

ORIONIX TOKEN TECHNICAL SPECIFICATION

ORX is a custom built token built with Ethereum smart contracts which adheres to the ERC20 standard interface. The smart contracts were developed using the latest security updates by Solidity and written in version 0.5.8 of the Solidity programming language.

The functionalities which the ORX token supports include: a **transfer** function which allows users to transfer tokens to others given that each user has enough funds in their accounts. The token also has an **approve** function which allows token holders to approve others to spend up to a certain amount of tokens on their behalf. To make use of approvals, the token has a **transfer from** function implemented which allows those who have been approved to spend tokens on behalf of other token holders to move tokens from their accounts (so long as they do not exceed the allowance which was granted by the token owners).

The above mentioned functions are part of the standard ERC20 interface. However, there also exists a **burn** function which allows only the owner of the token contract to burn tokens from the token supply, directly from his own personal balance. Therefore, it is not possible for the owner to burn tokens from the accounts of others.

Wallets which support the ORX token.

To store the ORX token, any Ethereum wallet is compatible as long as you possess the private key of the wallet. For this reason you should not attempt to store any tokens in a cryptocurrency exchange wallet unless the ORX token is listed there. If you do send tokens to an exchange wallet and you do not have the private key, you may permanently lose access to those tokens. For this reason, it is also strongly advised that you do **not** send ETH to the ICO contract address from an exchange because the tokens will automatically be sent to the address from which the ETH originated.

Here are the recommended wallets you can use:

- Trust Wallet – (<https://trustwallet.com>) - **ORX has been officially accepted by Trust Wallet**
- MyEtherWallet (<https://myetherwallet.com>)