HackaDon 2018

Tutoriel : Mise en pratique de BlockSY

Introduction

BlockSY est une API visant à faciliter l'utilisation d'une ou plusieurs Blockchain.

L'API BlockSY se décompose en trois parties :

- Gestion des Identifiants
  - /identity
- Gestion des Transactions
  - /transaction
- Gestion des Paiements
  - /payment

Nous verrons dans ce tutoriel comment on peut créer des transactions avec une Blockchain Ethereum sur le réseau de test Rinkeby (http://rinkeby.io).

Une transaction va être inscrite comme dans le marbre dans la blockchain, et elle sera accessible par tout le monde.

Une transaction peut contenir des données, qu'il est pour cela préférable d'encrypter si elles sont confidentielles.

Afin de pouvoir créer une transaction nous avons besoin :

- d'un émetteur
  - qui a des Ether (devise de la blockchain Ethereum) pour supporter le coût d'émission de la transaction
- d'un destinataire
- de données
- d'un montant à transférer
  - qui peut être à 0

Pour l'émetteur nous utiliserons une des entités pré-créées pour le HackaDon.

Création d'une entité destinataire

Dans un premier temps nous allons créer une entité afin de constituer pour ce tutoriel un destinataire de la transaction.

La création d'une entité se fait avec l'API REST de BlockSY, en émettant une requête HTTP de type POST sur l'URL /api/v1/identity/entity :

[POST] /api/v1/identity/entity

Avec le document JSON suivant :

```
{
   "client_id": "Nom unique",
```

```
    "spec_version": 1,
    "created_by": "blocksy",
    "name": "Nom unique",
    "type_uid": "111a2222-ffff-44cc-bbbb-aaaaa1111110",
    "type_version": 1,
    "status": 1
}
```

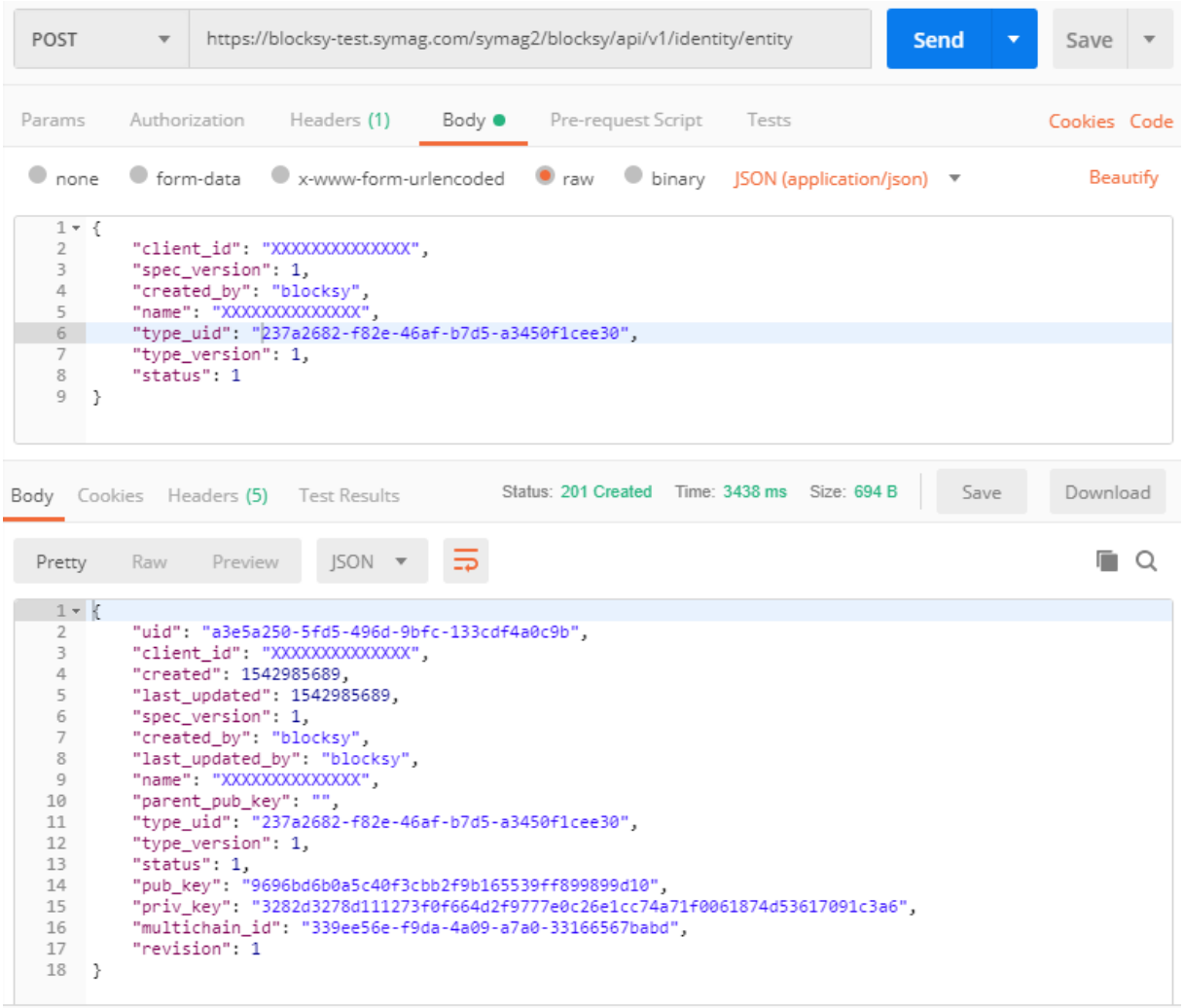
Il vous faut remplacer dans le document JSON :

- **« Nom unique »** par un nom arbitraire d'entité que vous souhaitez créer.

- « **111a2222-ffff-44cc-bbbb-aaaaa1111110 »** par le type de blockchain fourni par HackaDon :

Instance BlockSY	**Type de blockchain à utiliser**
https://blocksy-test.symag.com/symag2/blocksy/	237a2682-f82e-46af-b7d5-a3450f1cee30
https://blocksy-demo.symag.com/symag/blocksy/	0d3332ad-89e0-4d90-9312-c64085049747

La capture d'écran ci-dessous illustre un exemple d'appel réalisé avec l'outil Postman pour créer une entité **XXXXXXXXXXXXX** :



La partie supérieure de la capture d'écran montre le corps de la requête HTTP POST, tandis que la partie inférieure affiche la réponse JSON de BlockSY représentant la nouvelle entité créée. Notez que le statut de réponse HTTP, ici 201, correspondant à une création de ressource.

Création d'une transaction

Effectuons maintenant une transaction sans valeur (montant à 0), d'une entité prédéfinie hackadon2018-x vers notre nouvelle entité créée à l'étape précédente ; pour cela, nous allons envoyer une requête HTTP de type POST sur l'URL /api/v1/transaction :

[POST] /api/v1/transaction

```
{
  "priv_key": "be777aaaa0000000ffffff000000ddddddbbbbbb999999444444eeeeee55555",
  "password": "",
  "recipient": "333377777777777ccccccccccc55555555aaaaaaa",
  "digest": "donnée à inscrire dans la blockchain",
  "amount": 0
}
```

L’émetteur de la transaction est représenté par sa clé privée « priv_key ».

Vous mettrez dans le champ « priv_key » à la place de « **be777aaaa0000000ffffff000000ddddddbbbbbb999999444444eeeeee55555** » la valeur « **priv_key** » fournie par **HackaDon** pour l’une des entités **hackadon2018-x**.

Le destinataire est représenté par le champ « recipient », il s’agit de la clé publique hashée du destinataire.

Vous mettrez dans le champ « **recipient** » à la place de « **333377777777777ccccccccccc55555555aaaaaaa** » la valeur « **pub_key** » que vous avez obtenu lors de la création de votre entité (pour l’exemple fait avec Postman il s’agit de 9696bd6b0a5c40f3cbb2f9b165539ff899899d10).

Dans le champ « **digest** » vous pouvez mettre vos données que vous souhaitez inscrire dans la blockchain.

Attention : si vous souhaitez spécifier un montant, utilisez par exemple 3000, le champ « **amount** » définit une valeur entière exprimée en **Wei**.

Utiliser des valeurs trop importantes diminuerait trop rapidement les comptes HackaDon qui n’ont initialement que 3 **Ether**.

La dénomination Wei représente la plus petite mesure pour la crypto monnaie Ethereum, c’est un peu comme le cent pour l’Euro.

Un **Wei** représente 10^18 **Ether**, ou encore 1 Ether représente **1 000 000 000 000 000 000 Wei.**

Voici un exemple de réponse réussie (statut de réponse 201) que vous allez obtenir suite à la demande de création de transaction :


```
{
    "uid": "c21a1cd5-f60f-4aa5-bcca-217d1d00b529",
    "created": 1542207426,
    "last_updated": 1542207426,
    "from": [
        "9987c8fe3385ad4e422b2e33cc4f34056c20a315"
    ],
    "to": [
        "d33376acf6529e33d2020c2c341b839ca7d9a6b0"
    ],
    "amount": 3000,
    "block_height": -1,
    "confirmations": -1,
    "digest": "",
    "blockchain_ids": [
        {
```

```
            "id": "9bfe1014cfa47cc0693d712b535185cbc4b6f236e4755b864fa1d38ca6653688",
            "coin_type": 66145000
        }
    ],
    "status": "executed"
}
```

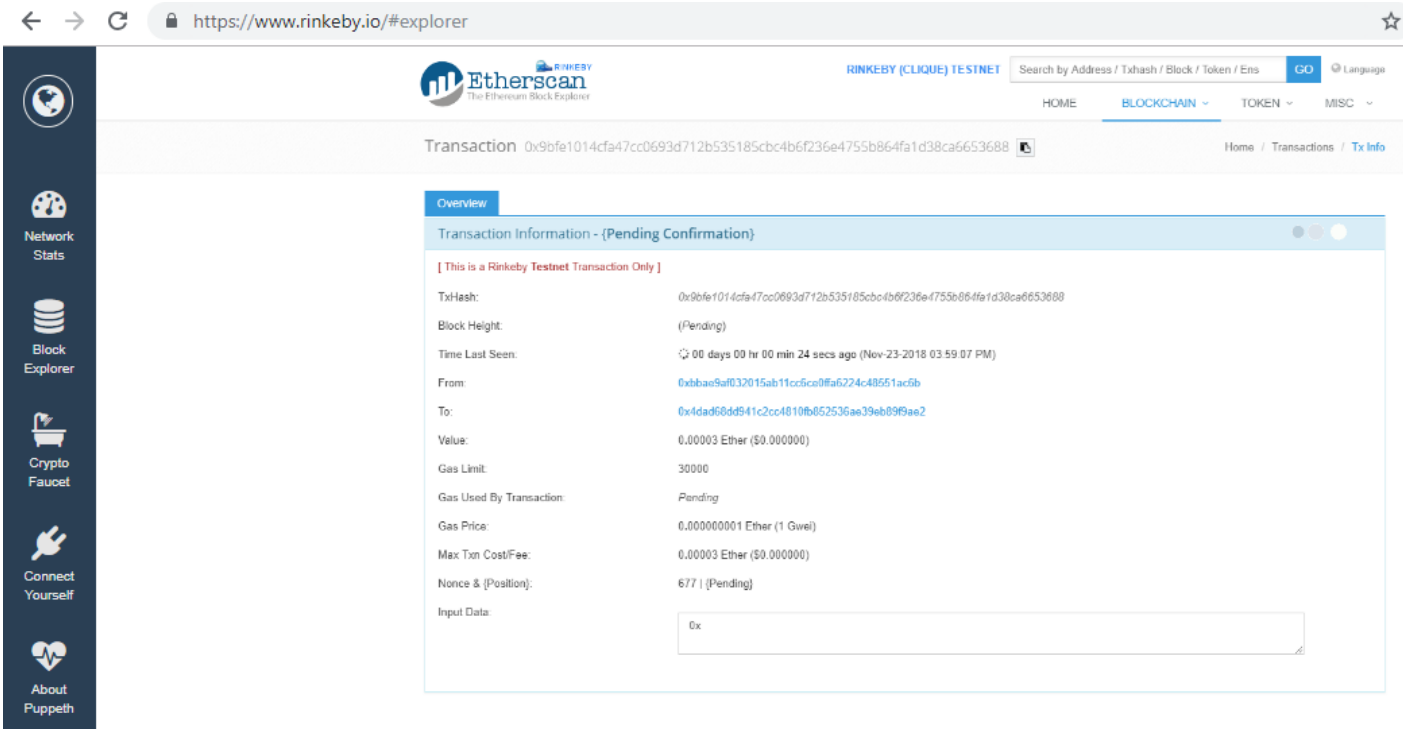

Le champ « **id** » dans « **blockchain_ids** » contient l'identifiant de la transaction dans la blockchain cible.

Les propriétés « **from** » et « **to** » indiquent respectivement les clés publiques hashées de l'entité émettrice et de l'entité destinataire.

Retrouver une transaction sur Rinkeby

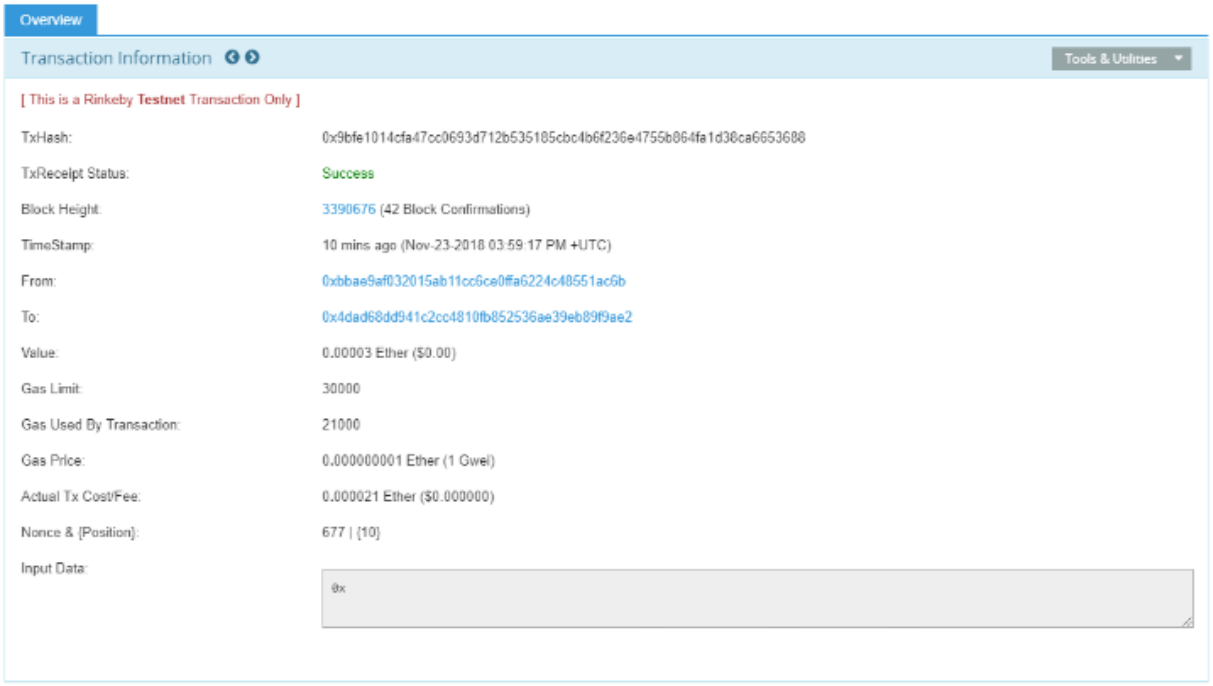
Si vous allez sur le site https://rinkeby.io, dans la rubrique « Block Explorer », vous pouvez retrouver la transaction que BlockSY a créée pour vous, en saisissant l'id mentionné à l'étape précédente dans le champ de recherche prévu à cet effet.

Vous obtiendrez alors des informations sur la transaction créée, comme sur la capture d'écran suivante :



Si la transaction n'est pas encore traitée, il est indiqué dans le titre « Pending Confirmation ».

Une fois traitée le champ « TxReceipt Status » indiquera « Success » en vert :



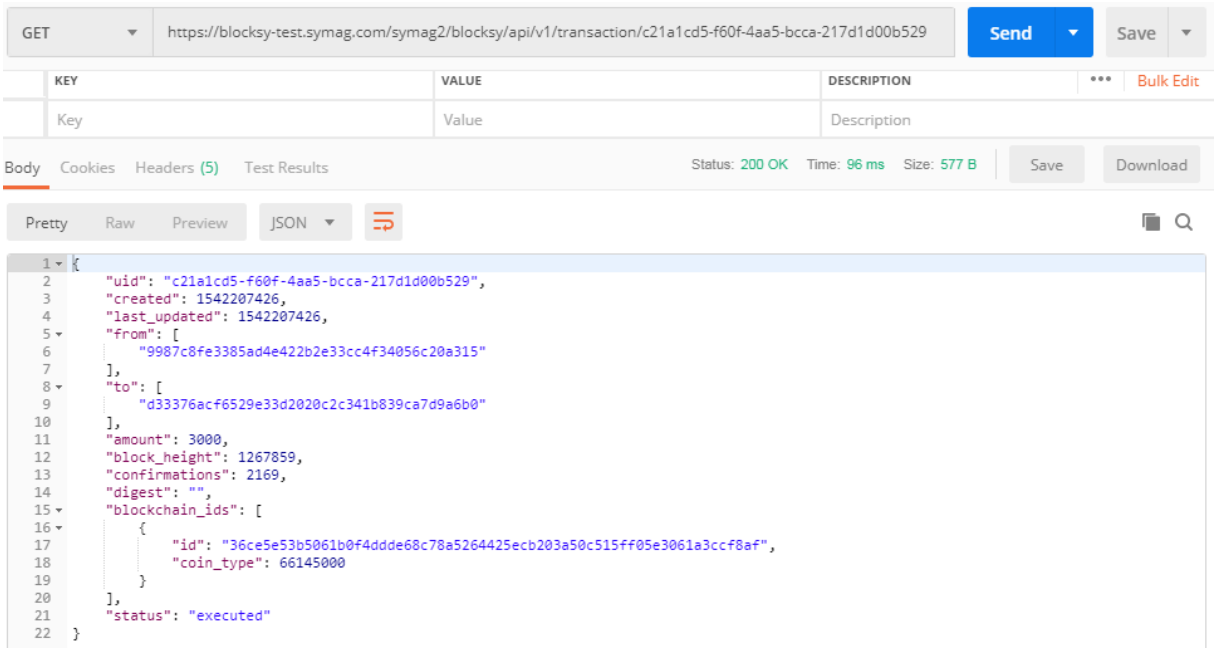
Remarque : si jamais vous avez spécifié des informations dans le champ digest lors de la création de la transaction, vous devriez alors constater que le champ « Input Data » faisant partie des informations de la transaction est différent de « 0x ».

Rechercher une transaction par BlockSY

Vous pouvez utiliser l'API BlockSY pour rechercher une transaction créée auparavant par celui-ci, en utilisant l'identifiant UID de la transaction.
Pour cela, nous envoyons une requête HTTP de type GET sur l'URL /api/v1/transaction/ suivi de l' « **uid** » de la transaction créée précédemment.

A titre d'exemple avec Postman, et en reprenant l'UID c21a1cd5-f60f-4aa5-bcca-217d1d00b529 que nous avions obtenu, BlockSY renvoie la réponse JSON décrivant la transaction demandée :

[GET] /api/v1/transaction



Outre le fait de pouvoir vérifier le statut d'exécution de la transaction (propriété « status »), on trouve également deux autres informations utiles :

- la propriété « **block_height** » indiquant à quel bloc de la blockchain la transaction appartient ;
- la propriété « **confirmations** », dont la valeur est actualisée à chaque appel, indiquant le nombre de confirmations pour la transaction.

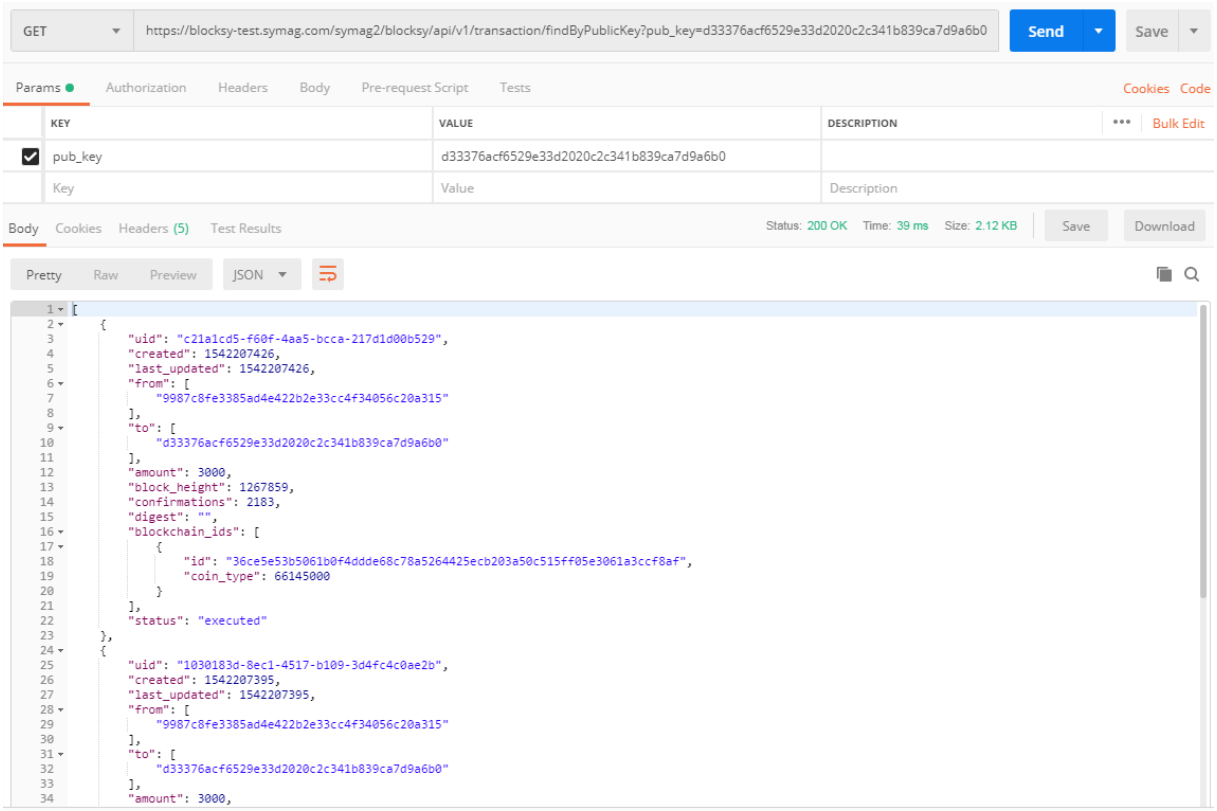
Rechercher les transactions d'une entité

L'API BlockSY permet également de lister les transactions émises par une entité donnée.

Voyons maintenant comment retrouver les transactions de l'entité destinataire créée au début de ce tutoriel ; pour cela, soumettez une requête HTTP de type GET sur l'API /api/v1/transaction/**findByPublicKey**?pub_key=<clé publique hashée>, où <clé publique hashée> représente la clé public hashée de l'entité ; celle-ci correspond à la valeur de la propriété « **pub_key** » de la réponse JSON à la création d'entité.

A titre d'exemple, la capture d'écran suivante de l'outil Postman, liste les transactions exécutées (réussies) relatives à l'entité de clé publique hashée d33376acf6529e33d2020c2c341b839ca7d9a6b0 :

[GET] /api/v1/transaction/findByPublicKey



Notez ici que la réponse JSON est une collection d'objets JSON de transactions.

Consultez également la documentation de l'API pour en savoir plus sur les paramètres d'URL pouvant être utilisés, car il est possible d'effectuer des recherches plus fines, mais aussi de pouvoir effectuer de la pagination.

Obtenir l'adresse d'une entité

L'API BLockSY permet d'obtenir l'adresse d'une entité dans une blockchain, à partir de sa clé publique.

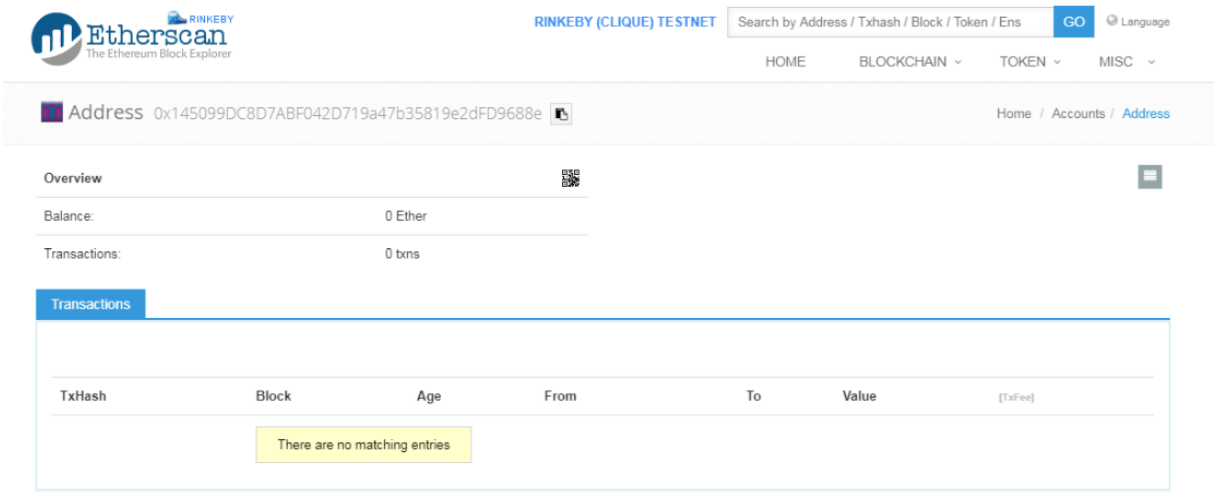
En l'occurrence, obtenez l'adresse Ethererum de l'entité créée au début du tutoriel en invoquant l'API /api/v1/payment/address/<clé publique hashée>, où <clé publique hashée> représente la clé public hashée de l'entité (celle-ci correspond à la valeur de la propriété « **pub_key** » de la réponse JSON de la création d'entité).

Il suffit donc d'émettre une requête HTTP de type GET sur une URL de la forme indiquée, comme par exemple :

/api/v1/payment/address/9696bd6b0a5c40f3cbb2f9b165539ff899899d10

Et BlockSY renvoie directement l'adresse demandée (type MIME text/plain), dans notre exemple : 145099dc8d7abf042d719a47b35819e2dfd9688e

Comme cela a été fait pour la recherche d'une transaction dans l'explorateur Rinkeby, vous pouvez également obtenir des informations relatives à l'adresse Ethereum 0x145099dc8d7abf042d719a47b35819e2dfd9688e, en lançant une recherche après avoir saisi l'adresse dans la zone d'édition prévue à cet effet :



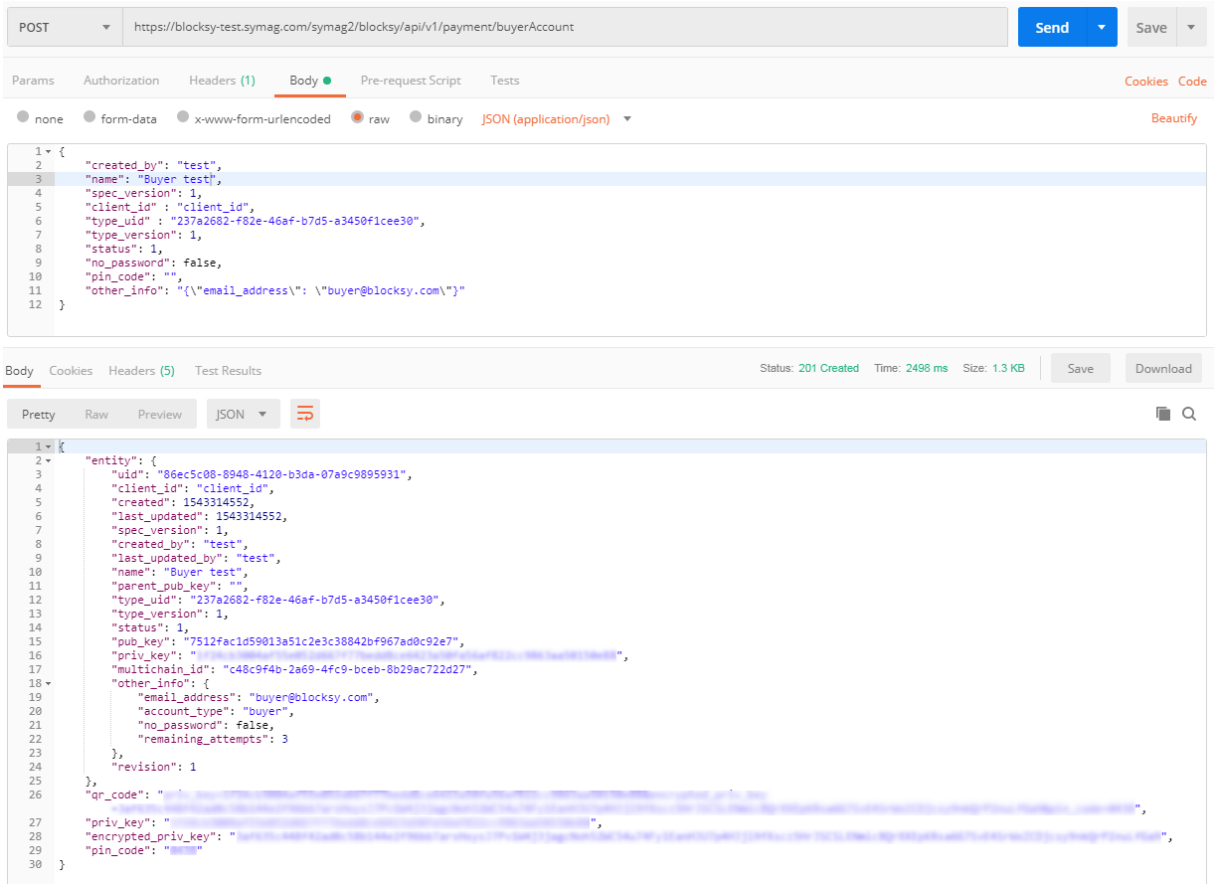
Notez que vous pouvez consulter le solde actuel du compte Ethereum, ainsi que toutes les transactions afférentes.

Création d'un compte « buyer »

Créer un compte « buyer » est essentiellement le fait de créer une entité avec des méta données supplémentaires qui vont permettre notamment d'avoir un code secret (PIN code) pour autoriser des paiements vers d'autres entités de la même blockchain.

Voici un exemple de création d'un compte « buyer » avec l'outil Postman :

[POST] /api/v1/payment/buyerAccount



Dans cet exemple, nous créons un compte « buyer » avec code PIN (indiqué avec « **no_password** » à false) qui sera déterminé aléatoirement par BlockSY (du fait de ne pas spécifier explicitement de valeur dans la propriété « **pin_code** » de la requête) ; ce dernier est retourné dans la propriété « **pin_code** » de la réponse JSON.

Une autre valeur importante retournée dans la réponse est celle de la propriété « **encrypted_priv_key** ».

En effet, afin d'autoriser un paiement d'un « buyer » vers une autre entité avec l'API /api/v1/payment, il sera nécessaire de passer la valeur de « **encrypted_priv_key** » ainsi que le code PIN (ce dernier permet de décrypter la clé privée qui a été encryptée).

Note : dans la requête JSON, notez de quelle manière nous avons ajouté une méta donnée supplémentaire (propriété « **email_address** ») dans la propriété « **other_info** ».

Création d'un ordre de paiement

La création d'un ordre de paiement permet de préparer un paiement.

L'ordre de paiement crée une transaction avec le statut « Prepared ».

Par rapport à une transaction classique, l'apport est de pouvoir envoyer à l'acheteur une demande d'autorisation par pin code.

Effectuons maintenant un ordre de paiement de 1 wei depuis une entité hackadon2018-x vers notre nouvelle entité créée précédemment ; pour cela, nous allons envoyer une requête HTTP de type POST sur l'URL /api/v1/payment/paymentOrder :

[POST] /api/v1/payment/paymentOrder

```
{
    "from_pub_key": "6ccccaaaaa4444444777777777111111111ffff",
    "encrypted_priv_key":
"3aXXXXXXXX22222222YYYYYYYY6bb7arsHxysJ7Pv1W4j3jagcNoh52WC54u74Fy1EanHJU7p4HJji9fXscz5HrJSCSLENmicBQrXXEpKRsa6G7SvE4SrWxZCDjcsy9nkQrPzzzzzzzz"
,
    "to_pub_key": "9696bd6b0a5c40f3cbb2f9b165539ff899899d10",
    "amount": 1,
    "currency_code": "ETH",
    "operation": "don"
}
```


from_pub_key représente la pub_key de l'entité qui souhaite émettre la transaction (Buyer).
encrypted_priv_key est la clé privée encryptée de l'émetteur obtenue à la création du Buyer.
to_pub_key est la pub_key de l'entité destinatrice de la transaction (Vendor).

Exemple de réponse obtenue :

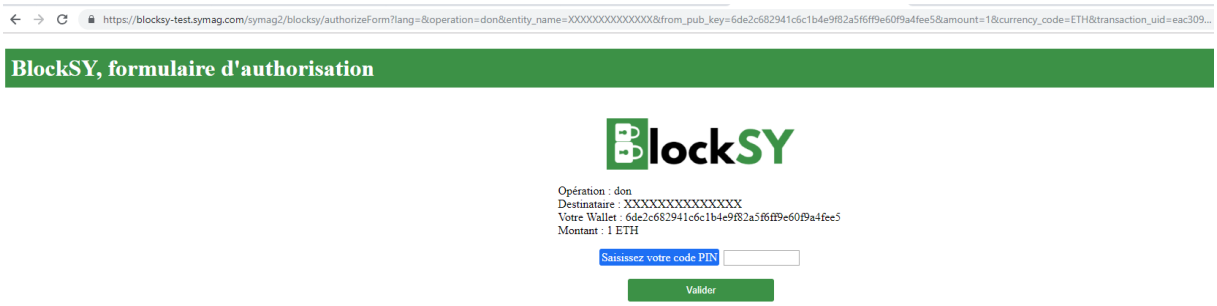

```
{
    "uid": "eac3094a-bcf6-46d5-9fa2-04d5bcaa1f9c",
    "created": 1543330563,
    "last_updated": 1543330563,
    "from": [
        "6de2c682941c6c1b4e9f82a5f6ff9e60f9a4fee5"
    ],
    "to": [
        "9696bd6b0a5c40f3cbb2f9b165539ff899899d10"
    ],
    "amount": 1,
    "block_height": -1,
    "confirmations": -1,
    "digest": "",
    "payment_order": {
        "amount": 1,
        "currency_code": "ETH",
        "operation": "don",
        "recipient_address": "145099dc8d7abf042d719a47b35819e2dfd9688e",
```

```
        "qr_code":
"bitcoin%3A145099dc8d7abf042d719a47b35819e2dfd9688e%3Famount%3D1%26label%3DX
XXXXXXXXXXXX%26message%3DPurchase+at+XXXXXXXXXXXXX",

        "redirect_url":
"/authorizeForm?lang=&operation=don&entity_name=XXXXXXXXXXXXX&from_pub_ke
y=6de2c682941c6c1b4e9f82a5f6ff9e60f9a4fee5&amount=1&currency_code=ETH&transa
ction_uid=eac3094a-bcf6-46d5-9fa2-
04d5bcaa1f9c&encrypted_priv_key=3aXXXXXXXX22222222YYYYYYYY6bb7arsHxysJ7Pv
1W4j3jagcNoh52WC54u74Fy1EanHJU7p4HJji9fXscz5HrJSCSLENmicBQrXXEpKRsa6G7
SvE4SrWxZCDjcsy9nkQrPzzzzzzzz"

    },

    "blockchain_ids": [],

    "status": "prepared"

}
```


Dans la réponse **redirect_url** permet de construire une url à faire ouvrir par l'acheteur pour qu'il puisse saisir son pin code.

Il faut combiner l'url de blocksy (par exemple « https://blocksy-test.symag.com/symag2/blocksy/ » pour la 1ère instance) avec la valeur fournie par le **redirect_url**.

Exemple de page qui sera alors affichée par le navigateur web :



L'acheteur saisit son pin code qui lui a été attribué (voir l'étape *Création d'un compte « Buyer »*, la réponse à la création du compte fournit le **pin_code**) et valide ainsi le paiement.

BlockSY va alors inscrire la transaction dans la blockchain, et passer le statut de la transaction à « **executed** ».