



wada
Côte d'Ivoire



Ministère de l'Enseignement
Supérieur et de la
Recherche Scientifique

cncci
Centre National de Calcul de Côte d'Ivoire

Samedi 04 février 2023



Plutus on chain

DÉ CODE
TA RICHESSE
AVEC LA BLOCKCHAIN
CARDANO



Malik Koné (PhD)

Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- Code d'un smart contract
- Compilation : plutus template

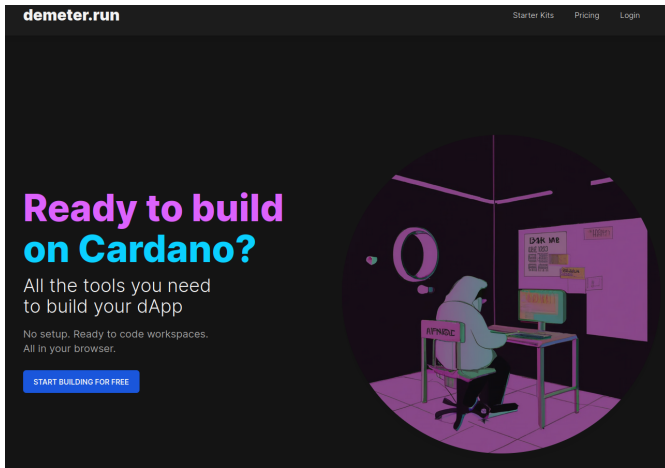
3. Tutoriel 2 : Comment compiler un script plutus et l'activer

- Tutoriel

4. Références

- Références

Créer un compte sur la Plateforme



Liens vers la plateforme -> demeter.run

Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- Code d'un smart contract
- Compilation : plutus template

3. Tutoriel 2 : Comment compiler un script plutus et l'activer

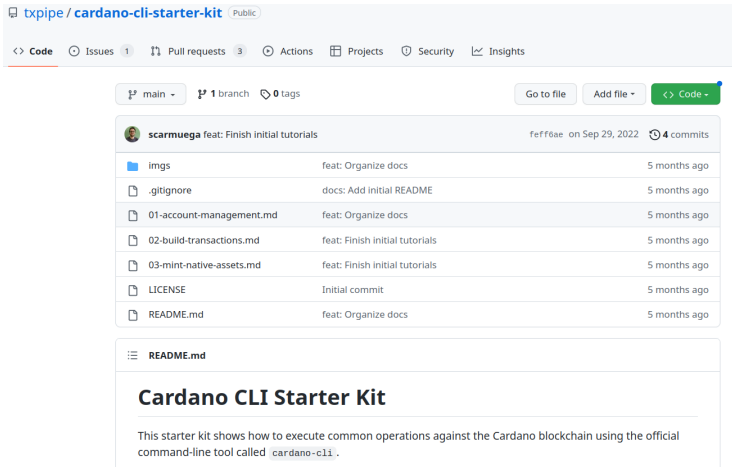
- Tutoriel

4. Références

- Références

Communiquer avec un noeud Cardano

Lien github -> [Cardano-cli.git](https://github.com/txpipe/cardano-cli-starter-kit)



txpipe / **cardano-cli-starter-kit** Public

<> Code Issues 1 Pull requests 3 Actions Projects Security Insights

main 1 branch 0 tags Go to file Add file <> Code

File	Commit Message	Commit Hash	Commit Date	Commits
imgs	feat: Organize docs	feff6ae	on Sep 29, 2022	4 commits
.gitignore	docs: Add initial README			5 months ago
01-account-management.md	feat: Organize docs			5 months ago
02-build-transactions.md	feat: Finish initial tutorials			5 months ago
03-mint-native-assets.md	feat: Finish initial tutorials			5 months ago
LICENSE	Initial commit			5 months ago
README.md	feat: Organize docs			5 months ago

☰ README.md

Cardano CLI Starter Kit

This starter kit shows how to execute common operations against the Cardano blockchain using the official command-line tool called `cardano-cli`.

Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- Code d'un smart contract
- Compilation : plutus template

3. Tutoriel 2 : Comment compiler un script plutus et l'activer

- Tutoriel

4. Références

- Références

Alice propose une enchère (1/4)



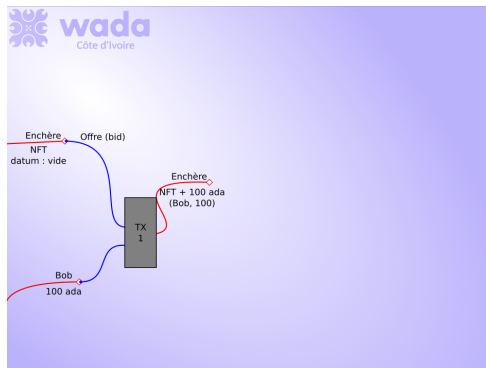
Alice fournit

- ▶ le NFT
- ▶ un datum (avec l'offre minimal)

Le smart-contract contient

- ▶ conditions de validité de l'offre (on-chain)
- ▶ logique pour fin d'enchère (off-chain)

Bob fait une offre, nouvelle transaction (2/4)



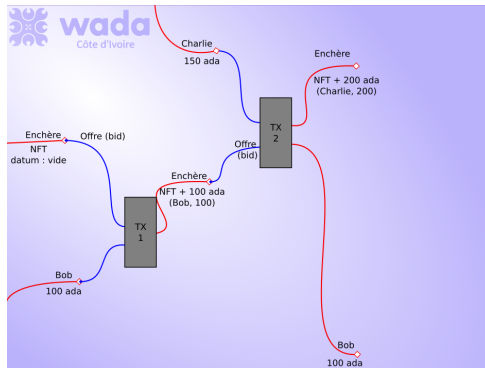
En entrée

- ▶ UTxO d'Alice
- ▶ UTxO de Bob

En sortie

- ▶ adresse du script -> NFT, datum (Bob, 100)

Charlie sur-offre (3/4)



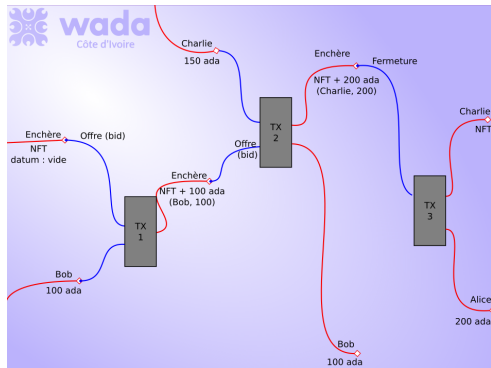
En entrée

- ▶ UTxO du script
- ▶ UTxO de Charlie

En sortie

- ▶ adresse du script -> NFT, datum (Charlie, 150)
- ▶ adresse de Bob -> 100

Fin de l'enchère (4/4)



Après un certain temps décidé par Alice ou le script

En entrée

- ▶ UTx0 du script

En sortie

- ▶ adresse Charlie -> le NFT
- ▶ adresse Alice -> le montant

Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- **Code d'un smart contract**
- Compilation : plutus template

3. Tutoriel 2 : Comment compiler un script plutus et l'activer

- Tutoriel

4. Références

- Références

Code d'un smart contract

```
{-# INLINABLE mkAuctionValidator #-}

mkAuctionValidator :: AuctionDatum -> AuctionAction -> ScriptContext -> Bool
mkAuctionValidator ad redeemer ctx =
  traceIfFalse "wrong input value" correctInputValue &&
  case redeemer of
    MkBid b@Bid{..} ->
      traceIfFalse "bid too low" (sufficientBid bBid) &&
      traceIfFalse "wrong output datum" (correctBidOutputDatum b) &&
      traceIfFalse "wrong output value" (correctBidOutputValue bBid) &&
      traceIfFalse "wrong refund" correctBidRefund &&
      traceIfFalse "too late" correctBidSlotRange
    Close ->
      traceIfFalse "too early" correctCloseSlotRange &&
  case adHighestBid ad of
    Nothing ->
      traceIfFalse "expected seller to get token" (getValue (aSeller auction) $ tokenValue <> Ada.lovelaceVal)
    Just Bid{..} ->
      traceIfFalse "expected highest bidder to get token" (getValue bBidder $ tokenValue <> Ada.lovelaceVal)
      traceIfFalse "expected seller to get highest bid" (getValue (aSeller auction) $ Ada.lovelaceVal)

where
  info :: TxInfo
  info = scriptContextTxInfo ctx
```

Voir Github du [programme pionniers de plutus](#) pionner

Survol du code

on chain

le code Haskell qui compilé, pour

- ▶ UPLC : plutus-core

off chain

Le code qui ne réside pas sur la blockchain.

- ▶ Sert à construire une transaction valide
- ▶ Se trouve généralement dans le portefeuille

Différentes parties du code

- ▶ data
- ▶ script
- ▶ compilation
- ▶ validation
- ▶ mockup code for playground

Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- Code d'un smart contract
- **Compilation : plutus template**

3. Tutoriel 2 : Comment compiler un script plutus et l'activer

- Tutoriel

4. Références

- Références

Compilation : plutus template

Partie du code qui compile en UPLC

```
typedAuctionValidator :: Scripts.TypedValidator Auctioning
typedAuctionValidator = Scripts.mkTypedValidator @Auctioning
    $$ (PlutusTx.compile [| mkAuctionValidator |])
    $$ (PlutusTx.compile [| wrap |])
where
    wrap = Scripts.wrapValidator @AuctionDatum @AuctionAction
```


Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- Code d'un smart contract
- Compilation : plutus template

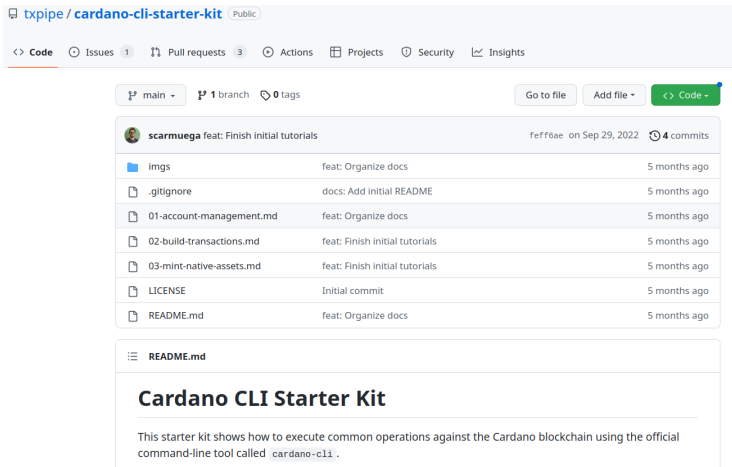
3. Tutoriel 2 : Comment compiler un script plutus et l'activer

- Tutoriel

4. Références

- Références

Comment compiler un script plutus et l'activer



txpipe / **cardano-cli-starter-kit** Public

<> Code Issues 1 Pull requests 3 Actions Projects Security Insights

main 1 branch 0 tags Go to file Add file <> Code +

scarmuega feat: Finish initial tutorials feff6ae on Sep 29, 2022 4 commits

imgs	feat: Organize docs	5 months ago
.gitignore	docs: Add initial README	5 months ago
01-account-management.md	feat: Organize docs	5 months ago
02-build-transactions.md	feat: Finish initial tutorials	5 months ago
03-mint-native-assets.md	feat: Finish initial tutorials	5 months ago
LICENSE	Initial commit	5 months ago
README.md	feat: Organize docs	5 months ago

README.md

Cardano CLI Starter Kit

This starter kit shows how to execute common operations against the Cardano blockchain using the official command-line tool called `cardano-cli`.

Lien github -> [plutus-start-kit.git](https://github.com/txpipe/plutus-start-kit)

Sommaire

1. Tutoriel 1 : Communiquer avec un noeud Cardano

- Mise en place
- Tutoriel

2. Script Plutus : Une introduction à Haskell

- Enchère à l'anglaise
- Code d'un smart contract
- Compilation : plutus template

3. Tutoriel 2 : Comment compiler un script plutus et l'activer

- Tutoriel

4. Références

- Références

Liens des références

IOG Academy

Lien -> [Vidéos et ressources devenir expert en Marlow, Plutus et Haskell](#)

Plutus pioneer programm

Lien -> [Plutus pioneer program](#) Lars Brünjes

Tutoriels Demeters

- ▶ [Github de txpipe](#)

D'autre tutos

- ▶ <https://github.com/iburzynski/emurgo-plutus-starter>
- ▶ <https://github.com/input-output-hk/Alonzo-testnet>