+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 01/07/2024

S'inscrire

Formation Solidity

4 jours (28 heures)

Présentation

Solidity est un langage de programmation orienté objet pour écrire des Smart Contracts. Il est utilisé pour mettre en œuvre des contrats intelligents sur différentes plates-formes basées sur la blockchain telles que Ethereum. Il est basé sur la syntaxe ECMAScript, ce qui en fait un choix familier pour les développeurs Web.

Ethereum est une plate-forme de développement d'applications décentralisées (les dApps), basée sur la technologie Blockchain, sans aucun risque d'interruptions, fraudes et intrusions. Grâce à Ethereum, il est désormais possible de programmer toute une gamme d'applications où il est habituellement nécessaire d'avoir un tiers de confiance (crowdfunding, vote, organisations, cadastre…).

Dans le cadre de cette formation, les participants apprendront à rédiger un Smart Contracts à l'aide de Solidity.

Comme dans toutes nos formations, celle-ci vous présentera la toute dernière version à savoir Solidity 0.8.

Objectifs

- Comprendre les fondements du développement de contrats intelligents sur la blockchain Ethereum
- Acquérir des compétences en programmation en Solidity pour la création de contrats intelligents
- Maîtriser les outils et technologies essentiels tels que Hardhat, OpenZeppelin et Ethers.js
- Apprendre à concevoir et à implémenter des contrats intelligents pour des cas d'utilisation réels
- Acquérir les compétences nécessaires pour développer des applications décentralisées (DAPP) sur Ethereum
- Savoir écrire des tests unitaires pour garantir la robustesse et la fiabilité des contrats intelligents
- Explorer les bonnes pratiques de sécurité et les stratégies de déploiement sur la blockchain
- Gagner en confiance pour créer des solutions blockchain innovantes et fonctionnelles

Public visé

- Développeurs
- Architectes
- Ingénieurs concepteurs

Pré-requis

- Au moins 6 mois d'expérience en programmation
- La connaissance de javascript et/ou de React est un plus
- Tester Mes Connaissances

Pré-requis techniques

- Node.js installé
- Un éditeur de code (type visual studio Code)

Pour aller plus loin

Si vous voulez en savoir plus sur les nouveaux enjeux de demain et la décentralisation du web actuel, notre formation Web3 peut vous intéresser.

Programme de la formation Solidity

Jour 1 - Introduction à la blockchain et les bases des Smart Contracts

Les concepts de base de la blockchain Ethereum

- Le réseau et la monnaie-Ethereum et l'Ether (ETH)
- Les différents types des réseaux Ethereum
- Les nœuds d'archives, complets et légers
- Se connecter aux réseaux Ethereum
- Comprendre les différences entre ETH, WEI et GAZ
- Comprendre et calculer les frais d’une transaction
- Les blocs et la blockchain Ethereum
- Les fonctions de hachage
- Les 2 types de comptes sur Ethereum
- La différence entre une transaction et un appel de message
- Les différents mécanismes de consensus sur la blockchain -PoW et PoS
- "The Merge" et la chaîne Beacon

Les contrats intelligents ("Smart Contracts")

- Les contrats intelligents : que sont-ils, comment fonctionnent-ils ?

- Anatomie et conception d'un contrat intelligent
- Compilation, déploiement et débogage des contrats intelligents sur Remix
- La configuration d’un environnement de développement locale avec Hardhat
- Compilation, test et déploiement des contrats intelligents avec Hardhat
- Utilisation de Ganache pour le déploiement local
- Projet avec exercices : développer, compiler, déboguer et déployer votre premièrecontrat intelligent sur Remix et Hardhat

JOUR 2 – SOLIDITY

Le langage Solidity

- Syntaxe et notions de base
- Sujets avancés
- Exemples et exercices de codage pour chaque thème
- Projet avec exercices : création d'un contrat de vote en utilisant la plupart des sujetsdiscutés précédemment

JOUR 3 – CRÉATION D'UNE DAPP - OPENZEPPELIN ET LES JETONS ERC20

Les applications décentralisées - DApps

- Installation et configuration du portefeuille Metamask
- L'API RPC de Metamask
- Utilisation de la bibliothèque Ethers.js - Provider, Signer, Contract et Utils
- Utilisation d’Alchemy (fournisseur tiers des nœuds Ethereum)
- Projet avec exercices : création d'une DApp en utilisant Ethers.js et l'API RPC deMetamask

OpenZeppelin

- Comment utiliser les contrats modulaires et réutilisables d'OpenZeppelin
- Minimiser les risques en utilisant les contrats d'OpenZeppelin
- Les différents types de contrats d'OpenZeppelin : jetons, contrôle d’accès,gouvernance, sécurité...

Les jetons ERC20

- Compréhension de la norme ERC20
- Création d'un jeton ERC20 (en implémentant toutes les méthodes et événementsdéfinis par la norme ERC20)
- Déploiement du contrat sur le réseau de test Goerli
- Test des méthodes et des événements en utilisant Ethers.js

JOUR 4 – LES TEST UNITAIRES ET PROJET DAPP NFT

Les tests unitaires sur les contrats intelligents

- Utilisation du framework Mocha avec Chai pour tester des contrats intelligents
- Utilisation de la bibliothèque d’assertion Hardhat-Chai pour évaluer des transactionsrenversées, des événements, le changement du solde d'Ether...
- Rendre les tests plus efficaces avec les "Hardhat Network Helpers"

Les NFT's

- Compréhension des NFT's et de la norme technique ERC721
- Différence entre ERC721 et ERC1155
- Création d'un jeton ERC721 en utilisant l'extension ERC721URIStorage
- Utilisation d'IPFS et Pinata pour stocker l'image et les métadonnées du NFT
- Déploiement du contrat sur le réseau de test Goerli et affichage du NFT sur OpenSea
- Création d'un DAPP (en React) permettant de créer des NFT's avec des paramètrespersonnalisés
- Utilisation du portefeuille Metamask (via l'API RPC) avec le DAPP

CONTENU FACULTATIF (+1 jour)

Les attaques de réentrée

- Protection d'un contrat intelligent contre les attaques de réentrée
- Modèle de conception : "Checks-Effects-Interaction"
- Création d'un contrat vulnérable (Banque) et d'un contrat attaquant
- Simulation d'une attaque de réentrée avec un script utilisant Ethers.js
- Amélioration du contrat vulnérable pour prévenir les attaques de réentrée

Création d'une DAO - Organisation Autonome Décentralisée

- Compréhension du fonctionnement d'une DAO et de la création et de l'exécution despropositions
- Création d'une DAO avec différents contrats de la bibliothèque OpenZeppelin :ERC20Votes, TimelockController, Governor...
- Déploiement de la DAO sur le réseau local Hardhat
- Création et exécution des propositions sur la DAO en utilisant Ethers.js

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son

inscription définitive, l'apprenant reçoit un questionnaire d’auto-évaluation nous permettant d’apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d’anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l’acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.