

BEP-20 Token Customization

To get started you will need to decide the following parameters for your token.

Open the "Token.sol" file and find line 353.

[illegible]

Parameters:

- `_name` - Token name, for example Prime Token or Prime Coin
- `_symbol` - Short name of your token. In this case PRM (Short of Prime)
- `_decimals` - By default most of the tokens have 18 decimals. You can change this to whatever you want.
- `_totalSupply` - The number of tokens you want to create. Keep in mind that you will need to add `_decimals * 0` after the amount. So if you want to have 1,000,000 tokens, you will need to add 18 zeros. It would look like `1000000000000000000000000`

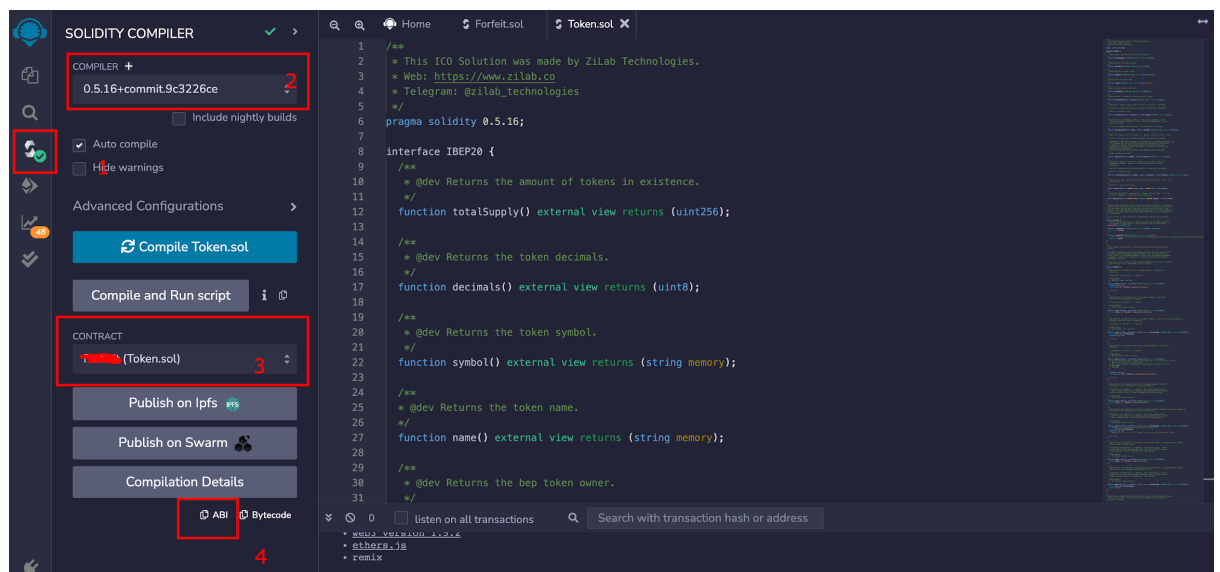
Token Deployment on Mainnet

Once you have made the changes on the previous step "Token Customization", now you need to follow the next steps to deploy your own token on BSC Mainnet.

Keep in mind that you will need Metamask with some amount of the BNBs to pay for the gas fees. Usually, it costs a maximum of 10-15 EUR.

1. Open the <https://remix.ethereum.org/>
2. Create a new file with the name "Token.sol".

3. Copy the Token.sol file content into newly created file.



- 4.

Select version - 0.5.16



- 5.

Your metamask wallet should have at least 0.1 BNB!

If you have followed the steps now you should have your own token deployed on the BSC Mainnet. Congratulations!

Smart Contract Verification

To be able to interact with the smart contract, you must verify it.

To do so, open your contract on BSCScan.com

The address looks similar to this:


<https://bscscan.com/address/{Contract Address}#contracts>

1. Click on the text "Verify and Publish"

2.

Verify & Publish Contract Source Code

COMPILER TYPE AND VERSION SELECTION



Source code verification provides **transparency** for users interacting with smart contracts. By uploading the source code, BscScan will match the compiled code with that on the blockchain. Just like contracts, a "smart contract" should provide end users with more information on what they are "digitally signing" for and give users an opportunity to audit the code to independently verify that it actually does what it is supposed to do.

Please enter the Contract Address you would like to verify

YOUR ADDRESS

Invalid Length

Please select Compiler Type

Solidity (Single file)

Please select Compiler Version

v0.5.16+commit.9c3226ce

☒ Un-Check to show all nightly Commits also

Please select Open Source License Type ⓘ

3) MIT License (MIT)

☒ I agree to the [terms of service](#)

Continue

Reset

3.

Enter the Solidity Contract Code below *Fetch from Gist

```
/*
 * @dev Destroys `amount` tokens from `account`; `amount` is then deducted
 * from the caller's allowance.
 *
 * See {_burn} and {_approve}.
 */
function _burnFrom(address account, uint256 amount) internal {
    _burn(account, amount);
    _approve(account, _msgSender(), _allowances[account][_msgSender()].sub(amount, "BEP20: burn amount exceeds allowance"));
}
}
```

Constructor Arguments [ABI-encoded](#) (for contracts that were created with constructor parameters) ↓

For additional information on Constructor Arguments [see Our KB Entry](#)

Contract Library Address (for contracts that use libraries, supports up to 10 libraries) →

Misc Settings (Runs, EvmVersion & License Type settings) →

On the textarea "Enter the Solidity Contract Code below" you must paste the contract code from the remix website.

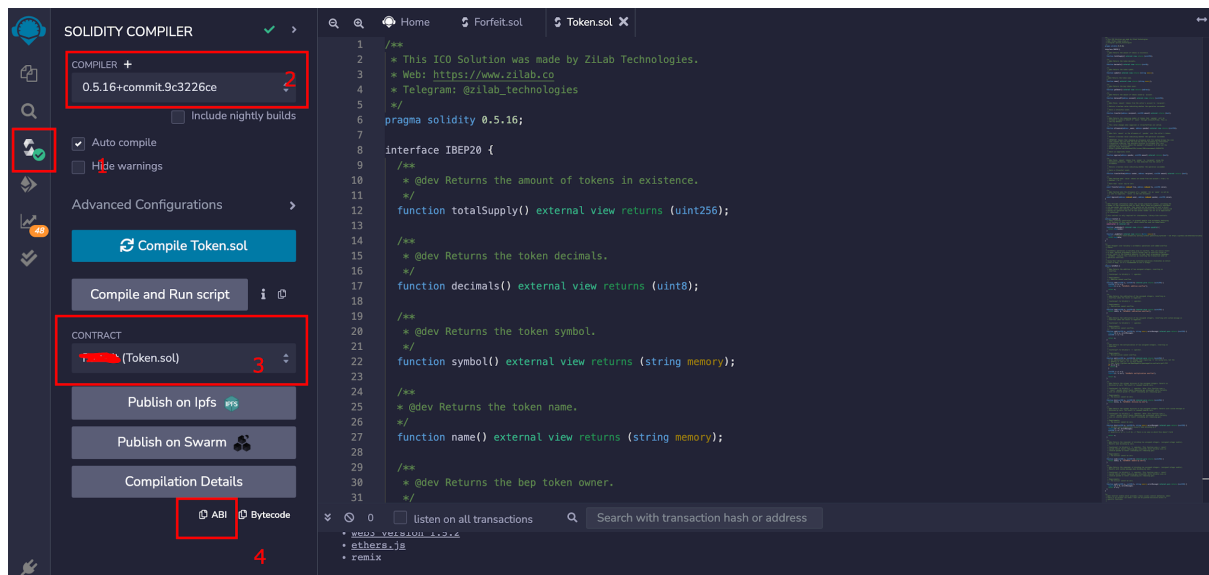
4. Click "Verify"

Presale Contract Deployment on Mainnet

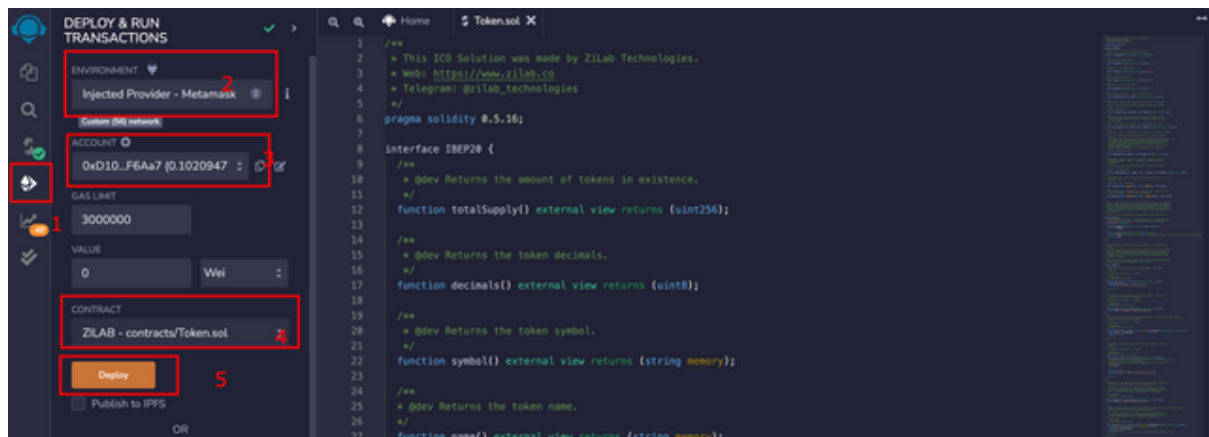
Once you have made the changes on the previous step "Token Customization", now you need to follow the next steps to deploy your own token on BSC Mainnet.

Keep in mind that you will need Metamask with some amount of the BNBs to pay for the gas fees. Usually, it costs a maximum of 10-15 EUR.

1. Open the <https://remix.ethereum.org/>
2. Create a new file with the name "ForzaPresale.sol".
3. Copy the PreSale.sol file content into newly created file.
- 4.



5.



If you have followed the steps now you should have your own presale contract deployed on the BSC Mainnet. Congratulations!

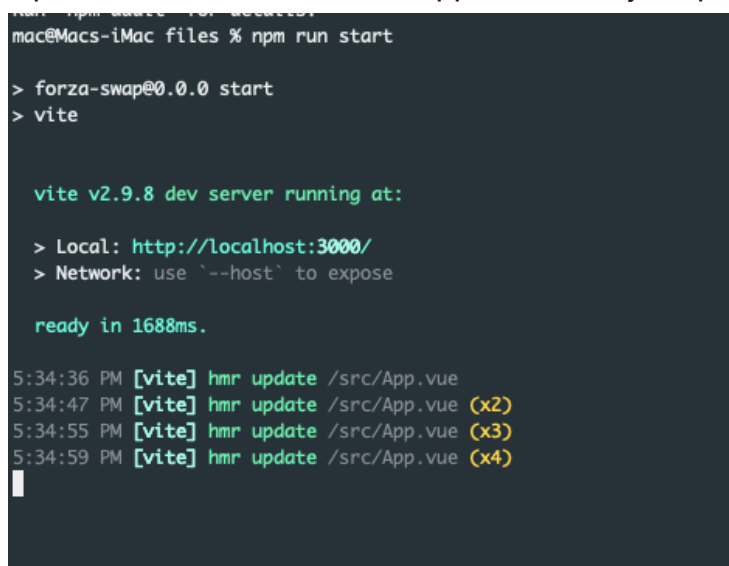
Website Setup

1. You need to install node dependencies with the command:

“npm install” on the project root

2. Run the command:

“npm run start” to launch the application on your pc.



Now you can open <http://localhost:3000>

All the website configuration are stored here:

src/environment.ts

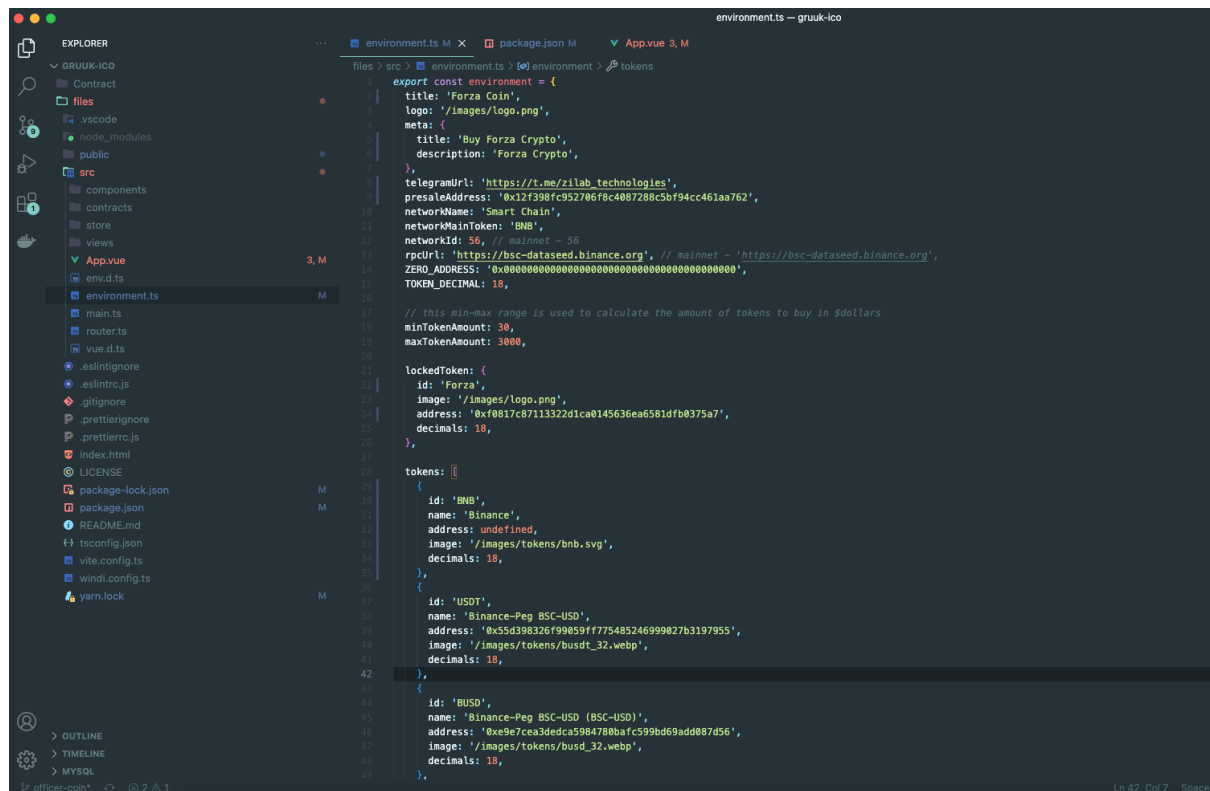
Min-max purchase per investor can be edited here:

```
// this min-max range is used to calculate the amount of tokens to buy in $dollars  
  
minTokenAmount: 30,  
  
maxTokenAmount: 3000,
```

Presale address:

```
presaleAddress: '0x12f398fc952706f8c4087288c5bf94cc461aa762',
```

Update this address with your just deployed presale address.



Website Deployment on cPanel

To deploy the application the cpanel you will need to run the following command:

Note: Run this command only when your local version is fully done and is ready for the production.

“npm run build”

It will compile all the assets and you will have a directory “dist” on the project root.

Create a ZIP of dist folder and upload this directory on your hosting.

And that's it!

If you have any questions regarding the ICO and setup, please let us know on Telegram

@zilab_technologies