

前言

最近在学习Polkadot底层的substrate框架及moonbeam兼容以太坊的智能合约平台,为了更好的学习熟练Rust的语法编程,整理个Rust以太坊转账的代码。

准备

- 1、使用自己的节点或者使用第三方的公共节点提供RPC地址
- 2、以太坊地址

使用Cargo创建项目

```
cargo new ethers-transfercd ethers-transfer
```

请自行选用文本编辑器打开 Cargo.toml 文件。需要安装ethers.rs库等。

```
[package]name = "ethers-transfer"version = "0.1.0"edition = "2021"# See more keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html[dependencies]ethers = "1.0.2"ethers-solc = "1.0.2"tokio = { version = "1", features = ["full"] }serde_json = "1.0.89"serde = "1.0.149"
```

发送交易代码示例

```
use ethers::providers::{Provider, Http};use ethers::{utils, prelude::*};type Client = SignerMiddleware<Provider<Http>, Wallet<k256::ecdsa::SigningKey>>;#[tokio::main]async fn main() -> Result<(), Box<dyn std::error::Error>> {    let provider: Provider<Http> = Provider::<Http>::try_from("http://127.0.0.1:8545")?; // ???RPC?? http://127.0.0.1:8545 // ?????? ??????????"0x"??    let wallet: LocalWallet = "a58ac7ea54b74896c6a3129370dea99140c42b6908dff628f9062b9d40da4b7e".parse::<LocalWallet>()? .with_chain_id(1281u64); // ?????ID // .with_chain_id(Chain::Moonbase); // ?????ID    let client = SignerMiddleware::new(provider.clone(), wallet.clone());    let address_from = "0x9295b5ca66CC2d8Abfa9024F482038A2D5Ff7Eaf".parse::<Address>()?;    let address_t
```

```

o = "0x108e35b0089e8069dae7e9fdc3a0353ac62a999e".parse::()?;
let block_number = provider.get_block_number().await?;
println!("??????: {}", block_number); // let
block = provider.get_block(81088u64).await?; // println!
("??????: {}", serde_json::to_string(&block)?); let ga
s_price = provider.get_gas_price().await?; println!("????
gas_price: {} Wei => {} Gwei", gas_price ,utils::format_unit
s(gas_price, "gwei").unwrap()); let none = provider.get_t
ransaction_count(address_from,None).await?; println!("???
???None?: {}", none ); send_transaction(&client, &addres
s_from, &address_to).await?; print_balances(&provider, &a
ddress_from, &address_to).await?; Ok({})} // ??????async
fn print_balances(provider: &Provider<Http>, address_from:
&Address, address_to: &Address) -> Result<(), Box<dyn std::e
rror::Error>> { let balance_from = provider.get_balance(a
ddress_from.clone(), None).await?; let balance_to = provi
der.get_balance(address_to.clone(), None).await?; println
!("{}", has {},eth:{}, address_from, balance_from, utils::for
mat_units(balance_from, "ether").unwrap()); println!("{}",
has {},eth:{}, address_to, balance_to, utils::format_units(
balance_to, 18).unwrap()); Ok({})} // ??????async fn send_t
ransaction(client: &Client, address_from: &Address, address_
to: &Address) -> Result<(), Box<dyn std::error::Error>> {
println!(
"?????1????? {} to {}.", address_fro
m, address_to ); let gas = U256::from(21000); let g
as_price = client.get_gas_price().await?; let tx = Transa
ctionRequest::new(
.to(address_to.clone())
.value(U256::from(utils::parse_ether(1)?))
.from(address
s_from.clone())
.gas(gas)
.gas_price(gas_price
); println!("??????: {}", serde_json::to_string(&tx)?);
let tx_hash = client.send_transaction(tx, None).await?.awa
it?; println!("??????: {}", serde_json::to_string(&tx_has
h)?); Ok({})}

```

构建并运行 Cargo 项目

使用 cargo run 一步构建并运行项目。

