ScaleBit

OpenBuild

Tact 合约测试和交互

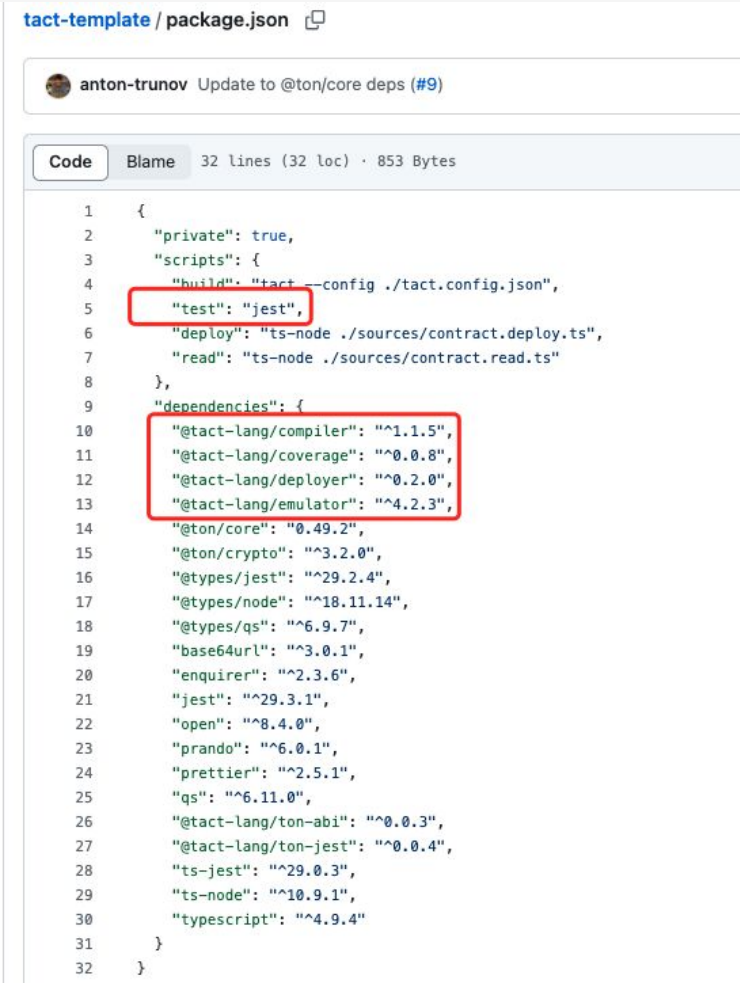
Jason

大纲

- 合约测试
  - 测试框架介绍
  - 编写测试代码
- 合约交互
  - Typescript SDK和FT交互调用
  - Typescript SDK和和NFT交互调用
- 总结
- 作业

测试框架介绍

- TON有一个sandbox的区块链模拟执行环境
- Tact的TypeScript SDK已经默认集成了sandbox，提供该环境
  - https://github.com/tact-lang/tact-template



智能合约测试的一般流程

智能合约测试可以覆盖安全性、优化 Gas 支出以及检查边缘情况。

- 创建测试账户
- 创建合约实例
- 发送特定消息
- 检查消息执行结果
- 检查状态

```typescript
import { toNano } from "@ton/core";
import { ContractSystem } from "@tact-lang/emulator";
import { SampleTactContract } from "./output/sample_SampleTactContract";

describe("contract", () => {
    it("should deploy correctly", async () => {
        // Create ContractSystem and deploy contract
        let system = await ContractSystem.create();
        let owner = system.treasure("owner");
        let nonOwner = system.treasure("non-owner");
        let contract = system.open(await SampleTactContract.fromInit(owner.address));
        system.name(contract.address, "main");
        let track = system.track(contract);
        await contract.send(owner, { value: toNano(1) }, { $$type: "Deploy", queryId: 0n });
        await system.run();
        expect(track.collect()).toMatchInlineSnapshot(`…
        `);

        // Check counter
        expect(await contract.getCounter()).toEqual(0n);

        // Increment counter
        await contract.send(owner, { value: toNano(1) }, "increment");
        await system.run();
        expect(track.collect()).toMatchInlineSnapshot(`…
        `);
```

执行测试

- Ton Emulator 允许您在 Nodejs 代码中拥有一个小型虚拟区块链，该库是专门为在单元测试中测试智能合约而构建的。
- 支持jest的静态测试。
  - yarn jest　　生成快照或测试
  - yarn jest -u　　更新快照

jest生成快照

```
describe("contract", () => {
    it("should deploy correctly", async () => {

        // Init test
        const system = await ContractSystem.create();
        const treasure = await system.treasure('my treasure');
        const contract = system.open(sample_Contract.fromInit(
        const tracker = system.track(contract.address);

        // Send a message
        await contract.send(treasure, { value: toNano(1) }, {
        await system.run();

        // Testing output
        expect(tracker.collect()).toMatchInlineSnapshot();
    });
});
```

```
expect(tracker.collect()).toMatchInlineSnapshot(`
    [
      {
        "$seq": 0,
        "events": [
          {
            "$type": "deploy",
          },
          {
            "$type": "received",
            "message": {
              "body": {
                "cell": "x{946A98B60000000000000000}",
                "type": "cell",
              },
              "bounce": true,
              "from": "kQAI-3FJVc_ywSuY4vq0bYrzR7S4Och4y7bTU_i5yLOB3A6P",
              "to": "kQBrSAP2y7QIUw4_1q0qciHfqdFmOYR9CC1oinn7kyWWRuoV",
              "type": "internal",
              "value": 1000000000n,
            },
          },
```

在合约中启用dump


```
31      receive("increment") {
32          dump("received increment");
33
34          self.add(1);
35          self.reply("incremented".asComment());
36      }
```

FT调用案例

合约代码如下，通过Tact Typescript SDK进行交互

https://github.com/howardpen9/jetton-implementation-in-tact

NFT调用案例

合约代码如下，通过Tact Typescript SDK进行交互

https://github.com/howardpen9/nft-template-in-tact/blob/tutorial/sources/contract.tact

总结

- 智能合约测试能极大程度保障合约代码的正确性
- Tact Typescript SDK功能强大，能实现和链的各种交互

作业

做一个TODO合约应用，用户可以录入每天的任务，用户将当天的所有任务完成后，可以获得一个完成任务的勋章NFT。

参考

本章参考代码

https://github.com/0xOutOfGas/tact-template-jetton

Thanks

Contact us:

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

ScaleBit

More information : www.scalebit.xyz