



ERC20 Token

Ethereum Blockchain

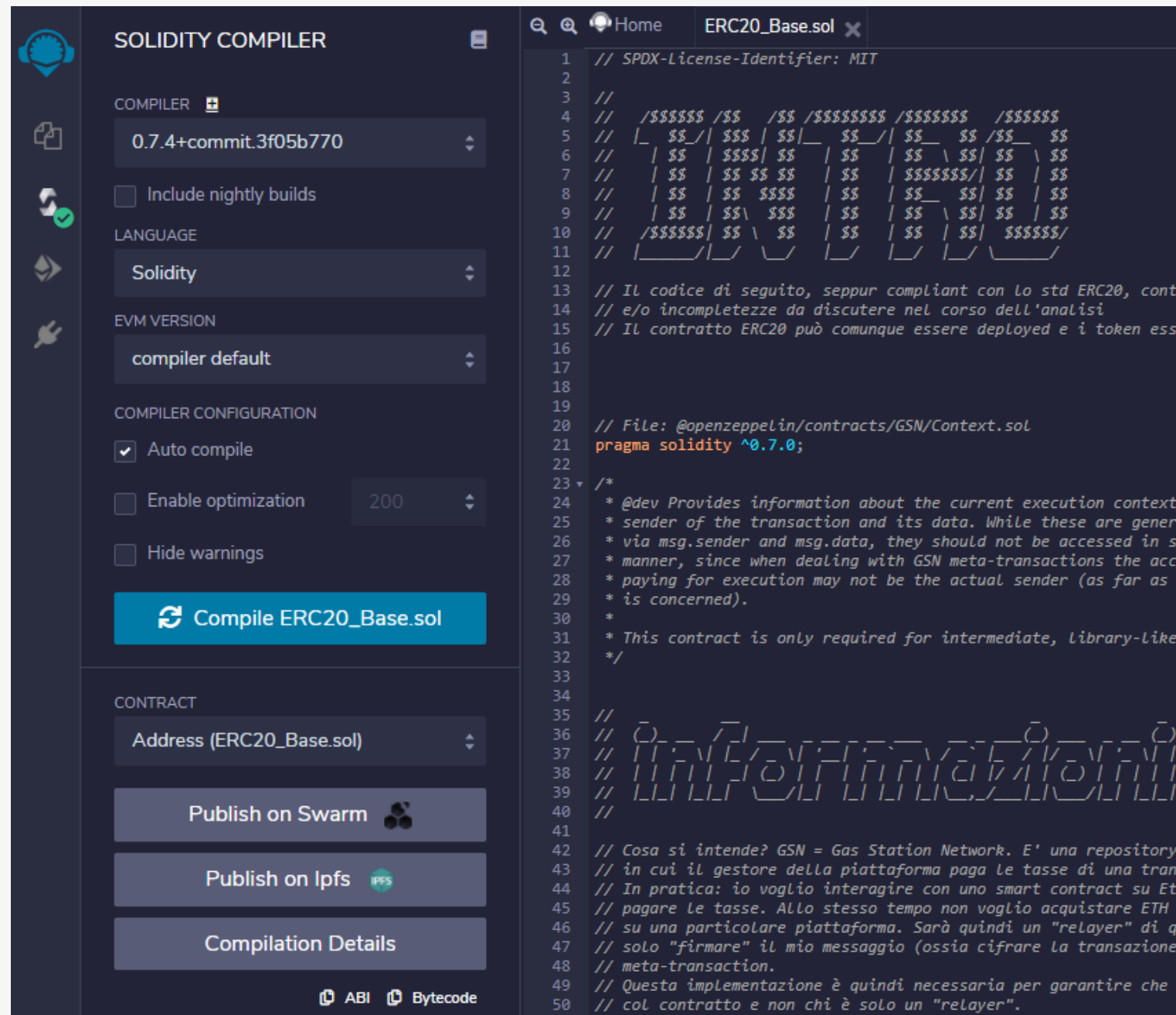
16 FEBBRAIO 2021

01

Corso Token ERC20



DEPLOYMENT SU TESTNET/MAINNET



1. Compilare il contratto

Una volta settata la corretta versione del compiler, in contratto viene automaticamente controllato e compilato (spunta verde sulla destra).

Eventuali warning saranno riportati con la spunta gialla (il codice è comunque compilato).

Gli errori invece che impediscono la compilazione sono settati in rosso.

DEPLOYMENT SU TESTNET/MAINNET

2. Connessione al testnet/mainnet e deployment

Impostando "Injected Web3, si utilizza la libreria Java in grado di comunicare con la dApp wallet per poter effettuare il deployment sul network a cui il wallet è connesso.

Bisognerà impostare il corretto gas limit a seconda della dimensione del contratto di cui fare il deployment, selezionare il contratto appropriato e inserire i vari argomenti richiesti dal constructor.

A questo punto di potrà effettuare il deployment pagando il gas necessario per l'operazione.

The image shows a web interface for deploying a contract. The left sidebar contains the 'DEPLOY & RUN TRANSACTIONS' section with the following settings:

- ENVIRONMENT:** Injected Web3
- ACCOUNT:** 0xE5a...631A7 (0.8768237 ETH)
- GAS LIMIT:** 3000000
- VALUE:** 0 wei
- CONTRACT:** MYERC20 - browser/ERC20_Base.sol

The main area displays the Solidity code for 'ERC20_Base.sol'. The code includes a license identifier, a pragma statement for Solidity ^0.7.0, and a constructor function.

An overlay window shows the 'Ropsten Test Network' status, the account address 'Test_04 0xE5a0...31A7', and the balance '0.8768 ETH'. It also features 'BUY' and 'SEND' buttons.

DEPLOYMENT SU TESTNET/MAINNET

[illegible]

3. Validazione tramite Etherscan

Una volta effettuato il deployment, il contratto e le relative transazioni saranno visibili su Etherscan, tuttavia, non essendo ancora stato validato come token ERC20, non si potrà interagire con esso.

Bisognerà selezionare Verify and Publish per validare il contratto e renderlo pubblico.

Per effettuare queste operazioni, sarà necessario fornire una serie di informazioni sul compiler e sullo stesso codice.

DEPLOYMENT SU TESTNET/MAINNET

The screenshot shows the HashEx website interface. At the top, there's a navigation bar with 'HashEx' on the left and links for 'What is this?', 'How it works?', and 'Contacts' on the right. Below the navigation bar, a heading reads 'Enter your contract's ABI to auto-parse'. Underneath is a large text area with a light blue border containing the JSON ABI: `[{"inputs": [], "name": [], "type": "function"}]`. To the right of the text area is a 'Parse' button. Below the text area, another heading reads 'Or enter your parameters manually'. This section includes a 'Function type' dropdown menu with 'constructor' selected, an 'Argument' dropdown menu with 'String' selected, and a text input field with the placeholder 'Enter arguments without quotes, arrays enter like [1, 2, 3]'. At the bottom of this section are two buttons: 'Add parameter' and 'Remove'.

4. ABI Input

L'elemento principale da fornire sarà l'ABI del contratto selezionato, facendo attenzione a fornirla con tutti gli argomenti inseriti nel constructor al deployment, pena il fallimento della validazione.

Il codice ABI codificato, può essere ottenuto tramite l'utility online

<https://abi.hashex.org/>

Una volta forniti tutti i dati, il contratto sarà validato e su Etherscan risulterà come

The screenshot shows the Etherscan interface for a contract. At the top, there's a navigation bar with tabs for 'Transactions', 'Erc20 Token Txns', 'Contract' (which is selected and has a green checkmark), 'Events', 'Analytics', and 'Comments'. Below the tabs, there are three buttons: 'Code', 'Read Contract', and 'Write Contract'. Below these buttons, a green checkmark icon is followed by the text 'Contract Source Code Verified (Exact Match)'.

E si potrà interagire con esso.