ScaleBit

TON&Tact介绍

Jason

个人介绍

- Jason
- ScaleBit安全研究员
- 智能合约审计、漏洞挖掘、产品研发
- 审计项目数量50+

大纲

- TON生态介绍
- Tact语言介绍
- 课程学习目标
  - 完成一个Web版割草游戏
- Tact开发环境搭建
- Helloworld演示
  - 账户介绍
  - 转账
  - 合约编译部署调用
- 总结
- 作业

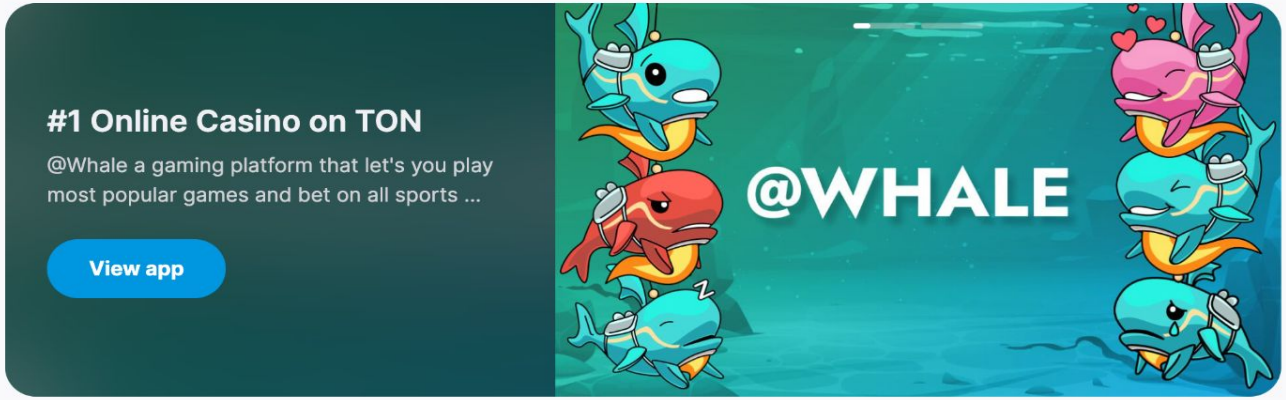
TON生态介绍

- Telegram于 2018 年创建了 Telegram Open Network
- ICO 获得 17 亿美元的代币销售
- SEC 命令 Telegram 停止销售之前与 TON 区块链相关的代币 $GRAM
- 2020 年 5 月，Telegram宣布终止参与区块链的开发，并开始向早期投资者发放退款
- 在 Telegram 放弃 TON 项目后，一群社区开发人员将其重新命名为 The Open Network
- Telegram App整合自托管钱包“Ton Space”，支持TON钱包
- 2023年 9 月，Telegram 已认定选择 TON 网络作为其 Web3 基础设施的区块链网络，TON 生态的多个项目和进展都将基于 Telegram 进行构建
- TON 背靠 Telegram 的 8 亿多用户支持，有着类似微信的发展前景

Explore 551 apps in TON Ecosystem

Search applications /

Exchanges CEX | Staking | Wallets | Explorers | Bridges | Utilities | Channels | NFT Collectio



News

See all

- TON processed over 80 million inscriptions transactions in 15 days TON Community is delighted to se
- Introducing fair launches in the TON Ecosystem with Ton Raffles Ton Raffles has pioneered a new er
- Layerswap integrates TON Layerswap has integrated TON, paving the way for users to access...
- HackenProof launches new bug

Promoted Apps

Add project

Jackpot.ton | Playmuse Marketplace | Bitsler.com | Crypto Ads Platform | JetTon Games

Toncoin
$2.22 -0.92% • 24h

Market Cap
$7.66B -11.87% • 24h

Tact语言介绍

TON 生态系统主要使用两种智能合 约编程语言：FunC 和 Tact。

- FunC 是一种低级语言，专为深入了解 TON 架构的开发人员而设计，它在开发复杂的多合约系统时可能具有挑战性。
- Tact 是 TON 区块链的一种新编程语言，注重效率和简单性。它的设计易于学习和使用，并且非常适合智能合 约。Tact 是一种静态类型语言，具有简单的语法和强大的类型系统。
- Tact 提供了受 JavaScript 和 Typescript、Rust 和 Swift 启发的熟悉语法。对于新开发人员来说，代数数据类型和编译时执行等强大功能看起来有机且友好。
- 编译器会把Tact代码转译为FunC代码，最终编译为类似Fift的字节码，在TON VM虚拟机上执行。

TACT UNLEASHING TON'S POWER WITH SAFE AND SCALABLE SMART CONTRACTS

Tact code

Write Tact code below

```
message (0x123123) TransferMsg {
  to: Address;
  text: String;
}

contract SimpleContract {
  init() {}
  receive() {}
  receive(msg: TransferMsg) {
      send(SendParameters{
          to: msg.to,
          value: 0,
          mode: SendRemainingValue,
          body: msg.text.asComment()
      });
  }
}
```

FunC result

Tact compiler will turn the program to FunC code on the fly

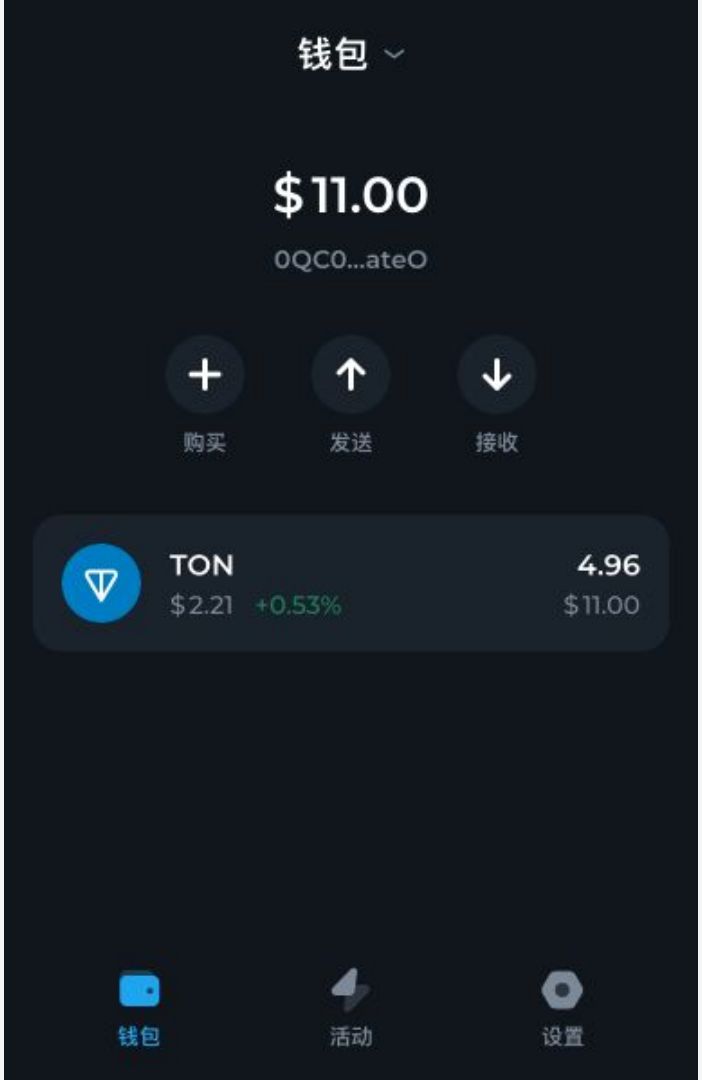
```
include "imports/stdlib.fc";

int op::transfer_coins() asm "0x123123 PUSHINT";
() recv_internal(int my_balance, int msg_value, cell i
  if (in_msg_body.slice_empty?()) { ;; ignore empty me
    return ();
  }

  slice cs = in_msg_full.begin_parse();
  int flags = cs~load_uint(4);
  if (flags & 1) { ;; ignore all bounced messages
    return ();
  }

  int op = in_msg_body~load_uint(32);
  if(op == op::transfer_coins()) {
    slice to = in_msg_body~load_msg_addr();
    cell text = in_msg_body~load_ref();
```

课程学习目标

课程总共6节课，会学习Tact语法、Jetton和NFT标准，完成一个Web版割草游戏

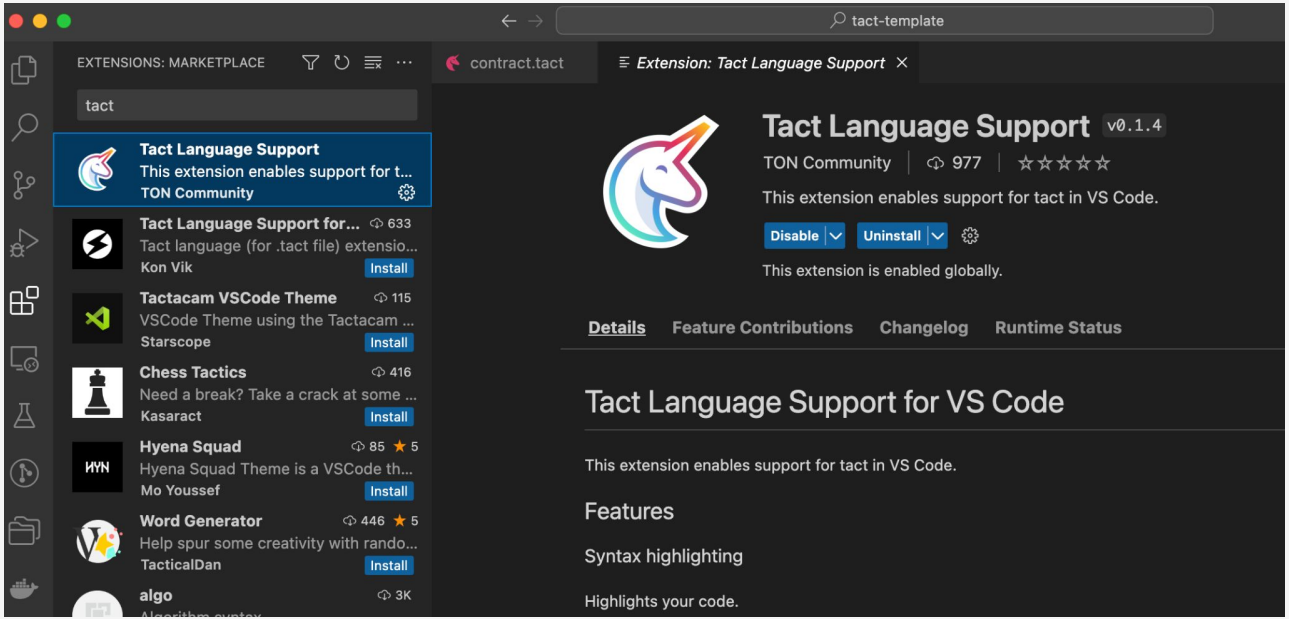
Tact开发环境搭建

- 安装TON钱包
- 创建区块链账户
- 配置测试区块链网络
- 申领测试币



Tact开发环境搭建

- 安装Visual Studio Code插件Tact Language Support



Helloworld 合约代码

合约说明

- 从模版工程开始
  git git@github.com:tact-lang/tact-template.git
- 一个示例合约SampleTactContract
- 具备状态：owner和counter
- 初始化函数：init
- 合约方法：add
- 消息处理函数：可处理两种类型消息
- getter方法：counter

contract.tact

sources > contract.tact

Howard Peng, 4 months ago | 2 authors (Steve Korshakov and others)

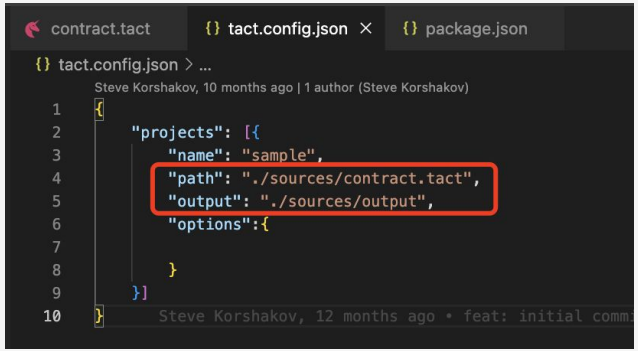
```tact
import "@stdlib/deploy";

message Add {
    amount: Int as uint32;
}

contract SampleTactContract with Deployable {

    owner: Address;
    counter: Int as uint32;

    init(owner: Address) {
        self.owner = owner;
        self.counter = 0;
    }

    fun add(v: Int) {

        // Check sender
        let ctx: Context = context();
        require(ctx.sender == self.owner, "Invalid sender");

        // Update counter
        self.counter = (self.counter + v);
    }

    receive(msg: Add) {
        self.add(msg.amount);
    }

    receive("increment") {
        self.add(1);
        self.reply("incremented".asComment());
    }

    get fun counter(): Int {
        return self.counter;
    }
}
```

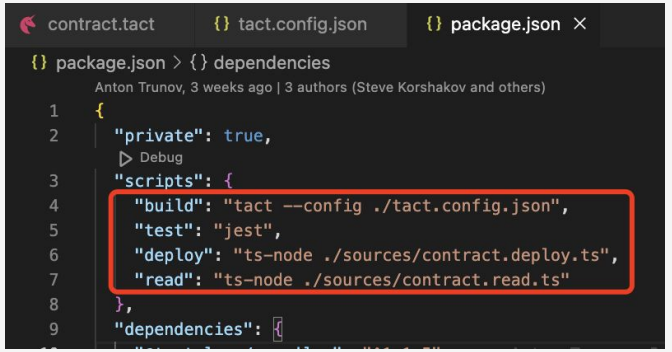
Steve Korshakov, 12 months ago • feat: initial commit and

Helloworld 编译

配置说明

- 在tact.config.json文件中配置源文件和 输出目录
- 在package.json文件中配置 编译命令
- 执行命令 yarn build 编译





Helloworld 部署

准备部署代码说明

- 导入相关的TS库
- 准备合约参数
- 准备部署合约
- 计算合约地址
- 合约部署网址

contract.tact | tact.config.json | package.json | contract.deploy.ts M | deploye

sources > contract.deploy.ts > <function>

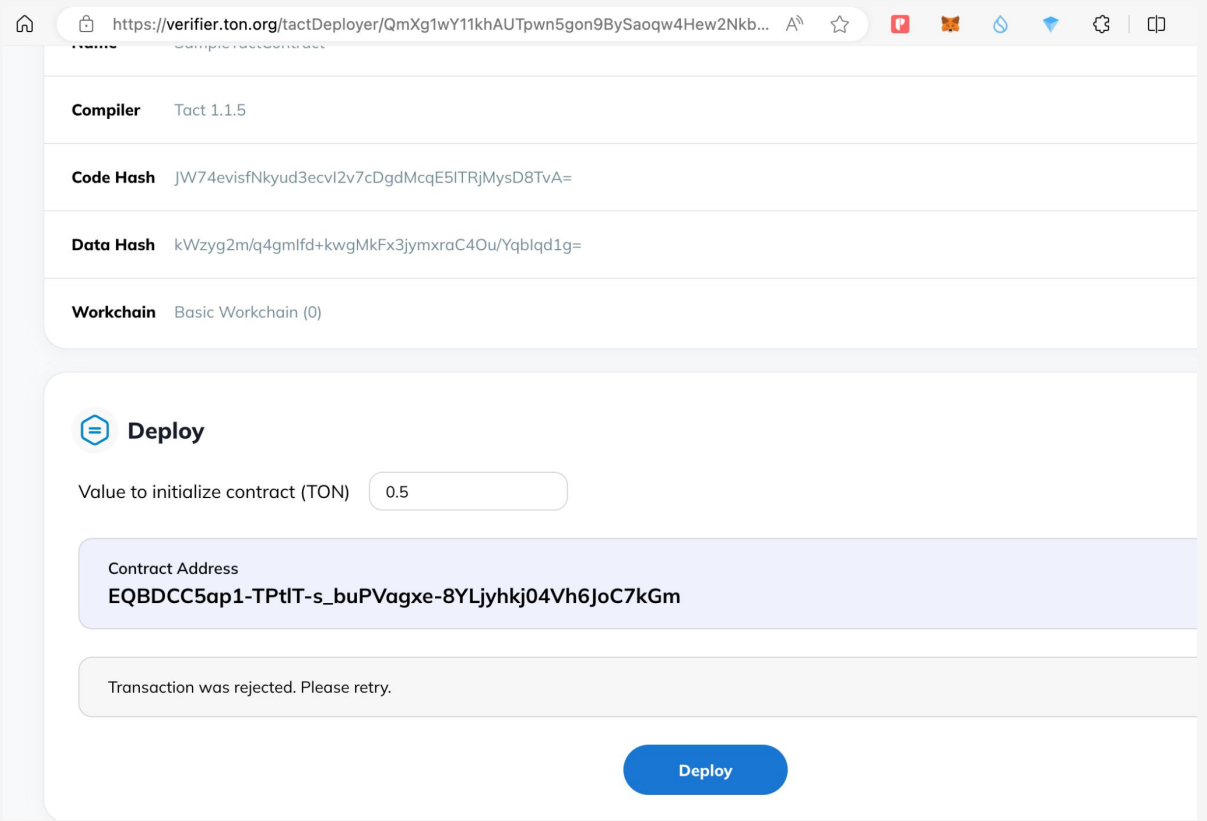
```ts
import * as fs from "fs";
import * as path from "path";
import { Address, contractAddress } from "@ton/core";
import { SampleTactContract } from "./output/sample_SampleTactContract";
import { prepareTactDeployment } from "@tact-lang/deployer";

(async () => {
    // Parameters
    let testnet = true;
    let packageName = "sample_SampleTactContract.pkg";
    let owner = Address.parse("0QC0BWOHCGvMpGQPUhg43b-EWAIl-FuNmSyNpm6HnV_cateO");
    let init = await SampleTactContract.init(owner);

    // Load required data
    let address = contractAddress(0, init);
    let data = init.data.toBoc();
    let pkg = fs.readFileSync(path.resolve(__dirname, "output", packageName));

    // Prepareing
    console.log("Uploading package...");
    let prepare = await prepareTactDeployment({ pkg, data, testnet });

    // Deploying
```

PROBLEMS 1 | OUTPUT | DEBUG CONSOLE | TERMINAL | GITLENS

```
Contract Address
================================================================================

kQBDCC5ap1-TPtlT-s_buPVagxe-8YLjyhkj04Vh6JoC7vos

================================================================================
Please, follow deployment link
================================================================================

https://verifier.ton.org/tactDeployer/QmXg1wY11khAUTpwn5gon9BySaoqw4Hew2NkbQ251CeqWk?testnet

================================================================================
✨  Done in 10.05s.
```

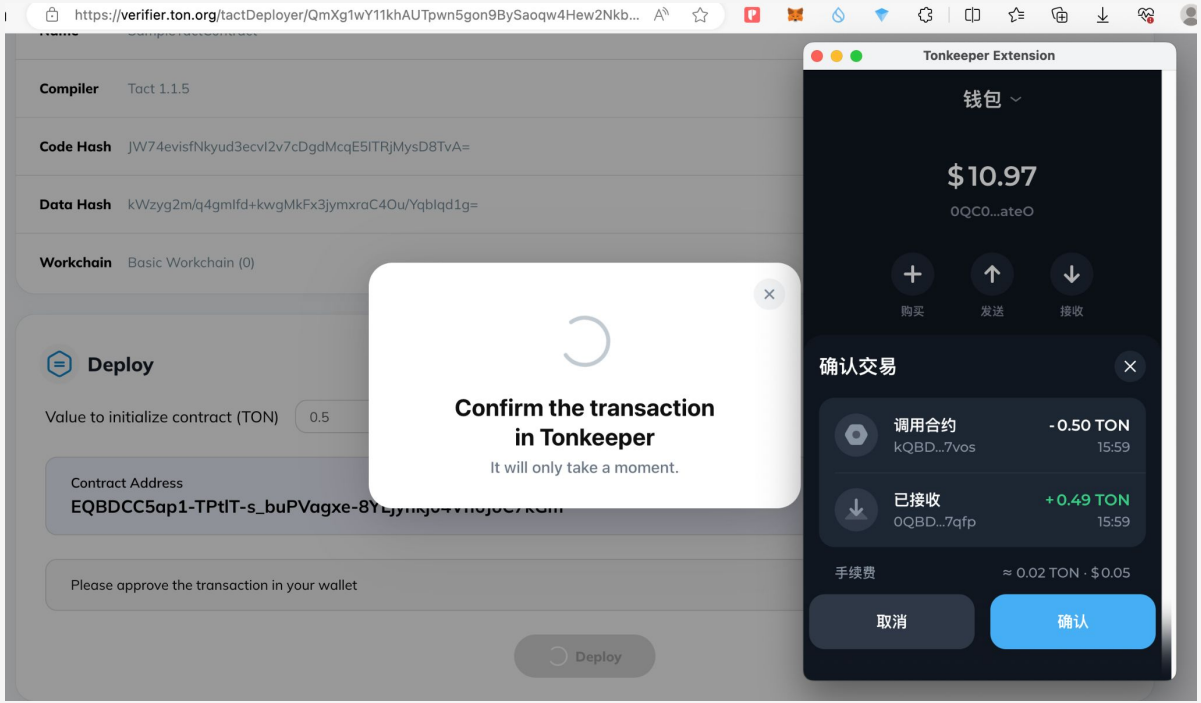
Helloworld 部署

浏览器打开网址



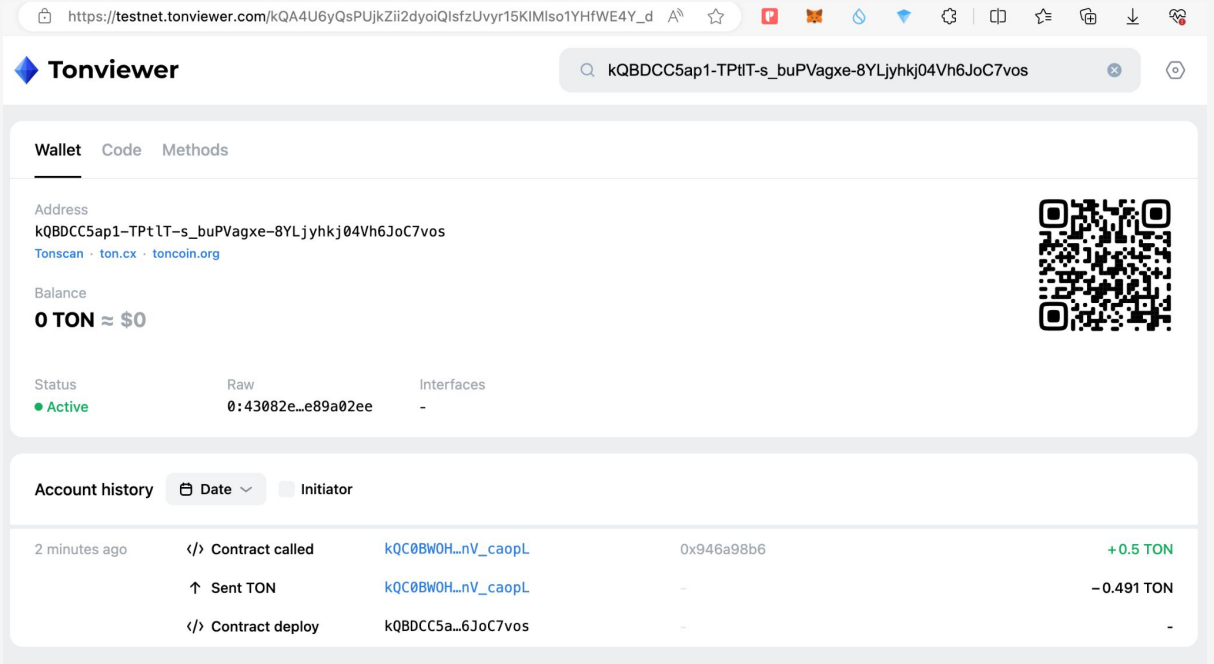
Helloworld 部署

钱包签名部署



Helloworld 部署

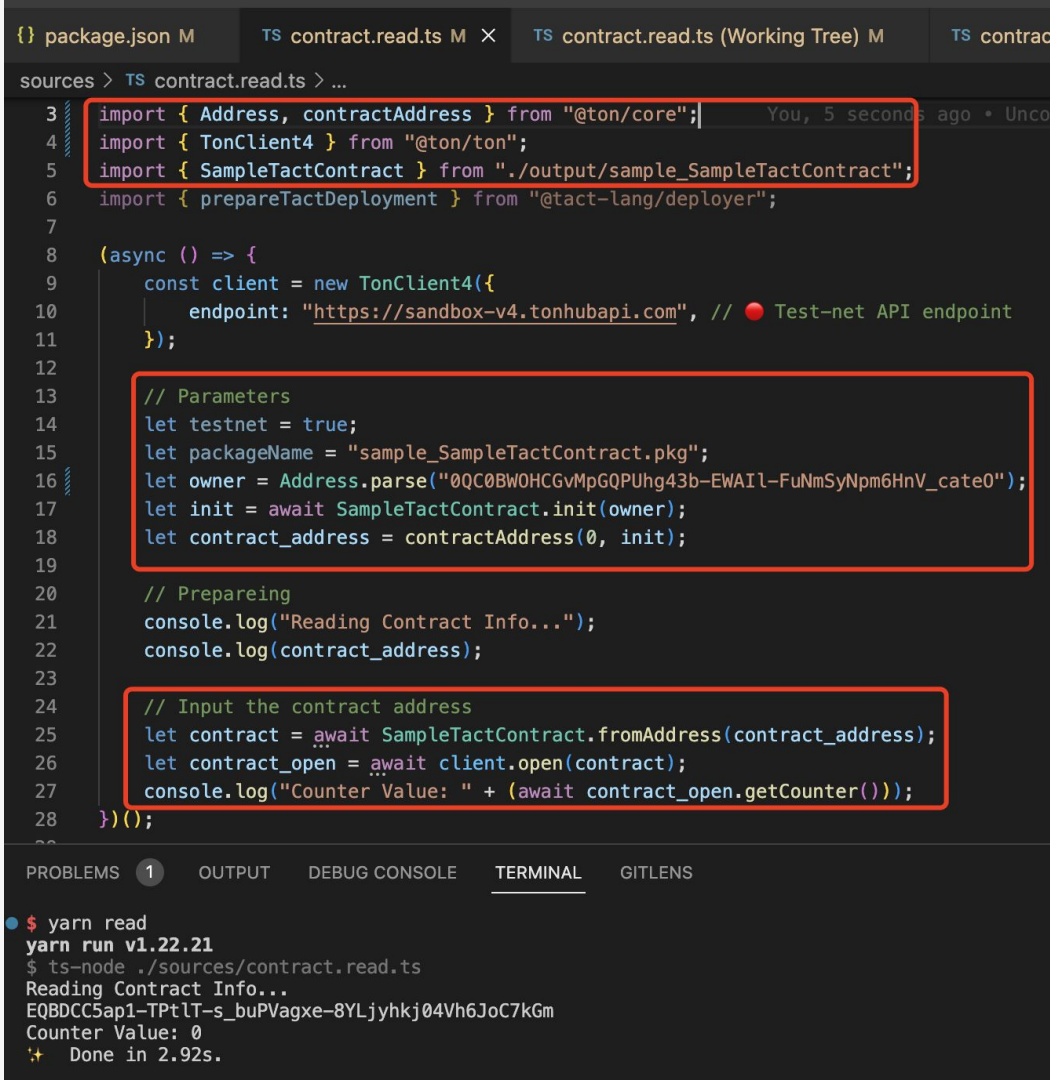
浏览器查看已部署合约



Helloworld 调用getter

合约调用说明

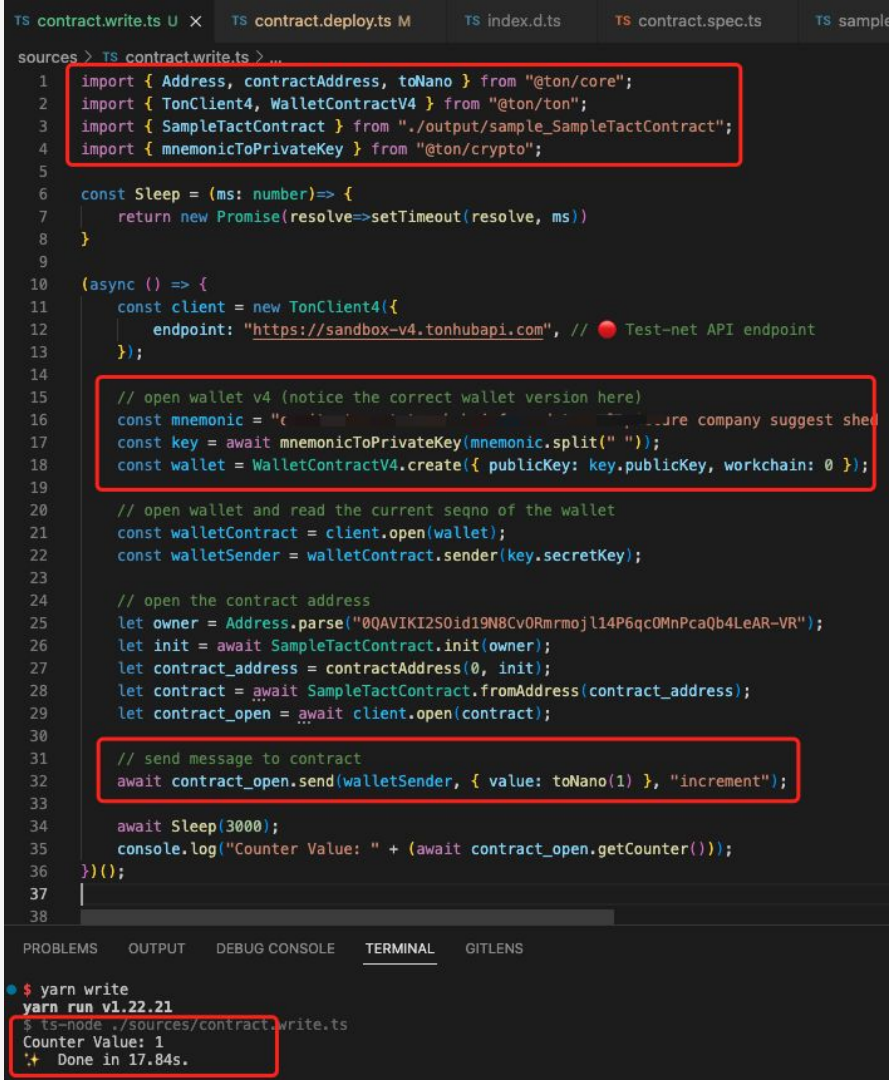
- 导入TON的开发库
- 准备合约参数
- 调用getCounter读取状态值



Helloworld 发送消息

合约调用说明

- 导入TON的开发库
- 准备合约参数
- 调用send方法发送消息



总结

- TON生态背靠Telegram
- TON的合约语言有FunC和Tact
- 在TON上操作，需要钱包和TON代币，开发时建议使用TON测试网络
- Tact开发环境搭建
- Tact Helloworld合约编写，编译、部署、和调用

参考资料

- https://docs.ton.org/
- https://tact-lang.org/
- https://github.com/tact-lang/awesome-tact
- https://docs.tact-lang.org/
- https://ton.app
- https://foresightnews.pro/article/detail/21329
- https://www.fx168news.com/article/392979
- https://www.panewslab.com/zh/articledetails/ty2spsyy.html
- https://www.techflowpost.com/article/detail_14664.html

作业

修改SampleTactContract合约，增加一个getter方法，重新部署，并 调用此新方法。
修改SampleTactContract合约，增加一个消息 处理方法，重新部署，并 发送此新消息。

Thanks

Contact us:

- Twitter: @scalebit_
- Email: contact@scalebit.xyz

ScaleBit

More information : www.scalebit.xyz