

# Add New Token to Optics

How to add a new token to Optics v2.

## Getting Started

Optics is a cross-chain communication system designed for use on any EVM chain. The Token Bridge xApp (an application built on top of Optics) facilitates the transfer of tokens between chains. Adding a new token to the Optics Bridge UI can be done at any time using the [optics-bridge-gui-community](#) GitHub repository.

info View [PR#20](#) to view an example of the updates required to add the PACT token.

## Steps to Add a New Token

- Step 1:
  - Add Token Identifier
- Step 2:
  - Add Token Symbol
- Step 3:
  - Add Representation Address

## Add Token Identifier

Add the Token Identifier insrc/config/tokens.environment.ts .

- Testnet:
  - [tokens.dev.ts](#)
- Mainnet:
  - [tokens.mainnet.ts](#)

```
export type TokenIdentifier

=

{ domain : string | number id : string }

export type mainTokenName = 'TOKEN'

// IMPORTANT make name same as tokens[token].symbol value const

TOKEN :

TokenIdentifier

=

{ domain :

'celo' , id :

" }

export

default

{ TOKEN , }
```

## Add Token Details

Add the token details to the token list insrc/config/config.environment.ts

- Testnet:
  - [config.dev.ts](#)
- Mainnet:
  - [config.main.ts](#)
- Community Testnet:
  - [config.stagingCommunity.ts](#)

- Community Mainnet:
- [config.mainnetCommunity.ts](#)

```
import
TOKEN
from
'../assets/token-logos/token.png'
import
mainnetTokens
from
'./tokens.main' import
{
TokenMetadata ,
NetworkMetadata
}
from
'./config.types' import
representationsMain
from
'./representations.main'
// Add TOKEN details here export
const
tokens :
{
[ key : string ] :
TokenMetadata
}
=
{ TOKEN :
{ nativeNetwork :
" , symbol :
" , name :
" , icon :
, decimals :
, coinGeckoId :
" , tokenIdIdentifier :
, nativeOnly :
, minAmt : } , }
```

# Add Representation Addresses

Add the representation addresses insrc/config/representations.environment.ts .

- Testnet:
- [representations.dev.ts](#)
- Mainnet:
- [representations.main.ts](#)

```
const ethereum =
```

```
{ TOKEN :
```

```
"" , } ;
```

```
const celo =
```

```
{ TOKEN :
```

```
"" , } ;
```

```
const polygon =
```

```
{ TOKEN :
```

```
"" , } ;
```

```
export
```

```
default
```

```
{ celo , ethereum , polygon , } Edit this page Previous Optics Bridge GUI Next Migrating to Optics v2
```