



Token Launch Documentation

Create, Design, and Deploy
your own token

MARCH 8, 2022



Zilliqa changed the blockchain world forever when we launched our sharding mechanism in 2018. It created a new model for blockchain scalability and has become the foundational building block for sharding technology.



Purpose

The purpose of the **Token Launch** documentation is to:

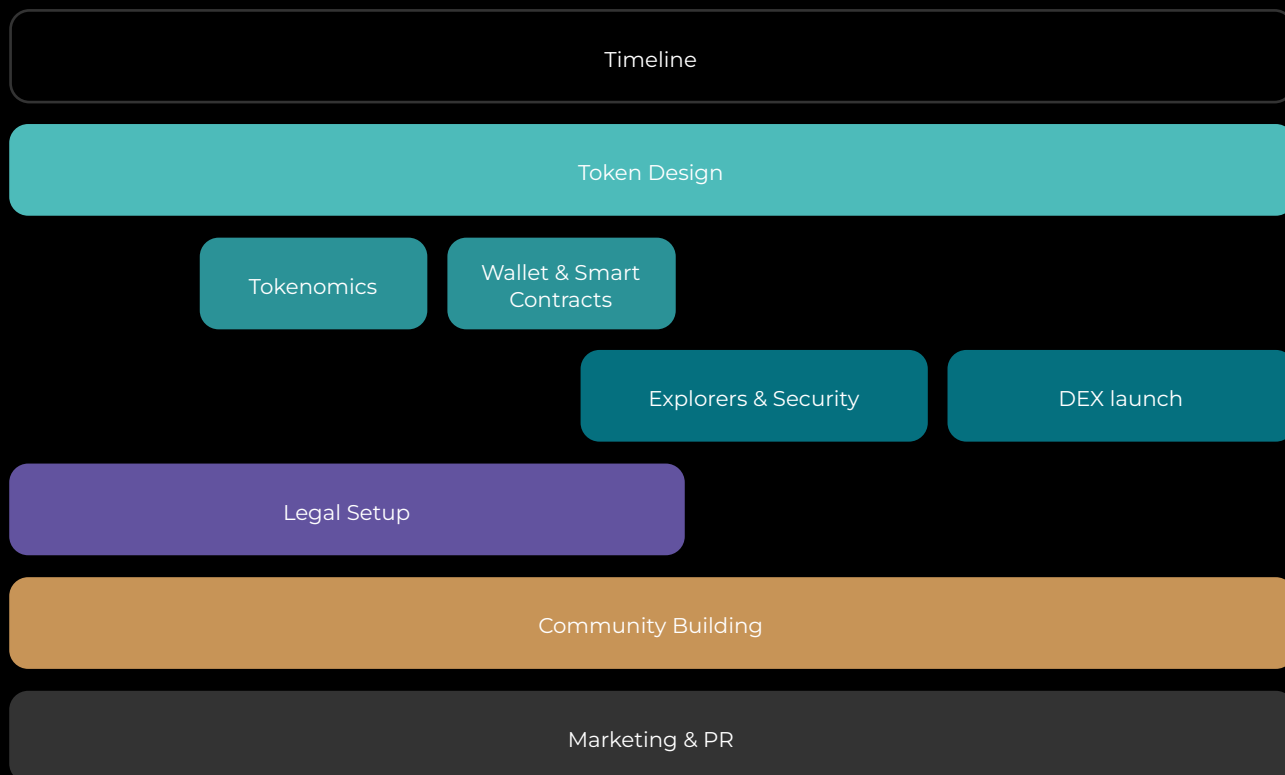
- Provide projects with the building blocks and structure to launch a token on the Zilliqa blockchain.
- Serve as a reference on how to launch tokens on the Zilliqa blockchain.
- Enable community projects and partners.



Introduction

Fundamentally, launching a token has multiple components, which are listed below:

- Token Design
- Tokenomics
- Wallets and Smart Contracts Deployment
- Explorers and Security
- DEX Launch
- Legal Setup
- Community Building
- Marketing and PR

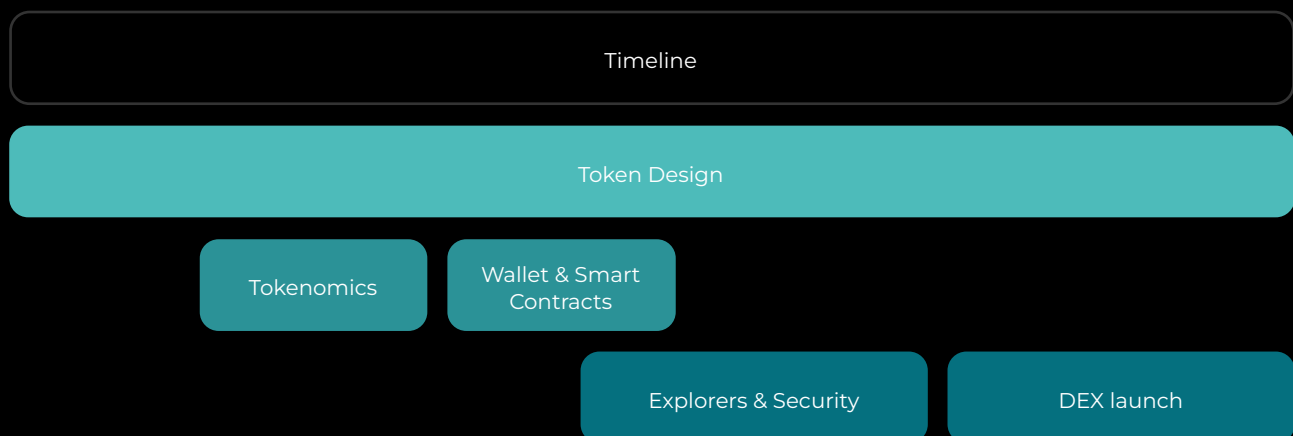




Scope

The scope of this document is **limited** to the following components of Token Launch:

- Token Design
- Tokenomics
- Wallets
- Explorers
- Smart Contract Deployment
- DEX Launch





Tokenomics

Tokenomics is an important pillar of tokens and a good tokenomic model is underpinned by tokens being **valuable**, **feasible** and **usable**.

It is highly recommended to come up with a tokenomics strategy that align with your business, value proposition, and community engagement model. Publishing a whitepaper/lightpaper is a sign that you have a well thought-out plan in place, making it easier to earn the trust of community members and investors.

Important aspects of tokenomics involve the following:

- **Token Name**

Full name of the token.

- **Token Ticker**

Symbol of the token for listings and trading.

- **Token Total Supply**

Total number of tokens minted.

- **Token Allocation**

Token allocation will identify the amount of tokens allocated to the founding team, project developers, initial coin offerings (ICO), private sale(s), or partners for their contributions to the protocol or dApps.

- **Phases of Token Distribution**

Tokens can be transferred upfront on launch or with a vesting schedule.



- **Vesting Schedule**

Vesting schedule incentivises commitment to the project for a greater reward later.

- **Rewards**

Token allocation for rewards including governance, staking, liquidity, airdrops, NFT allocations, metaverse access, exclusive access, yield farming. Includes Reward distribution time (daily, weekly, monthly, yearly) from the reward pool.

- **Staking**

Strategy for staking in conjunction with rewards and liquidity.

- **Minting, Burning, and Buyback Strategies**

Token minting and burning strategies post- launch based on tokenomics and use case. The amount of tokens to be minted will be decided based on the marketing strategy. Having too many tokens floated in the market will dilute the liquidity and price. Burning of tokens will help maintain liquidity of tokens in the market and allocated buyback will support scarcity and underpin token price appreciation.

- **Airdrop**

Strategy for airdropping tokens to the community.

Seeking legal advice on the token's utility or investment class is highly recommended to understand whether the token may be classified as a security or utility token. Zilliqa does not provide legal advice on token launches.

Please reach out to Zilliqa if you would like to have a deeper understanding on tokenomics. We have a dedicated team of experts who can help you validate your token model and economics.



Wallet

To keep tokens secure for users as well as finding a balance between the level of accessibility and depth of security for organizations, it is recommended to use a combination of different types of wallets to benefit from the accessibility of hot wallets and security of cold wallets. As such, Zilliqa suggests the following:

1. [Cold wallet] Use the Ledger hardware wallet - for individual token holders. Both for normal (single user) wallets and for multi-sig wallets.
2. For organization level - use a Zilliqa multi-sig wallet for holding the treasury tokens / ZILs / ZRC-2 tokens - the number of multi-sig holders should be a minimum of XX. [Multi-sig reference link can be found here](#). The individual wallets used in multi-sig should be cold (hardware) wallets.
3. [Hot wallet] Have an individual rewards (staking rewards or LP rewards) distribution wallet that developers can access programmatically and periodically fund it from the multi-sig wallet.

The strategy for the wallet selection is the sole responsibility of the business/project owner.

[ZilPay](#), [Moonlet](#), [Ledger](#), [Trust Wallet](#), [Zillet](#) are a few wallets that support Zilliqa's native token and ZRC-2 tokens.

For a full list of browsers please visit [here](#).



Explorers

Explorers are dedicated web portals used for the following purposes:

- Displaying block history;
- Querying transactions;
- Check balances of wallet addresses;
- View transaction details;
- View contract details; and
- Implementation of XXXXX

The following block explorers can be used to explore Zilliqa blockchain:

- [ViewBlock](#)
- [Devex](#)

To display your token with your custom logo, you should submit a pull request to [this GitHub repository](#). It is recommended to use a .svg file for your token logo.

A **sample** pull request for token listing on ViewBlock can be [found here](#).

A **sample** to add assets folder with token address in lowercase can be [found here](#).



Smart Contract Deployment

Deployment of the token contract comes with the following prerequisites:

1. For first-time users of the Zilliqa blockchain, setting up an account (wallet address) is required. The private key will be furnished when the account is created. The private key is the most important piece of information essential to access your account/wallet, so please ensure that you keep it backed up in multiple secure locations. If the private key is lost, then access to your account may be lost as well.
2. Install the ZilPay browser extension in your Chrome or Firefox browser. The ZilPay extension can be [found here](#).

ZRC-2 is the formal standard for fungible tokens on the Zilliqa blockchain. It is an open standard for creating utility tokens on the Zilliqa blockchain.

1. The ZRC-2 standard allows for the following functionalities:
2. Minting/burning of tokens;
3. Transferring tokens from one account to another;
4. Querying account's token balance;
5. Querying total token balance; and
6. Approving third-party to send a certain amount of tokens.

The reference ZRC-2 contract code can be [found here](#).

Zilliqa's [IDE](#) or [this dedicated tool](#) can be used to deploy your token contract.



The ZRC-2 contract requires you to provide initial parameters such as:

1. Contract owner
2. Token name
3. Token symbol
4. Decimal count
5. Initial supply

Set your token variables

contract owner
zllv25at4s3eh9w34uqqhe3vdfsvcwq6un3tupc2

token name
MyDummyToken

token ticker
MDT

decimal count
10

initial supply
1000000000000

Please note that once the contract is deployed, the initial parameters become immutable, and the contract cannot be modified afterwards.

For assistance, please get in touch with the Zilliqa team.



Launching on a Decentralized Exchange (DEX)

After the token is deployed on mainnet, it should be listed for trading on exchange(s) in order to ensure liquidity. The most widely used and liquid DEX in the Zilliqa ecosystem is ZilSwap. We recommend to coordinate the deployment of token contract and Liquidity pool creation in [Zilswap](#) in advance. Once the contract is deployed, the owner must immediately create a Liquidity Pool in the DEX so that no one else can do and list their tokens for sale.

ZilSwap supports ZRC-2 tokens with ZIL (native token of Zilliqa) as the trading pair.

Tokenomics will provide clarity on the price of token launch to the public. However, the foremost important step is to list the token. ZilSwap supports configuring and whitelisting of token address(es) on ZilSwap.

After the token is whitelisted, a liquidity pool for the token/ZIL pair should be immediately created as soon as the token is launched. It is important to:

1. Launch token/ZIL liquidity pool upon whitelisting to maintain control of the pool and prevent against third-parties forming and controlling the liquidity pool.
2. Ensure that there is enough liquidity in the pool to avoid price fluctuation. This is to enable support for maintaining the token's to ZIL pair with desired price.



Post liquidity pool creation, close coordination is required to be maintained with the ZilSwap team to enable the trade for the ZRC-2 token and ZIL trading pair.

Adding your project to the Zilstream website

To list your token on ZilStream (Zilstream.com), ensure that the following criteria are met

- I have added my token to the [ViewBlock Meta Repo](#).
- I have added smart contracts and team wallet addresses (or any other address containing locked funds) to supply_skip_addresses.

Go to <https://github.com/ZilStream/tokens> and create a PR, it should include your token in `tokens.json`. The Zilstream team will review the request and add it to their platform.

ZilBridge (Process to bridge tokens)

In the event that you wish to bridge the ZRC-2 tokens to ERC-20, the tokens will have to meet the following criteria:

- No rebasing token;
- Must be **strictly either** ZRC-2 or ERC-20 compliant (i.e. the token should not be on both Zilliqa and Ethereum chains);
- At least 500 token holders;
- At least USD\$1m market capitalisation for the last 2 weeks; and
- Listed on both [CoinGecko](#) and [CoinMarketCap](#) for ERC-20 tokens, or listed on [ZilStream](#) for ZRC-2 tokens.

Once the criteria has been checked, clone <https://github.com/Zilliqa/switchero-tradehub-eth> and follow the instructions provided in the README.md file to deploy a modified version of



an ERC-20 token contract on the Ethereum mainnet.

Once the ERC-20 contract is deployed, contact us at zilbridge@zilliqa.com from your official email account and provide the following information -

- Project team name
- Project team contact
- ERC-20 bridged token contract address
- ZRC-2 token contract address
- Desired launch date
- Audit report (if any)
- Any other information for the team to take note

SLA for a response will be between 1-2 weeks.



Adding your project to the Zilliqa website

To list your project on Zilliqa ecosystem page on Zilliqa.com, go on to <https://github.com/Zilliqa/zil-ecosystem> and create a PR by using the following format.

your-project-name

- **project.md**

- ensure the filename is named exactly project.md. All lowercase with the extension .md
- create frontmatter tags by encasing tags in three dashes above and below in the following

(required) **name:** The name of the project

(required) **date:** The date of project addition

(required) **description:** The one line summary of the project and its integration to Zilliqa

(required) **logo:** all image files must be 100x100px

(required) **categories:** A comma separated list of categories describing the project. List of categories can be found on <https://github.com/Zilliqa/zil-ecosystem#categories>

(required) **status:** The status of the project: [live | building]

(optional) **twitter:** URL to Twitter page

(optional) **website:** URL to the website

(optional) **telegram:** URL to Telegram channel

(optional) **discord:** URL to Discord invite

- **your-logo.png**

- Ensure logo is in one of the following formats (png|jpg|svg)
- Logo has to be visible on a light gray background (#F5F5F5)
- Logo should have a 1:1 ratio. e.g. 300px x 300px.
- Don't include your project name in the logo. Include only the icon.



ZILLIQA

Connect with us on
the following social
media channels

We are happy to assist you on your successful
token launch - hope to speak to you soon.

