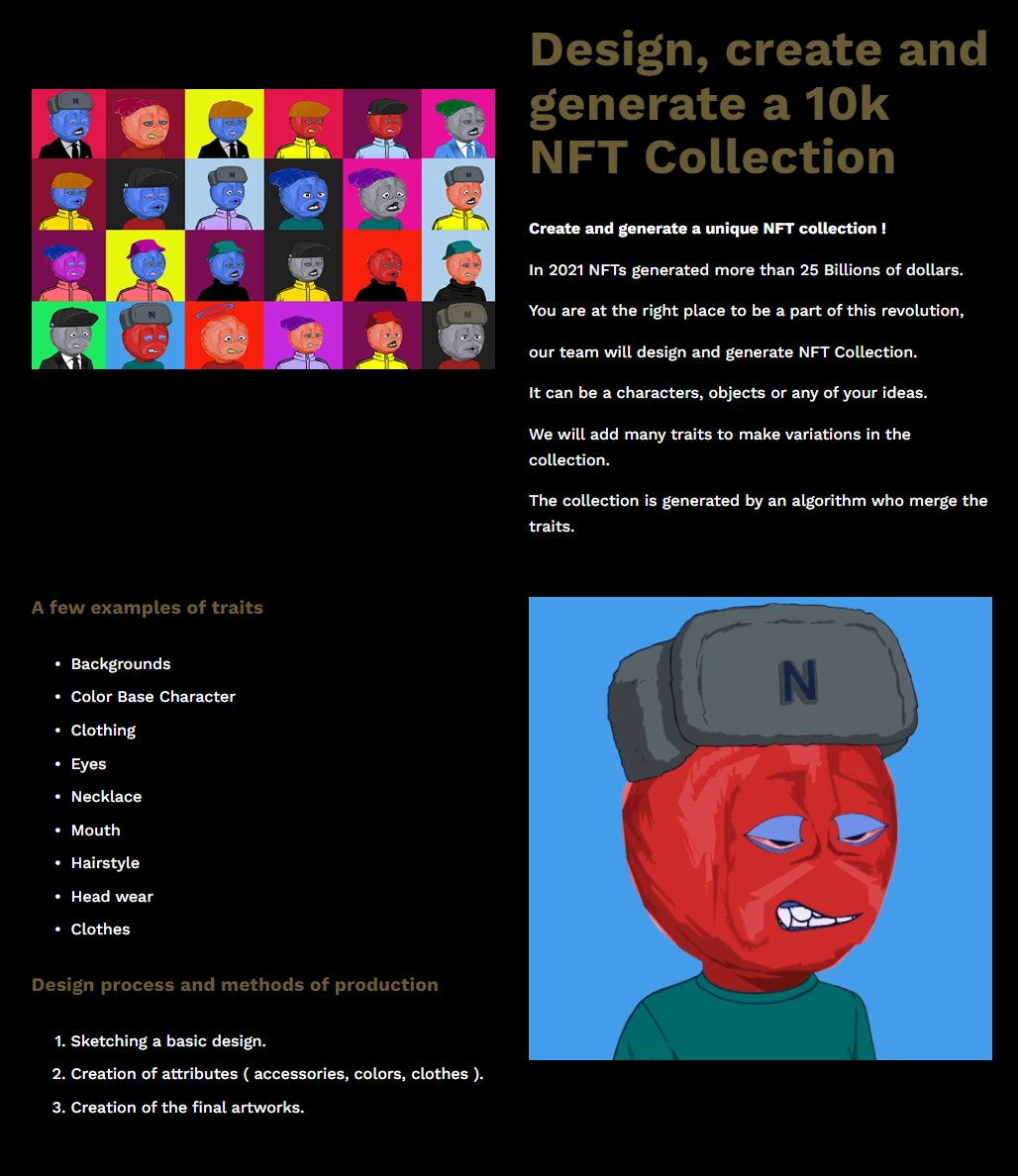
Pour le design, j’ai imaginé une webpage utilisant quelques images qui ont été générés afin d’avoir un rendu coloré.

Ensuite il ne reste plus qu’à upload le site sur FireBase comme réalisé précédemment.

Une fois le générateur réalisé, il faut créer le compte fiverr et mettre les annonces.

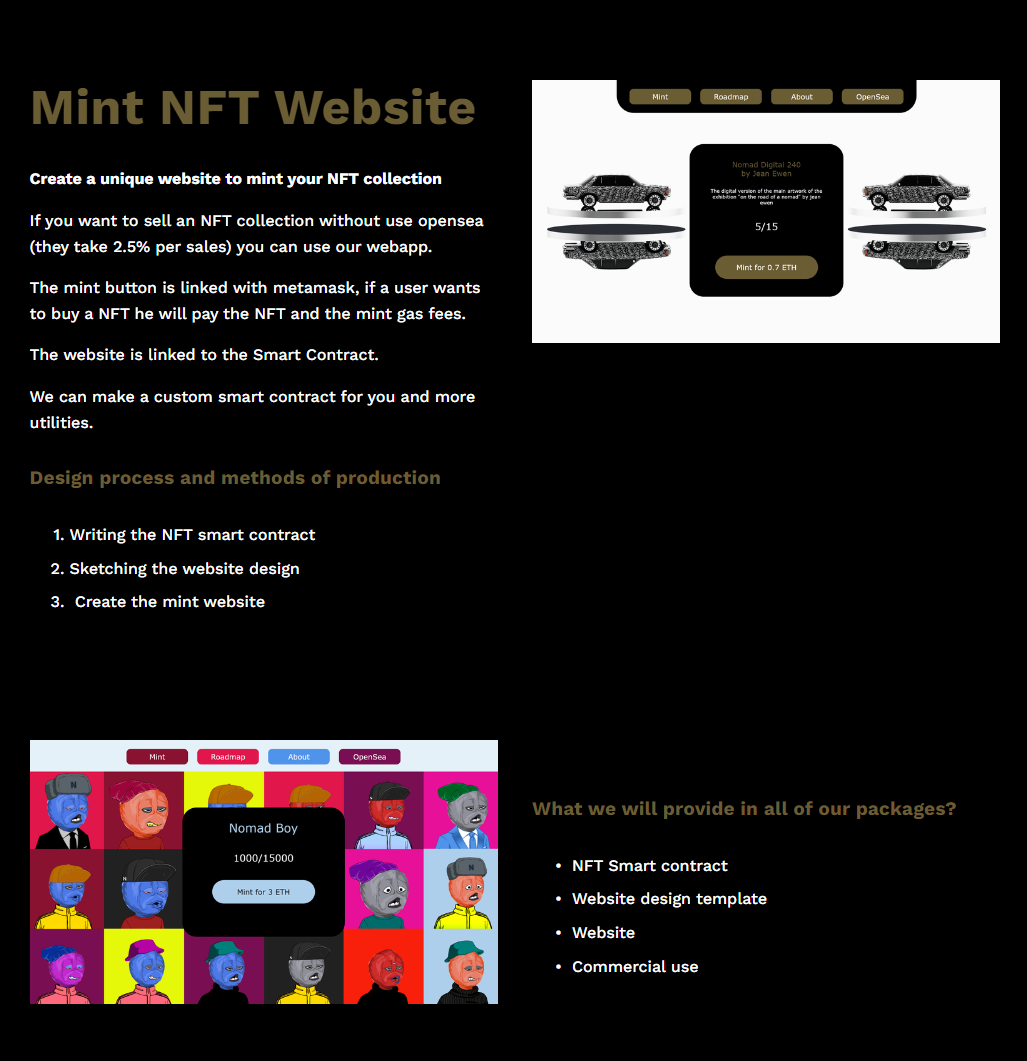


J’en ai également profité pour mettre les artworks réalisés sur le site web.



Une image contenant texte

Description générée automatiquement



## Conclusion :

Pour conclure ce stage, Samuel, mon référent de stage m’a emmené à une exposition ou la voiture était exposé. C’était une bonne conclusion a ce stage que de voir la voiture sur laquelle on avait travaillé durant plusieurs mois en vrai.

Images de l’expo

Pour faire un bilan plus général, je suis vraiment heureux d’avoir pu faire cette aventure avec toute l’équipe, l’ambiance de travail était très bonne. J’ai pu découvrir et apprendre beaucoup de choses :

J’ai découvert tout le milieu de la blockchain et comprendre les enjeux derrière celle-ci, beaucoup de gens sont passionnés par cette technologie mais malheureusement, on retrouve beaucoup de gens qui ne sont la que pour faire du profit, parfois au dépend des autres. J’ai vu beaucoup d’arnaques et de gens mal intentionnés dans ce milieu.

En termes d’apprentissage technique, j’ai pu m’initier au développement de blockchain, j’ai appris à utiliser le solidity et le react. Après avoir découvert ces technologies et avoir appris le fonctionnement et les enjeux de la blockchain, je pense pouvoir affirmer que c’est dans ce domaine que je souhaite m’orienter. Je me spécialiser dans le développement de blockchain. Le développement Solidity est un domaine sur lequel je compte me perfectionner dans les années à venir et j’aimerai réaliser des projets persos qui utilisent cette technologie.

Sur le point de vue social, j’ai dû apprendre à réaliser des projets avec des gens ayant différents métiers et intégrer leur façon de travailler dans mon processus. Le travail dans une entreprise était une première pour moi et j’ai découvert cet univers

Le point négatif est que le projet n’a pas pu aboutir. Les NFTs n’ont pas réussis à se vendre et quand nous sommes arrivés à la fin du stage, la mode des NFTs s’est essoufflé. Nous avons alors dû nous faire une raison, faire des NFTs pour les collectionneurs d’art n’est pas forcément à la portée de tous et la principale chose qui intéresse les gens du milieu c’est la spéculation. Le projet aurait peut-être pu aboutir si nous avions eu un spécialiste de la communication dans l’équipe. Je pense que c’est sur ce point que nous aurions pu améliorer l’entreprise. En effet s’occuper nous même de la communication a été une tache fastidieuse et s’est révélée être un échec.

Pour finir sur une note positive, je continuerai de travailler avec les artistes de Nomad Gallery sur des projets de sites web et je suis très enthousiaste à l’idée de pouvoir poursuivre la collaboration.