

Documentation utilisateur

Cette documentation décrit la procédure pour utiliser ce projet dans un environnement de développement.

- [Périmètres](#)
- [Installation](#)
- [Déploiement](#)
 - [Environnement de développement](#)
 - [Environnement de test](#)
- [Utilisation](#)
 - [Environnement de développement](#)
 - [Environnement de test](#)

Périmètres

- [Node.js](#)
- [Truffle](#)
- [Ganache](#)
- [MetaMask](#)

Installation

Les composants sont à installer dans l'ordre.

Node.js

Merci de se reporter à cette [documentation](#). L'installation de la version v13.X est recommandé, mais le projet reste compatible avec toute version supérieur ou égale à la v10.X.

Truffle

Ouvrez un terminal et exécuter la commande suivante:

```
npm install -g truffle
```

En cas de problème avec la commande précédente sur Windows, se référer à cette [documentation](#).

Ganache

Se référer à la [documentation officielle](#).

MetaMask

Se référer à la section *Installing MetaMask de cette [documentation](#).

Déploiement

Environnement de développement

1. Lancer Ganache
2. Cliquer sur le bouton **New workspace**
3. Dans l'onglet *Workspace*, indiquer:
 - dans le champ **Workspace name**: `monkey-voting`
 - dans le champ **Truffle projects**: le chemin complet vers le fichier `truffle-config.js`.
4. Ouvrir un terminal à la racine du projet et taper la commande `truffle migrate`
5. Si la commande précédente ne conduit pas à un déploiement en succès, forcer le déploiement avec la commande `truffle migrate --reset --compile-all --skip-dry-run`
6. Suivre ce [tutorial](#) pour configurer MetaMask

Environnement de test

Installation initiale

Dans un terminal, taper la commande `geth --testnet account new` afin de créer un compte dans le réseau de test

L'adresse de compte sera affichée dans la console. Il faut la conserver.

Il faut ensuite démarrer la synchronisation avec la commande suivante:

```
geth --testnet --syncmode fast --cache 1024 --rpc --rpcapi eth,net,web3,personal
```

La synchronisation va se dérouler en arrière plan.

Afin d'obtenir des `eth` pour nos tests, il faut utiliser le site <https://faucet.ropsten.be/> et taper l'adresse précédemment obtenue dans le champ associé.

Interaction avec le serveur Geth

Dans un nouveau terminal, taper la commande:

```
geth attach http://127.0.0.1:8545
```

Un nouveau prompt Javascript sera affiché. Y taper la commande:

```
eth.syncing
```

Si cette commande ne retourne pas `false`, c'est que la synchronisation est toujours en cours. Il faut attendre que la synchronisation soit terminée, et relancer la commande ultérieurement.

Une fois la synchronisation terminée, il faut déverrouiller le compte avec la commande:

```
personal.unlockAccount(eth.accounts[0])
```

Y taper la passphrase utilisée pour créer le compte.

Réaliser la migration Truffle

Dans un nouveau terminal, exécuter la commande:

```
truffle migrate --network ropsten
```

La migration va débuter et va déployer les contrats sur le réseau Ropsten. Quand elle sera terminée, l'adresse associée au contrat devra être conservée.

Cette adresse sera utilisée depuis notre Dapp.

Configuration de Swarm

Afin de déployer la webapp sur la blockchain, il faut utiliser Swarm. Dans un terminal, exécuter la commande suivante en remplaçant <ACCOUNT> par l'adresse lors de la création du compte dans le réseau Ropsten:

```
swarm --ens-api "" --bzzaccount <ACCOUNT>
```

Pour uploader la webapp, se rendre à la racine du projet [MonkeyVotingDapp](#) et exécuter la commande suivante:

```
swarm --recursive up ./
```

Un hash sera retourné. Il établi le lien à Swarm. Il faut conserver ce hash.

Utilisation

Environnement de développement

1. Lancer Ganache dans le workspace [monkey-voting](#)
2. À la racine du projet, ouvrir un terminal et taper la commande `ng serve`
3. Dans un navigateur, se rendre à l'adresse <http://localhost:4200>

Environnement de test

1. Dans un terminal taper la commande `geth attach http://127.0.0.1:8545`
2. Dans un navigateur, se rendre à l'adresse http://localhost:8500/bzz:/<swarm_upload_hash>/index.html en remplaçant <swarm_upload_hash> par le hash obtenu à la suite de l'upload avec Swarm

