

Formation Tezos - Apprendre à développer sur Tezos

Description de la formation Tezos

Française et publique de son état, la **blockchain Tezos** se distingue des précédentes générations (Bitcoin, Ethereum) par trois caractéristiques principales : la gouvernance permettant l'auto-évaluation du protocole, le consensus Liquid Proof Of Stake et le développement des smart contracts avec le langage Michelson autorisant la vérification formelle.

Tezos a par ailleurs été écrit dans le langage fonctionnel OCaml, réputé pour sa sécurité grâce à sa capacité à écrire des programmes informatiques proches des formules de la logique mathématique.

À travers des exercices concrets, cette **formation Tezos** vous propose de préparer un environnement puis d'apprendre à faire tourner un nœud de blockchain, et de développer, tester, déployer et interagir avec des smart contracts. La formation vous permettra aussi de mieux comprendre Ligo et le langage Michelson.

Objectifs

Objectifs opérationnels :

Savoir développer des smart contracts sur la Blockchain Tezos.

Savoir construire des architectures de contrats répondant aux divers besoins fonctionnels.

Objectifs pédagogiques :

À l'issue de cette **formation Tezos**, vous serez à même de :

- Appréhender les concepts de la Blockchain Tezos
- Modéliser et créer un smart contract en Ligo
- Mettre en place les tests unitaires sur un smart contract
- Déployer un smart contract avec Michelson
- Interagir avec un smart contract déployé
- S'approprier des patterns simples de smart contracts

À qui s'adresse cette formation ?

Public :

Ce cours Tezos s'adresse aux développeurs, architectes et ingénieurs concepteurs.

Prérequis :

Pour suivre cette formation Tezos, il est nécessaire d'avoir des connaissances de l'environnement Linux (ssh, bash) et de connaître au moins un langage de programmation.

Il est également demandé d'avoir suivi la formation La Blockchain - État de l'art et cas concrets (DBEA) ou de posséder les connaissances équivalentes.

Contenu du cours Tezos

Jour 1

Blockchain : les bases de la décentralisation

Introduction

Consensus

Historique

Blockchain Tezos

Historique

Architecture



ITgate
Training

Your Gateway to Excellence

Enjeux

Gouvernance en chaîne

Discussion vérification formelle

Blockchain Tezos : interactions

Installation

Lancement d'un noeud

Outil de ligne de commande

Interactions par http

Mise en pratique

Noeud sandbox et transactions

Michelson

Smart contract (paramètres, storage, code)

Langage et commandes

Point d'accès

Typage

Mise en pratique

Exécution d'un exemple simple

Jour 2

Ligolang (Pascaligo)

Les base de la programmation fonctionnelle

Langage

Storage

Entrypoint

Transaction

Outil transpilation

Mise en pratique

Counter smart contract

Simulation d'un smart contract Ligo

Tests unitaires

Mise en place / Utilisation pyTezos

Paramètres d'appel et storage

Gestion des exceptions

Mise en pratique

Counter smart contract

Déploiement de smart contract

Préparation des paramètres et storage

Comptes et faucet

Simulation du déploiement et gaz

Déploiement / baking

Mise en pratique

Premier déploiement

Invocation de smart contract

Préparation des paramètres d'invocation

Exécution

Mise en pratique

Invocation et accès au storage

Jour 3

Ligolang (deuxième partie)

Bonnes pratiques

Mise en pratique

Smart contrat de Vote

Fonction anonyme

Introduction à la vérification formelle avec Tezos

Coq et Mi-cho-coq

Démarche et modélisation

Exemple Vote

Introduction à la librairie SmartPy

Présentation de SmartPy

Exemple de code

Jour 4

Pattern de smart contract

Mise en pratique

Multisig

Mise en pratique

Proxy

Interactions entre smart contracts et « polymorphism »

Travaux Pratiques

Les travaux pratiques représentent 70% du temps de formation.