

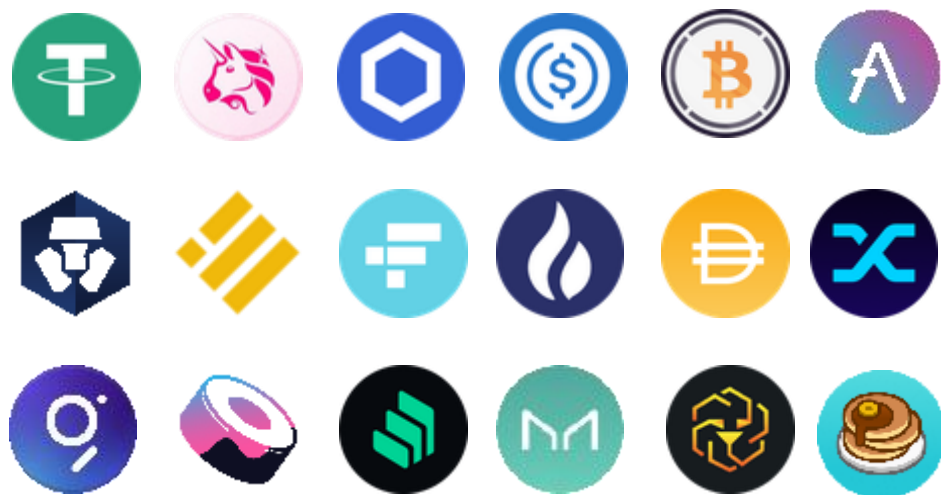
Workshop Ethereum Tokens

Sheets

<https://web3examples.com/Saxion>

Fungible tokens (ERC20)

ERC20



ERC20 generator



ERC20 Token Generator | Create your Token for FREE

Easily deploy Smart Contract for a Standard, Capped, Mintable, Burnable ERC20 Token.

Token name * Token symbol * Token decimals *

Your token name Your token symbol Your token decimals

Choose a name for your token. Choose a symbol for your token (usually 3-4 chars). Insert the decimal precision of your token. If you don't know what to insert, use 18.

Token cap * Token initial balance * Network *

Your token cap Your token initial balance Main Ethereum Network

Insert the maximum number of tokens available. Insert the initial number of tokens available. Will be put in your account. Choose your Network.

Enable transfer

Enable transfer during deploy

Choose to enable transfer during deploy or enable manually later.

Create Token

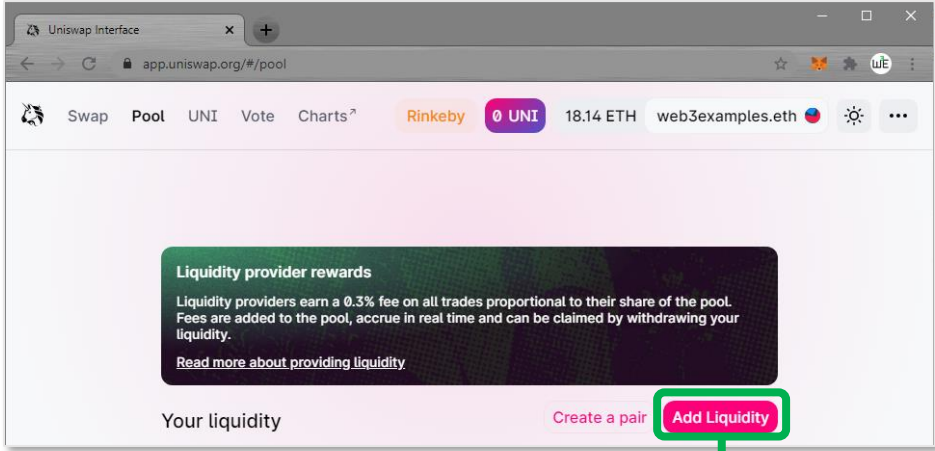
<https://vittominacori.github.io/erc20-generator>

http://web3examples.com/ethereum/demo/Create_ERC20_via_web.html

Deploy token contract (on Rinkeby)

```
ERC20Remix.sol x
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4 import "https://github.com/OpenZeppelin/openzeppelin-contracts/contracts/token/ERC20/ERC20.sol";
5
6 contract TestToken is ERC20 {
7
8     /// @dev Constructor that gives _msgSender() all of existing tokens.
9     constructor () ERC20("Web3Examples", "W3ETST") { // max 6 chars
10         _mint(msg.sender, 10000 * (10 ** 18));
11     }
12 }
```

Add liquidity on Uniswap (Rinkeby)



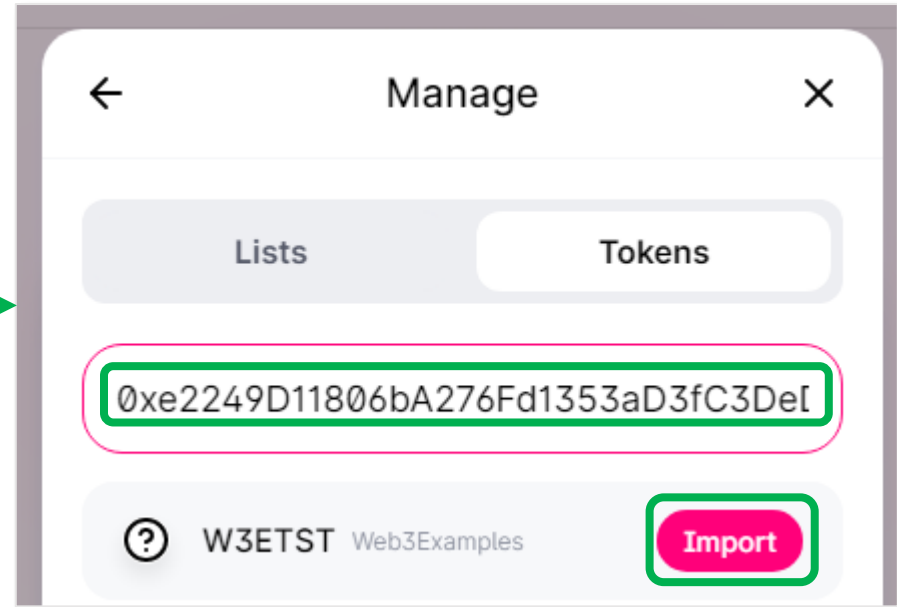
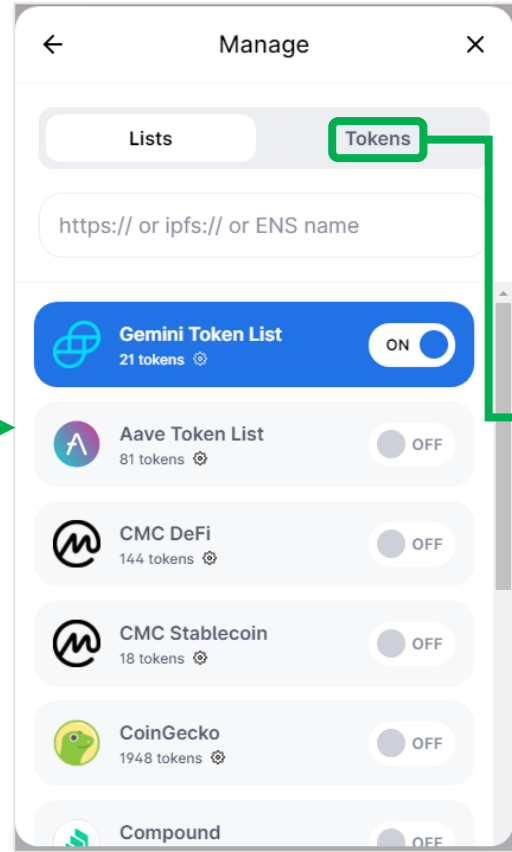
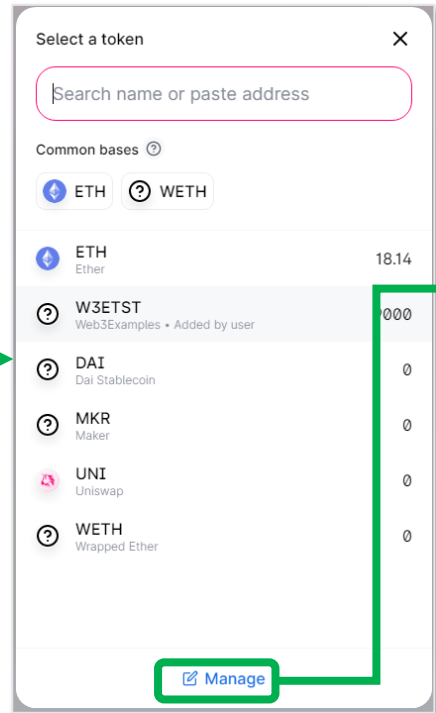
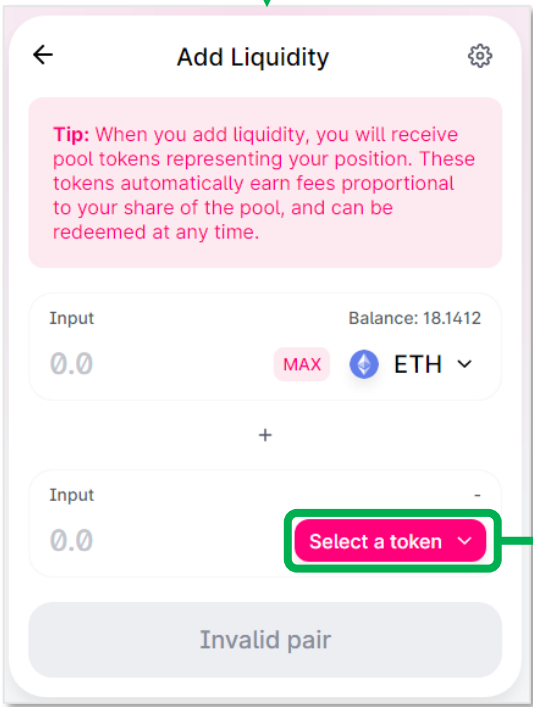
Token address

0xe2249D11806bA276Fd1353aD3fC3DeD714A20B88

<https://rinkeby.etherscan.io/address/0xe2249D11806bA276Fd1353aD3fC3DeD714A20B88>

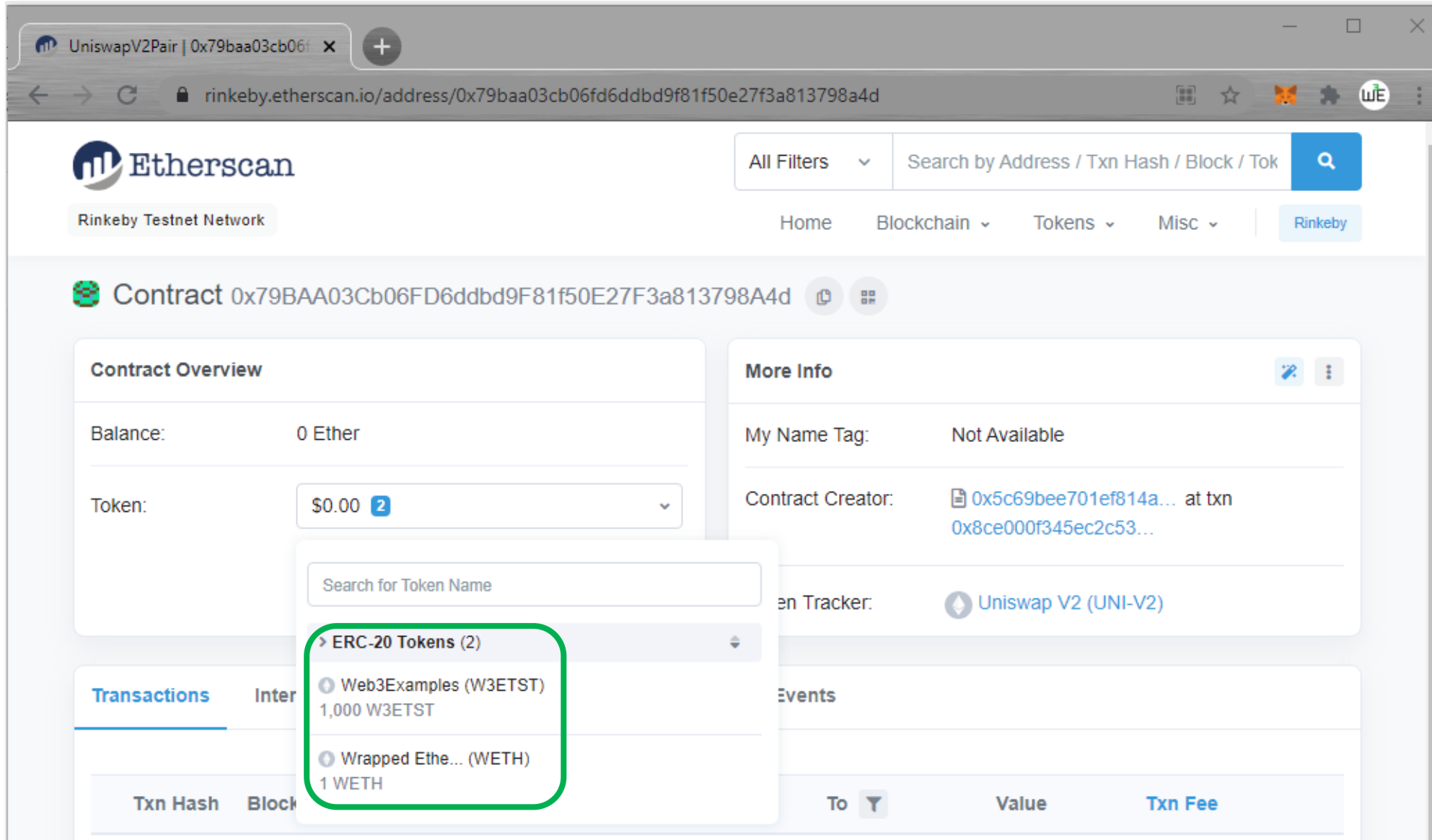
Add liquidity

<https://app.uniswap.org/#/add/ETH/0xe2249D11806bA276Fd1353aD3fC3DeD714A20B88>



<https://app.uniswap.org/#/pool>

Uniswap contract that contains liquidity



UniswapV2Pair | 0x79baa03cb06f

rinkeby.etherscan.io/address/0x79baa03cb06fd6ddbd9f81f50e27f3a813798a4d

Etherscan

Rinkeby Testnet Network

All Filters Search by Address / Txn Hash / Block / Tok

Home Blockchain Tokens Misc Rinkeby

Contract 0x79BAA03Cb06FD6dDbd9F81f50E27F3a813798A4d

Contract Overview

Balance: 0 Ether

Token: \$0.00 2

Search for Token Name

> ERC-20 Tokens (2)

- Web3Examples (W3ETST)
1,000 W3ETST
- Wrapped Ethe... (WETH)
1 WETH

More Info

My Name Tag: Not Available

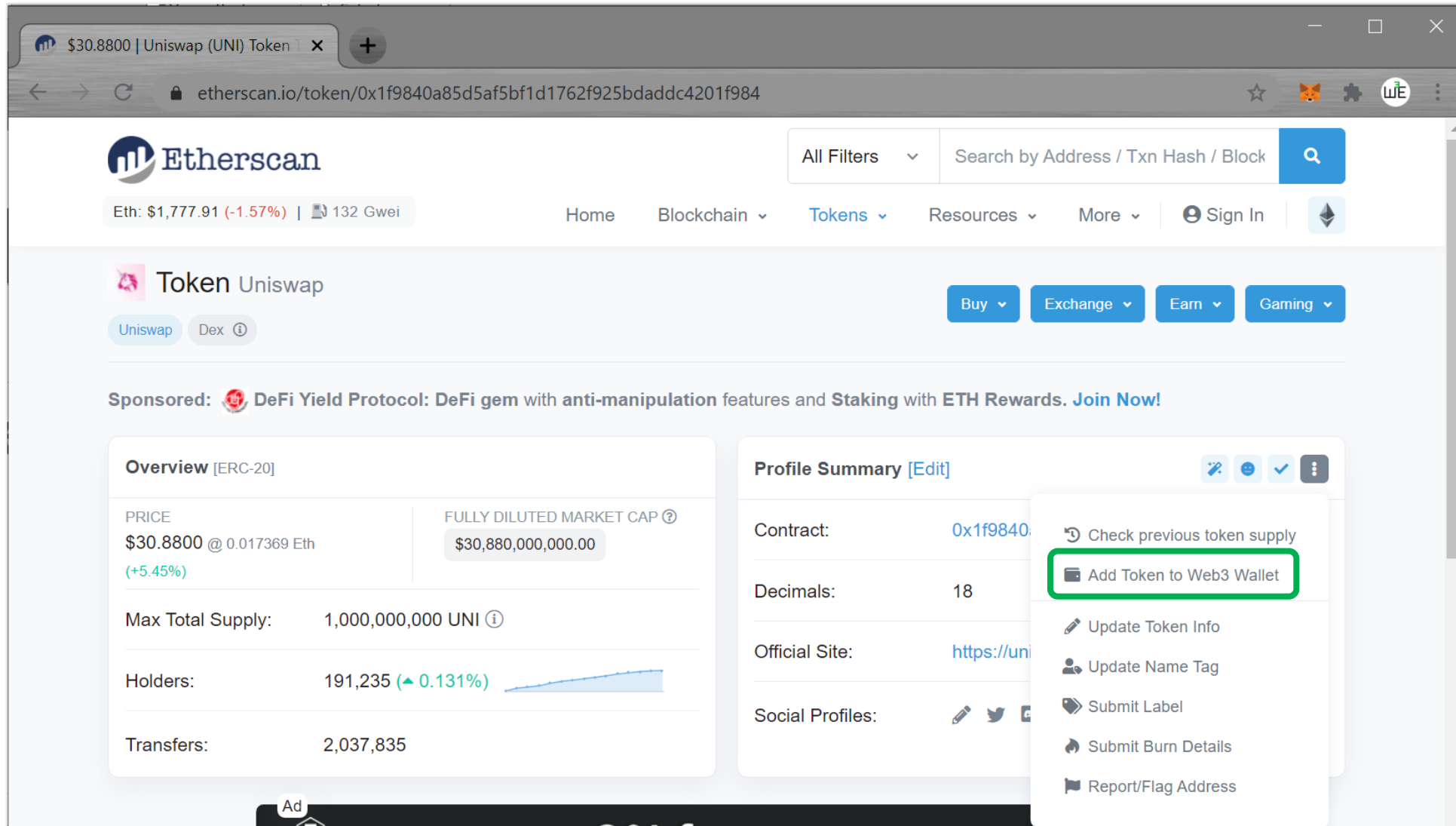
Contract Creator: 0x5c69bee701ef814a... at txn 0x8ce000f345ec2c53...

Contract Name: Uniswap V2 (UNI-V2)

Transactions

Txn Hash	Block	To	Value	Txn Fee
----------	-------	----	-------	---------

Add token in MetaMask (via etherscan)



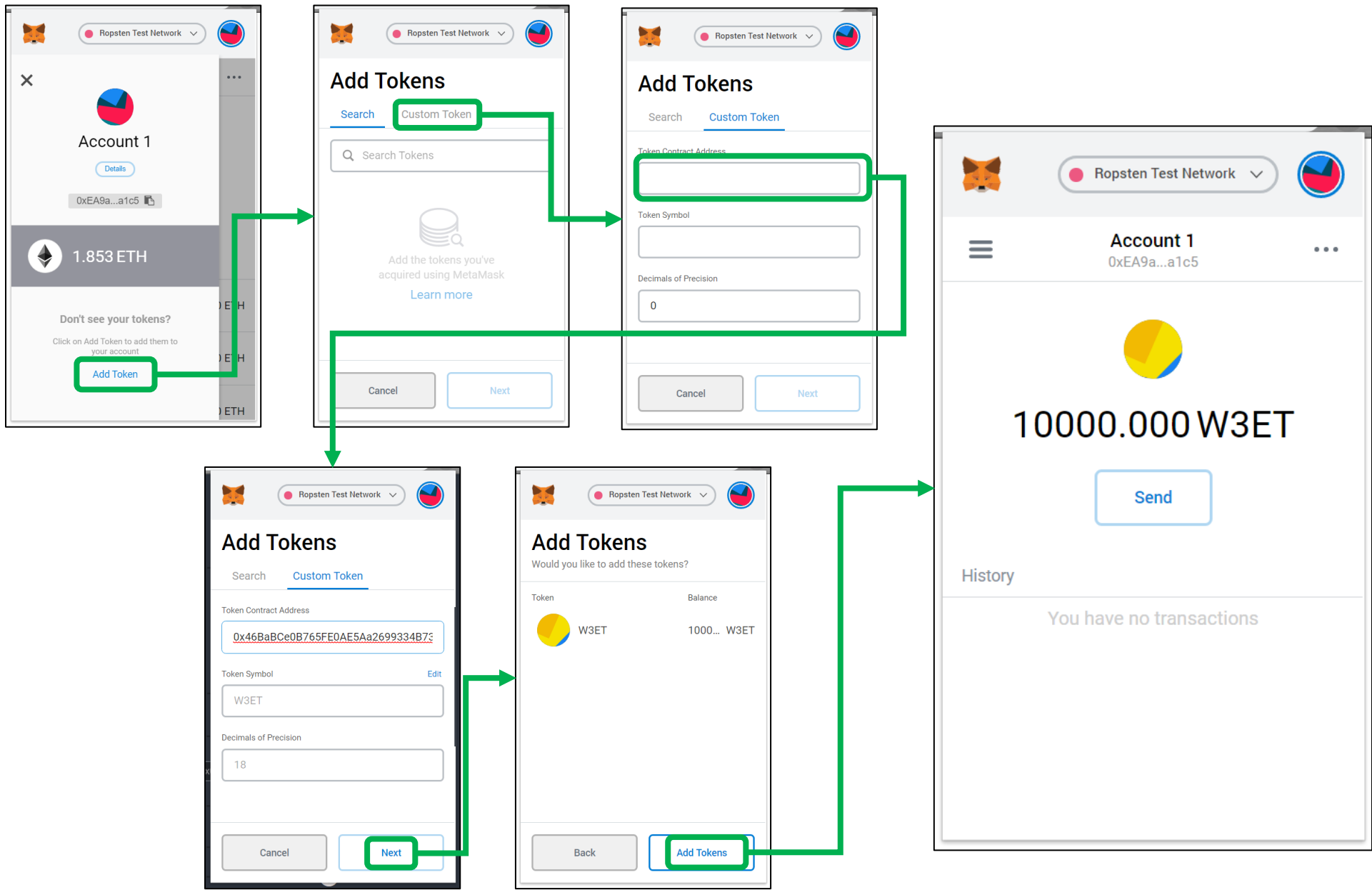
The screenshot shows the Etherscan website interface for the Uniswap token. The browser address bar displays the URL: etherscan.io/token/0x1f9840a85d5af5bf1d1762f925bdaddc4201f984. The page title is "Token Uniswap". The current price is \$30.8800, with a 5.45% increase. The fully diluted market cap is \$30,880,000,000.00. The max total supply is 1,000,000,000 UNI, and there are 191,235 holders. A dropdown menu is open over the "Profile Summary" section, with the option "Add Token to Web3 Wallet" highlighted in green. Other options in the menu include "Check previous token supply", "Update Token Info", "Update Name Tag", "Submit Label", "Submit Burn Details", and "Report/Flag Address".

Overview [ERC-20]	
PRICE	FULLY DILUTED MARKET CAP [?]
\$30.8800 @ 0.017369 Eth (+5.45%)	\$30,880,000,000.00
Max Total Supply:	1,000,000,000 UNI ⁱ
Holders:	191,235 (▲ 0.131%)
Transfers:	2,037,835

Profile Summary [Edit]	
Contract:	0x1f9840a85d5af5bf1d1762f925bdaddc4201f984
Decimals:	18
Official Site:	https://uniswap.org
Social Profiles:	Twitter Telegram Facebook

- Check previous token supply
- Add Token to Web3 Wallet**
- Update Token Info
- Update Name Tag
- Submit Label
- Submit Burn Details
- Report/Flag Address

Add token in MetaMask (direct)



Non fungible tokens (NFT) (ERC721)



Records

3LAU
<https://3lau.com>

NFT
Collateralized
loans


<https://nftfi.com>


ERC20
fractions
of NFTs

<https://niftex.com>

Market
places











 <https://niftygateway.com>

 <https://rarible.com>

 <https://opensea.io>

Art

Collectables


1		CRYPTOPUNKS
2		SORARE
3		SUPERRARE
4		HASHMASKS
5		ART BLOCKS
6		CRYPTOKITTIES
7		MAKERSPLACE
8		AXIE INFINITY
9		DECENTRALAND
10		KNOWN ORIGIN


Fan tokens
(social)

Identities

Authorisation

<https://nonfungible.com>

 <https://v.cent.co>

 <https://nbatopshot.com>

NFT Solidity code



```
ERC721RemixOZ2.sol
1  // SPDX-License-Identifier: MIT
2  // Store the NFT json files at https://jsonkeeper.com/ and provide the suffix as input for CreateNFT
3  pragma solidity ^0.8.0;
4  import "https://github.com/OpenZeppelin/openzeppelin-contracts/contracts/token/ERC721/ERC721.sol";
5
6  contract TestNFT is ERC721 {
7
8      constructor() ERC721("TestNFT", "TNFT") {
9          CreateNFT(msg.sender, "b/UR59");
10     }
11
12     function tokenURI(uint256 tokenId) public pure override returns (string memory) {
13         bytes memory ba = new bytes(32);
14         uint id=tokenId;
15         for (uint i=0;i<32;i++) { // convert number to string
16             ba[i]=bytes1(uint8(id % 256)); // get 1 byte
17             id = (id>>8); // next byte
18         }
19         string memory s=string(ba);
20         return string(abi.encodePacked("https://jsonkeeper.com/", s));
21     }
22
23     event NewNFT(string, uint);
24     function CreateNFT(address tokenholder, string memory str) public returns (uint256) {
25         bytes memory b = bytes(str);
26         uint newItemId=0;
27         for (uint i=b.length;i>0;i--) // reverse order, convert string to number
28             newItemId = (newItemId<<8) + uint(uint8(b[i-1]));
29         _mint(tokenholder, newItemId);
30         emit NewNFT(str, newItemId);
31         return newItemId;
32     }
33 }
34 // example via: https://jsonkeeper.com/b/UR59
35 // https://testnets.opensea.io/assets/0x9bfbaEa68d831cFEC17a64098246864Eb726Ba7C/62901177364322
```

ERC721 Token Functions

```
function _mint(address to, uint256 tokenId) public {
    require(to != address(0), "ERC721: invalid receiver");
    require(!_exists(tokenId), "ERC721: token already minted");
    _beforeMint(to, tokenId);
    _mint(to, tokenId);
}

function _burn(uint256 tokenId) public {
    require(!_exists(tokenId), "ERC721: token doesn't exist");
    _beforeBurn(tokenId);
    _burn(tokenId);
}

function _transfer(address from, address to, uint256 tokenId) public {
    require(_exists(tokenId), "ERC721: invalid token ID");
    require(from == msg.sender || _isApprovedOrOwner(msg.sender, tokenId), "ERC721: caller is not token owner or approved");
    _beforeTransfer(from, to, tokenId);
    _transfer(from, to, tokenId);
}

function _approve(address owner, address operator, uint256 tokenId) public {
    require(_exists(tokenId), "ERC721: invalid token ID");
    require(owner == msg.sender || _isApprovedOrOwner(msg.sender, tokenId), "ERC721: caller is not token owner or approved");
    _approve(owner, operator, tokenId);
}

function _isApprovedOrOwner(address caller, uint256 tokenId) public view returns (bool) {
    return caller == owner[tokenId] || _isApproved(caller, tokenId);
}

function _isApproved(address caller, uint256 tokenId) public view returns (bool) {
    return _isApprovedForAll(owner[tokenId], caller);
}

function _beforeMint(address to, uint256 tokenId) internal {
    _mint(to, tokenId);
}

function _beforeBurn(uint256 tokenId) internal {
    _burn(tokenId);
}

function _beforeTransfer(address from, address to, uint256 tokenId) internal {
    _transfer(from, to, tokenId);
}

function _beforeApprove(address owner, address operator, uint256 tokenId) internal {
    _approve(owner, operator, tokenId);
}

function _mint(address to, uint256 tokenId) internal {
    _mint(to, tokenId);
}

function _burn(uint256 tokenId) internal {
    _burn(tokenId);
}

function _transfer(address from, address to, uint256 tokenId) internal {
    _transfer(from, to, tokenId);
}

function _approve(address owner, address operator, uint256 tokenId) internal {
    _approve(owner, operator, tokenId);
}

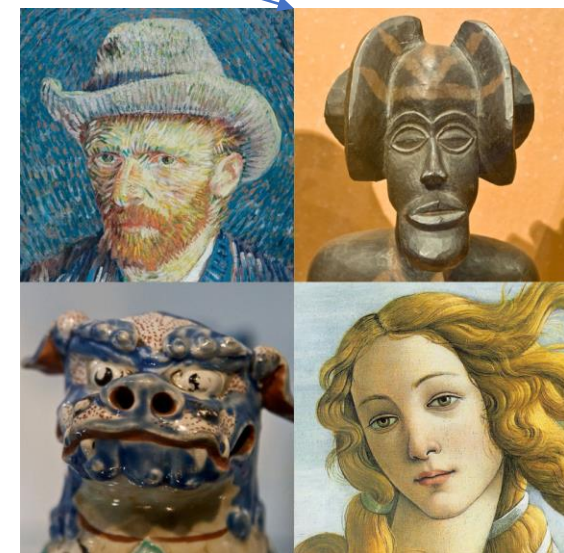
function _isApprovedOrOwner(address caller, uint256 tokenId) internal view returns (bool) {
    return caller == owner[tokenId] || _isApproved(caller, tokenId);
}

function _isApproved(address caller, uint256 tokenId) internal view returns (bool) {
    return _isApprovedForAll(owner[tokenId], caller);
}
```

<https://jsonkeeper.com>

<https://jsonkeeper.com/b/UR59>

```
{
  "description": "Wikipedia art",
  "external_url": "https://nl.wikipedia.org/wiki/Kunst",
  "image": "https://upload.wikimedia.org/wikipedia/commons/3/34/Art-portrait-collage_2.jpg",
  "name": "Wikipedia art"
}
```



https://upload.wikimedia.org/wikipedia/commons/3/34/Art-portrait-collage_2.jpg

On OpenSea



A screenshot of a web browser displaying an OpenSea NFT listing. The browser's address bar shows the URL: testnets.opensea.io/assets/0x9bfbaEa68d831cFEC17a64098246864Eb726Ba7C/62901177364322. The OpenSea interface includes a search bar, navigation links (Browse, Activity, Rankings, Blog, Help, Create), and a 'Sell' button. The main content area features a 2x2 grid of art images: Vincent van Gogh's 'Self-Portrait with Bandaged Ear', a bronze mask sculpture, a blue ceramic bulldog head, and a classical painting of a woman. The listing title is 'Wikipedia art' (TestNFT - wggk17O0qt), owned by the user. A 'Price History' section shows 'All Time' with a 'No trading data yet' message and a line graph icon. Below are sections for 'Listings' and 'Offers'. A 'Details' sidebar on the left shows 'Created by you' and the name 'Wikipedia art'.

<https://testnets.opensea.io/assets/0x9bfbaEa68d831cFEC17a64098246864Eb726Ba7C/62901177364322>

Troubleshoot on Opensea

https://rinkeby-api.opensea.io/api/v1/asset/<contract>/<id>/?force_update=true

<https://rinkeby-api.opensea.io/asset/<contract>/<id>/validate/>

https://rinkeby-api.opensea.io/api/v1/asset/0x9bfbaEa68d831cFEC17a64098246864Eb726Ba7C/62901177364322/?force_update=true

<https://rinkeby-api.opensea.io/asset/0x9bfbaEa68d831cFEC17a64098246864Eb726Ba7C/62901177364322/validate/>

Subjects next time

- Oracles
- Security
- Layer2

- IPFS
- More solidity & best practices
- Ethereum nodes (&rpc)
- Ethereum deployment tools (truffle)
- Websites (javascript) to access smart contracts
- Testing
- Ethereum name system
- TheGraph

Date & Time?