2019-03-31工作记录-捐款按钮处理

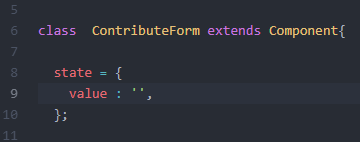
之前实现了查询详情页面，里面的捐款操作并没有实现，现在我们一起来编写捐款功能。

第一部分 实现捐款按钮功能

1 标准化操作， 把用户的输入存入状态变量中。通过Input控件的onChange事件设置状态值。修改文件：components/contributeForm.js:

import Campaign from '../ethereum/campaign';

import web3 from '../ethereum/web3';





增加函数：

//捐款按钮点击处理函数

onSubmit = async ()=>{

event.preventDefault();

//this.props就是由外部show.js调用ContributeForm传入的参数.

//Campaign(this.props.address)不存在这样的构造函数，只能理解为强制类型转换。把地址转换为Campaign类型

const campaign = Campaign(this.props.address);

const accounts = await web3.eth.getAccounts();

//注意这里采用ether单位，输入框中的数字是ether单位，需要转成wei

await campaign.methods.contribute().send({

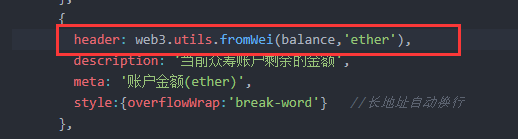
from : accounts[0],

value : web3.utils.toWei(this.state.value,'ether')

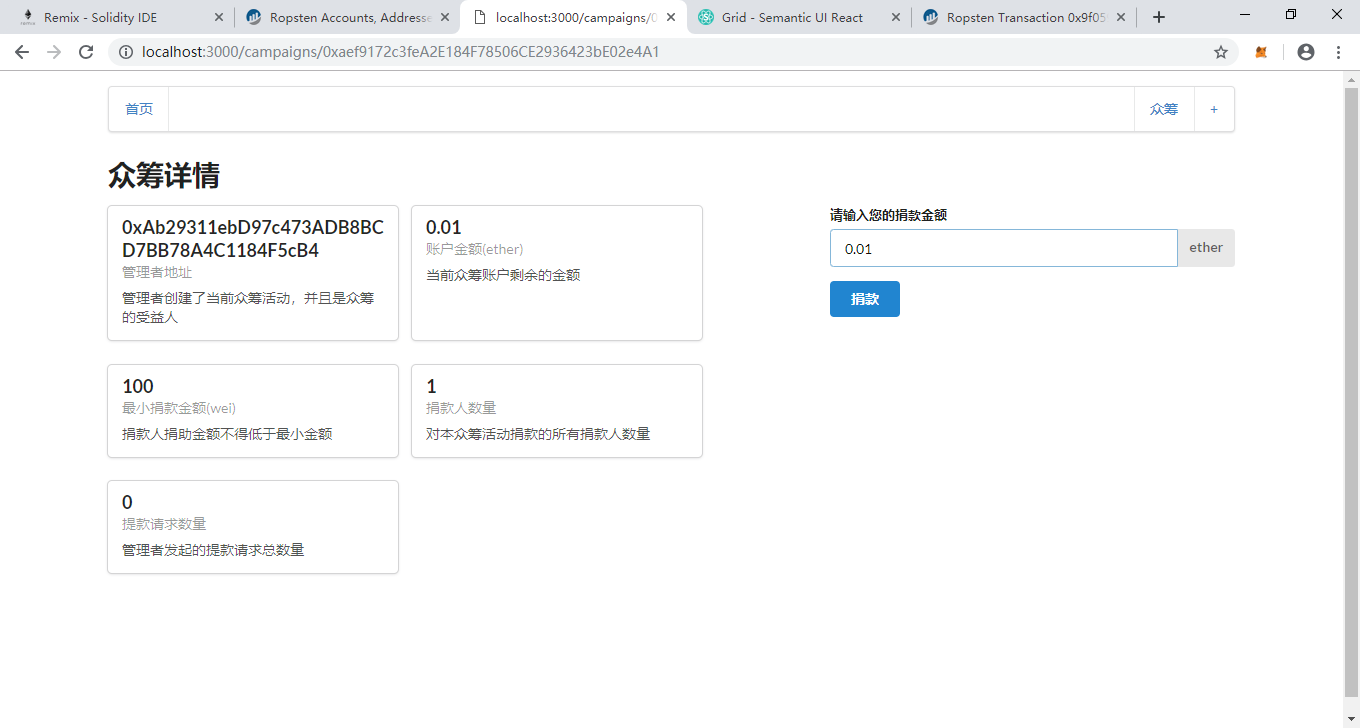
});

}

2 由于wei的单位太小，用户输入不方便，因此输入框换成ether单位。修改pages/campaigns/show.js:



3 完成后的实际界面。 输入0.01，点击“捐款”按钮，将会弹出metamask确认界面，等待一会成功后刷新页面，看到众筹账户金额增加了0.01。



第2部分 输入错误处理

仿照03-29工作记录中的对pages/campaigns/new.js的输入错误处理，修改contributeForm.js文件，内容如下：

import React, {Component} from 'react';

import { Form,Button,Input } from 'semantic-ui-react';

import {Message} from 'semantic-ui-react';

import Campaign from '../ethereum/campaign';

import web3 from '../ethereum/web3';

class ContributeForm extends Component{

state = {

value : '',

errmsg:'', //输入异常信息

loading:'', //加载中标记

};

//捐款按钮点击处理函数

onSubmit = async ()=>{

event.preventDefault();

this.setState({errmsg:''});

this.setState({loading:true}); //一旦提交就切换成加载中转圈圈...

if(this.state.value == '' ){

console.log("输入数字不允许为空！");

this.setState({errmsg:"输入数字不允许为空！"});

}else if(this.state.value \* 10\*\*18 <= 100){

console.log("最小捐款金额100wei！");

this.setState({errmsg:"低于最小捐款金额100wei！"});

}else{

try{

//this.props就是由外部show.js调用ContributeForm传入的参数.

//Campaign(this.props.address)不存在这样的构造函数，只能理解为强制类型转换。把地址转换为Campaign类型

const campaign = Campaign(this.props.address);

const accounts = await web3.eth.getAccounts();

//注意这里采用ether单位，输入框中的数字是ether单位，需要转成wei

await campaign.methods.contribute().send({

from : accounts[0],

value : web3.utils.toWei(this.state.value,'ether')

});

}catch(err){

this.setState({errmsg:err.message});

}

}

this.setState({loading:false}); //一旦提交完成恢复正常

}

render(){

return (

<Form onSubmit = {this.onSubmit} error={!!this.state.errmsg} >

<Form.Field>

<label>请输入您的捐款金额</label>

<Input

value = {this.state.value}

onChange = {event => this.setState({value:event.target.value})}

label='ether'

labelPosition='right' />

</Form.Field>

<Message error header='错误！' content={this.state.errmsg} />

<Button loading={this.state.loading} primary>捐款</Button>

</Form>

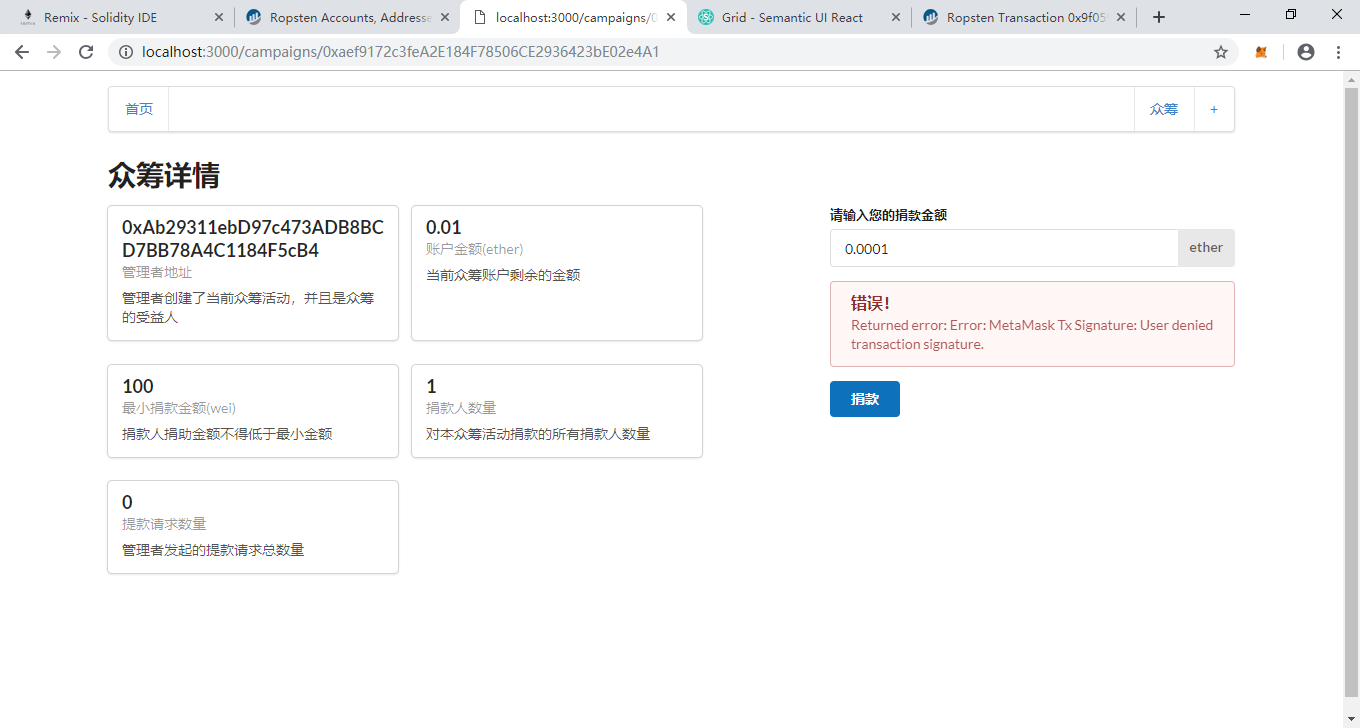
);

}

}

export default ContributeForm;

实际页面：



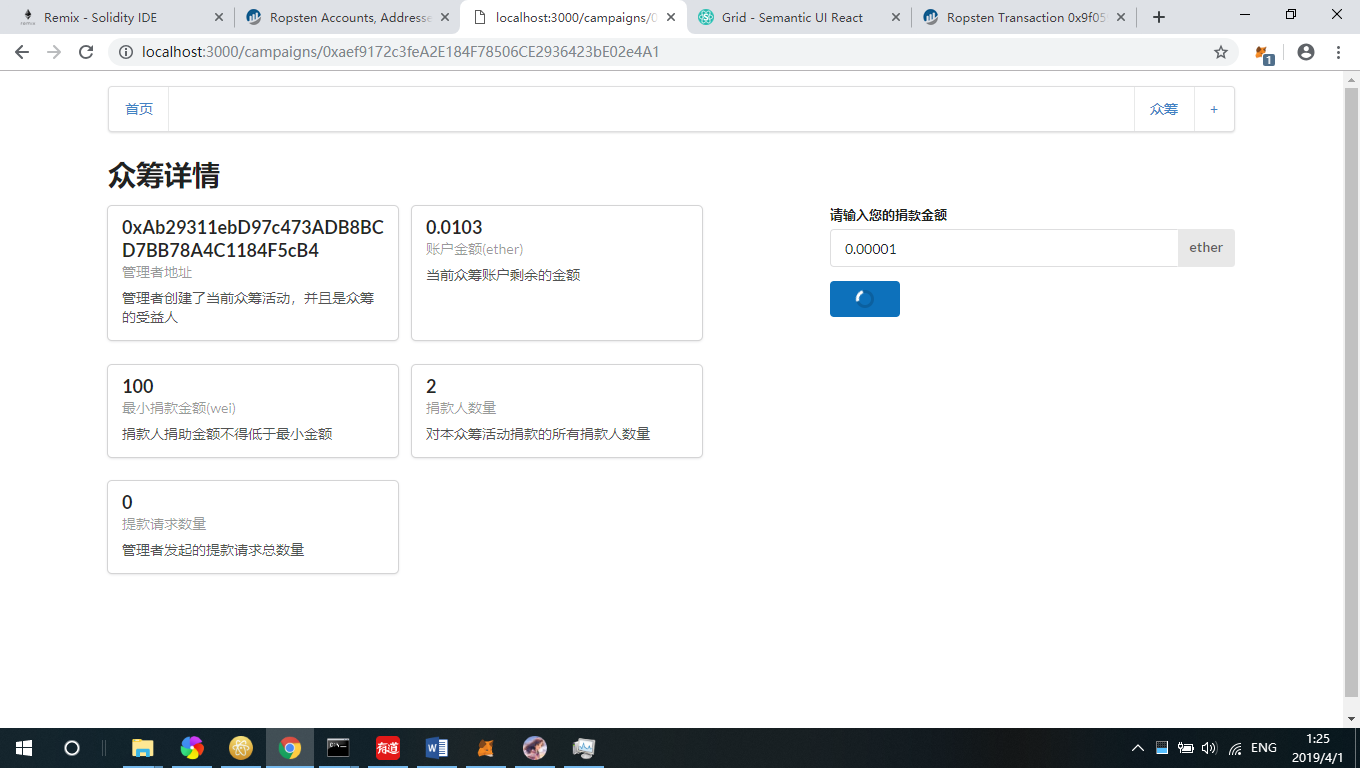
第三部分 捐款完成后自动刷新页面

上面的步骤实现了捐款操作，但是需要手动刷新下页面才能显示出新的账户金额，如何实现自动刷新呢？ 要用到Router的replaceRoute函数。

import {Router} from '../routes';



实际效果：捐款等待中…



捐款成功后，页面自动变化：

