区块链募捐系统

首页
发起募捐
所有募捐列表
我发起的
我捐赠的

募集金额数	捐款次数	项目数	募集成功次数
61	3	8	0

区块链募捐系统

首页
发起募捐
所有募捐列表
我发起的
我捐赠的

发起募捐

项目名称: 12321
目标金额: − 2 +
募捐截止时间: 🕐 2023-12-30 00:00:00
申请描述: 123
凭证: Click to upload
📄 Snipaste_2023-06-09_14-35-... ✓
凭证Hash: bd3e85bc329e1ff509d898b1243232335cd1f3892b9a2dbfc46436c2ba32e4ad

提交

MetaMask Notification — □ ×
? local
Account 2 → 0x7b37B...230...
http://localhost:14200
0x7b37B...23085 : 合约交互

详细信息 十六进制

🌐 建议的网站 > ⓘ
燃料 (估算) ⓘ   0.00750005
                    0.00750005 ETH
很可能在 15 秒 以内   最大费用: 0.00750005 ETH

共计   0.00750005
   **0.00750005 ETH**
金额 + 燃料费   最大金额: 0.00750005 ETH

拒绝   确认

区块链募捐系统

首页
发起募捐
所有募捐列表
我发起的
我捐赠的

项目名称 [项目名称] 查询

项目名称	发起者	目标金额	截止时间	操作
1	0x4C0108407439c6eD1835B211a9d6109EE716006B	1	2037-07-25 18:55:32	查看详情
1	0x4C0108407439c6eD1835B211a9d6109EE716006B	1	2037-07-25 18:55:32	查看详情
12321	0x4c0108407439c6eD1835B211a9d6109EE716006B	2	2023-12-30 00:00:00	查看详情
123	0x4C0108407439c6eD1835B211a9d6109EE716006B	1	2023-12-30 00:00:00	查看详情
3213	0x4C0108407439c6eD1835B211a9d6109EE716006B	2	2023-12-30 00:00:00	查看详情
3123	0x128F4233F4DdF26FbB8C73C5D1CC26C138dDfE17	2	2023-12-30 00:00:00	查看详情
222	0xE06b56d6C528429105b263005b67AbA5f5a8268c	1	2023-12-27 00:00:00	查看详情
111	0x128F4233F4DdF26FbB8C73C5D1CC26C138dDfE17	100	2023-12-30 00:00:00	查看详情
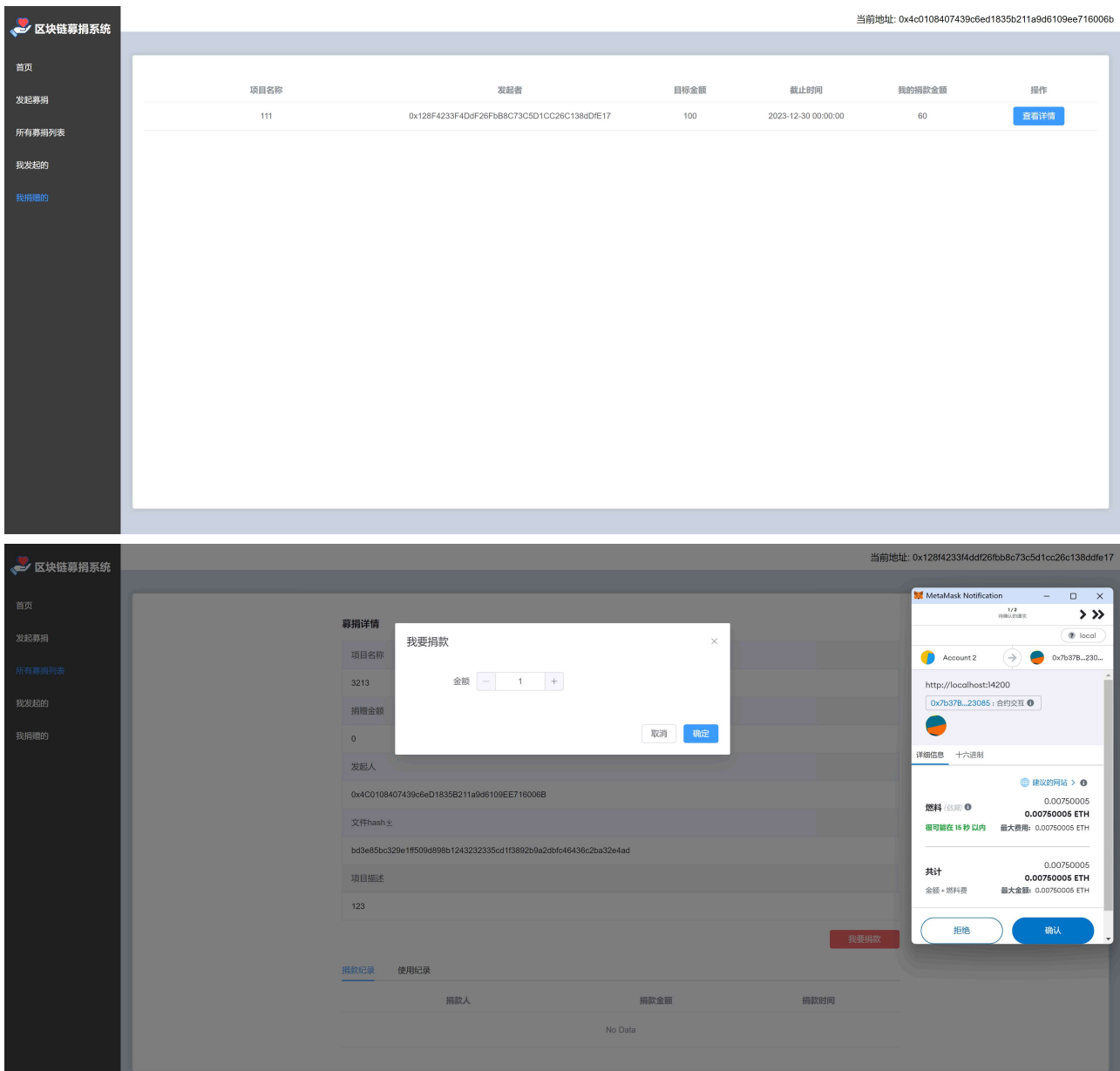
介绍

基于以太坊（Ethereum）的募捐系统可以利用智能合约和区块链技术来创建透明、安全和可追溯的募捐平台。这样的系统允许捐款者直接向特定的慈善机构或项目捐款，并确保资金的流向和使用情况可以公开审查

功能

发起募捐

- id
- 项目名称
- 目标金额
- 截止时间

- 申请描述
- 凭证hash
- 发起人

查看所有募捐项目

> 可使用项目名称进行查询

- id
- 项目名称
- 目标金额
- 截止时间
- 申请描述
- 凭证hash
- 发起人
- 项目状态

查看我发起的募捐项目

> 可使用项目名称进行查询

查看我捐赠的募捐项目

> 可使用项目名称进行查询

查看募捐详情

> - id

- 项目名称
- 目标金额
- 截止时间
- 申请描述
- 凭证hash
- 发起人
- 已募捐到的金额
- 已使用资金
- 捐赠人数

捐赠

- 金额

募捐纪录

- id
- 金额
- 捐赠时间

使用

- 申请使用原因

募捐使用纪录

- id
- 使用时间
- 申请使用原因

技术栈

> 前端调用小狐狸插件，进行快捷转账,后端保存支持的公链地址，和合约地址 abi

合约

- solidity
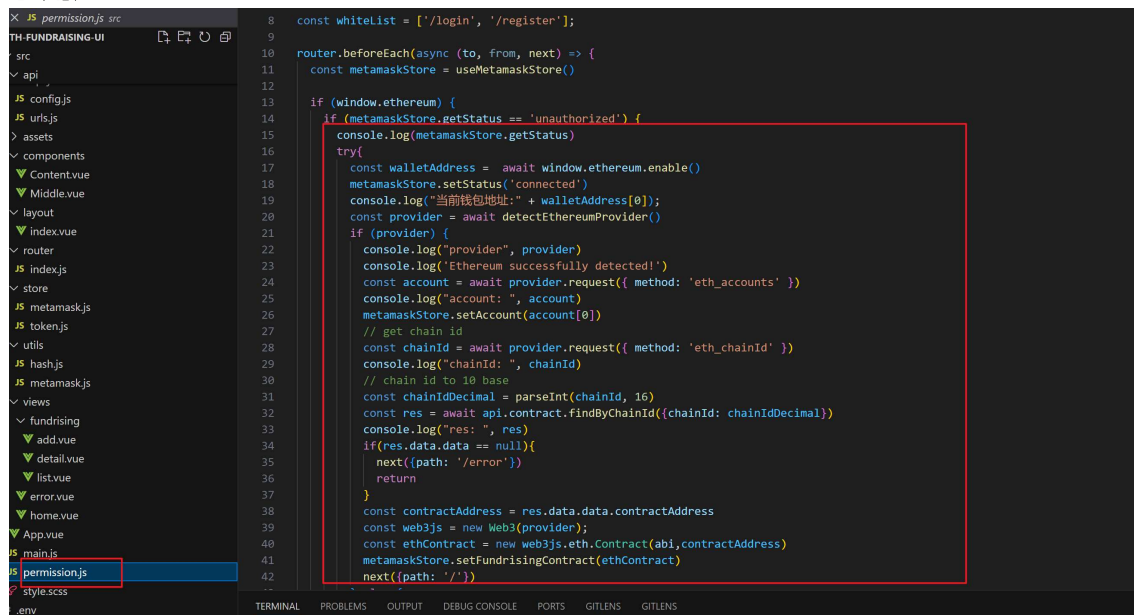
前端

- vue3
- vite
- pinia
- vue-router
- element-plus
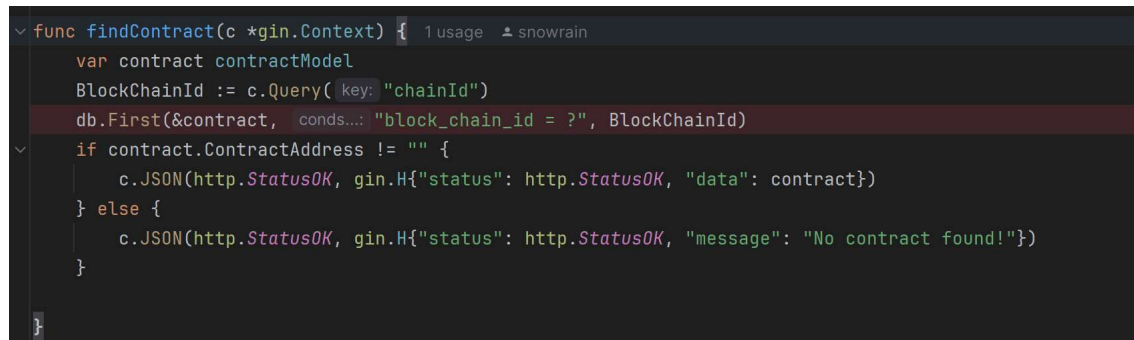- axios
- unocss

后端

- golang+gin

- mysql

详细设计

登陆

前端判断web3是否有注入，小狐狸插件启动会注入web3，相关api，使用对应api获取当前小狐狸激活的地址，区块链id。使用区块链id访问后端接口获取合约地址，然后初始化合约，保存激活地址。 如果没用web3注入会跳到error页面提示用户
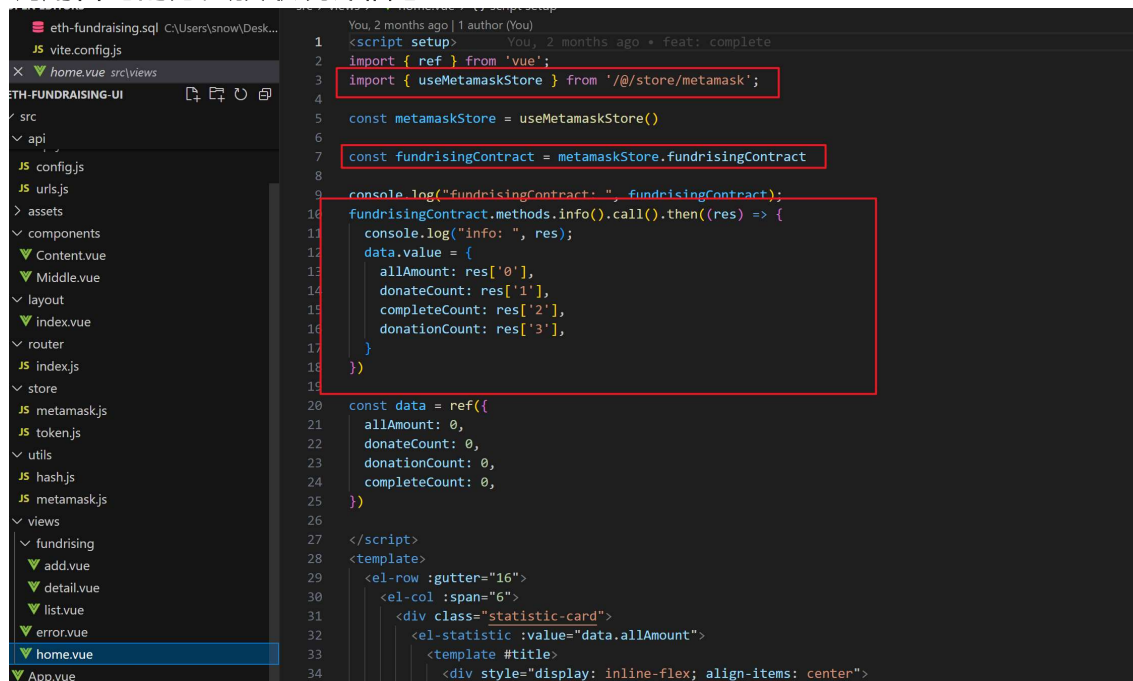


```js
const whiteList = ['/login', '/register'];

router.beforeEach(async (to, from, next) => {
  const metamaskStore = useMetamaskStore()

  if (window.ethereum) {
    if (metamaskStore.getStatus == 'unauthorized') {
      console.log(metamaskStore.getStatus)
      try{
        const walletAddress =  await window.ethereum.enable()
        metamaskStore.setStatus('connected')
        console.log("当前钱包地址:" + walletAddress[0]);
        const provider = await detectEthereumProvider()
        if (provider) {
          console.log("provider", provider)
          console.log('Ethereum successfully detected!')
          const account = await provider.request({ method: 'eth_accounts' })
          console.log("account: ", account)
          metamaskStore.setAccount(account[0])
          // get chain id
          const chainId = await provider.request({ method: 'eth_chainId' })
          console.log("chainId: ", chainId)
          // chain id to 10 base
          const chainIdDecimal = parseInt(chainId, 16)
          const res = await api.contract.findByChainId({chainId: chainIdDecimal})
          console.log("res: ", res)
          if(res.data.data == null){
            next({path: '/error'})
            return
          }
          const contractAddress = res.data.data.contractAddress
          const web3js = new Web3(provider);
          const ethContract = new web3js.eth.Contract(abi,contractAddress)
          metamaskStore.setFundrisingContract(ethContract)
          next({path: '/'})
```

```go
func findContract(c *gin.Context) {
    var contract contractModel
    BlockChainId := c.Query( key: "chainId")
    db.First(&contract, conds…: "block_chain_id = ?", BlockChainId)
    if contract.ContractAddress != "" {
        c.JSON(http.StatusOK, gin.H{"status": http.StatusOK, "data": contract})
    } else {
        c.JSON(http.StatusOK, gin.H{"status": http.StatusOK, "message": "No contract found!"})
    }
}
```
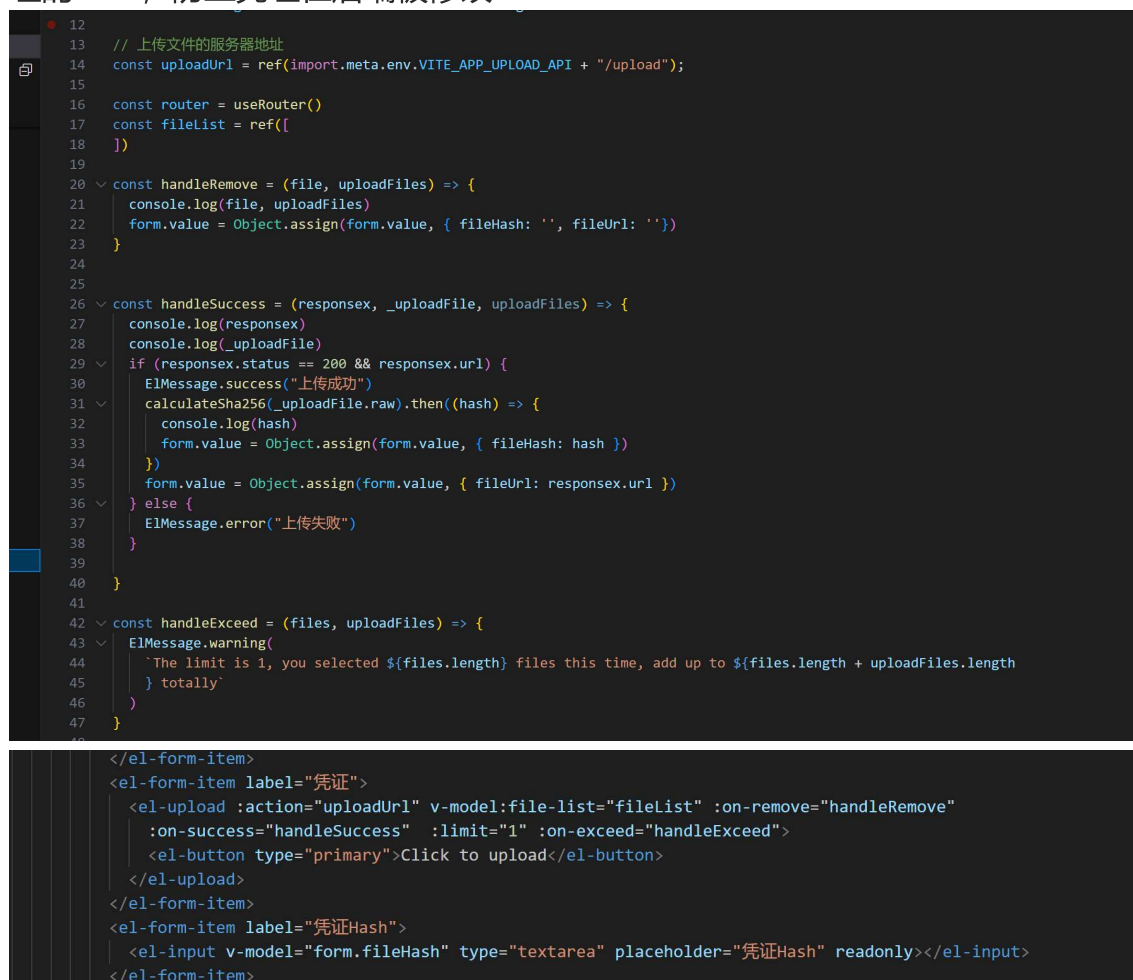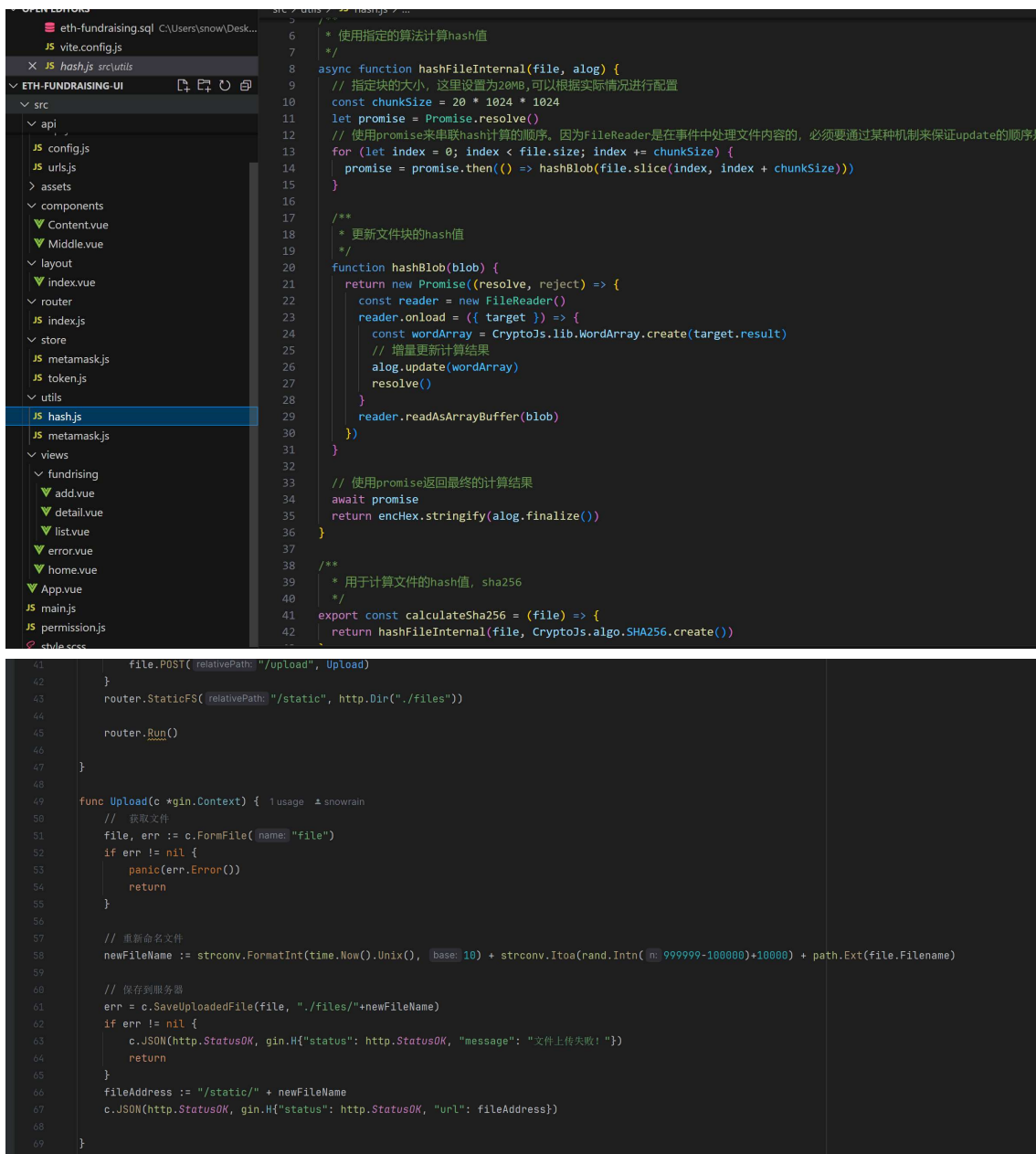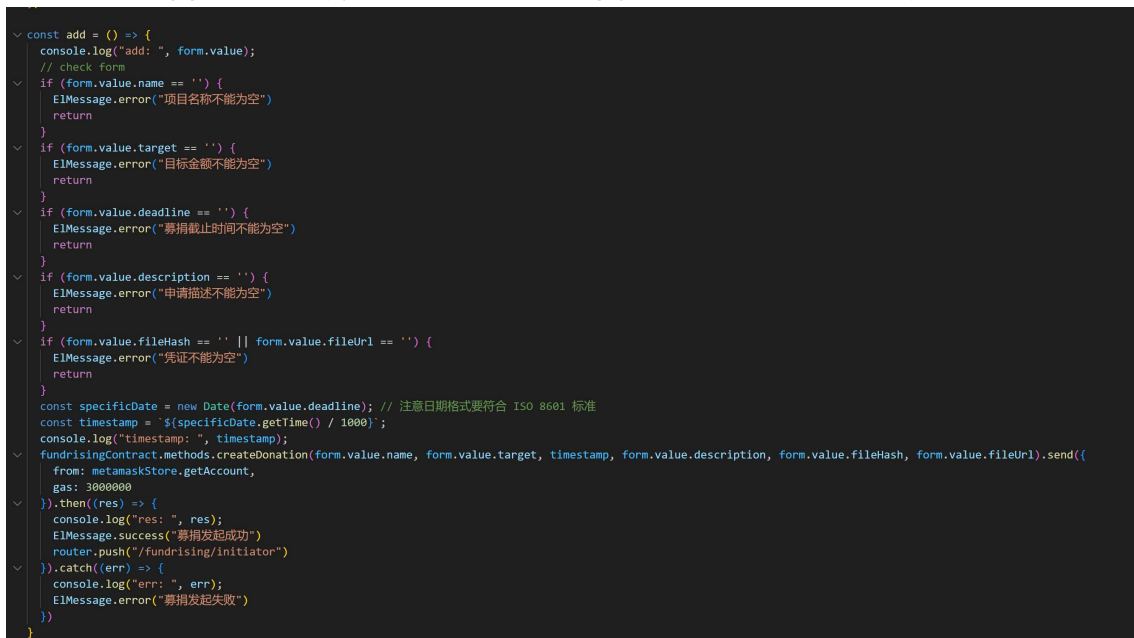
首页

调用合约的方法获取统计信息

```
1  <script setup>         You, 2 months ago • feat: complete
2  import { ref } from 'vue';
3  import { useMetamaskStore } from '/@/store/metamask';
4
5  const metamaskStore = useMetamaskStore()
6
7  const fundrisingContract = metamaskStore.fundrisingContract
8
9  console.log("fundrisingContract: ", fundrisingContract);
10 fundrisingContract.methods.info().call().then((res) => {
11   console.log("info: ", res);
12   data.value = {
13     allAmount: res['0'],
14     donateCount: res['1'],
15     completeCount: res['2'],
16     donationCount: res['3'],
17   }
18 })
19
20 const data = ref({
21   allAmount: 0,
22   donateCount: 0,
23   donationCount: 0,
24   completeCount: 0,
25 })
26
27 </script>
28 <template>
29   <el-row :gutter="16">
30     <el-col :span="6">
31       <div class="statistic-card">
32         <el-statistic :value="data.allAmount">
33           <template #title>
34             <div style="display: inline-flex; align-items: center">
```

发起募捐

上传凭证

凭证通过后端接口上传到后端同级目录 files 下，上传成功后前端进行计算得出凭证的hash，防止凭证在后端被修改

```
12
13   // 上传文件的服务器地址
14   const uploadUrl = ref(import.meta.env.VITE_APP_UPLOAD_API + "/upload");
15
16   const router = useRouter()
17   const fileList = ref([
18   ])
19
20   const handleRemove = (file, uploadFiles) => {
21     console.log(file, uploadFiles)
22     form.value = Object.assign(form.value, { fileHash: '', fileUrl: ''})
23   }
24
25
26   const handleSuccess = (responsex, _uploadFile, uploadFiles) => {
27     console.log(responsex)
28     console.log(_uploadFile)
29     if (responsex.status == 200 && responsex.url) {
30       ElMessage.success("上传成功")
31       calculateSha256(_uploadFile.raw).then((hash) => {
32         console.log(hash)
33         form.value = Object.assign(form.value, { fileHash: hash })
34       })
35       form.value = Object.assign(form.value, { fileUrl: responsex.url })
36     } else {
37       ElMessage.error("上传失败")
38     }
39
40   }
41
42   const handleExceed = (files, uploadFiles) => {
43     ElMessage.warning(
44       `The limit is 1, you selected ${files.length} files this time, add up to ${files.length + uploadFiles.length
45       } totally`
46     )
47   }
```

```
    </el-form-item>
    <el-form-item label="凭证">
      <el-upload :action="uploadUrl" v-model:file-list="fileList" :on-remove="handleRemove"
        :on-success="handleSuccess" :limit="1" :on-exceed="handleExceed">
        <el-button type="primary">Click to upload</el-button>
      </el-upload>
    </el-form-item>
    <el-form-item label="凭证Hash">
      <el-input v-model="form.fileHash" type="textarea" placeholder="凭证Hash" readonly></el-input>
    </el-form-item>
```

```
     * 使用指定的算法计算hash值
 7     */
 8  async function hashFileInternal(file, alog) {
 9      // 指定块的大小，这里设置为20MB,可以根据实际情况进行配置
10      const chunkSize = 20 * 1024 * 1024
11      let promise = Promise.resolve()
12      // 使用promise来串联hash计算的顺序。因为FileReader是在事件中处理文件内容的，必须要通过某种机制来保证update的顺序是
13      for (let index = 0; index < file.size; index += chunkSize) {
14          promise = promise.then(() => hashBlob(file.slice(index, index + chunkSize)))
15      }
16
17      /**
18       * 更新文件块的hash值
19       */
20      function hashBlob(blob) {
21          return new Promise((resolve, reject) => {
22              const reader = new FileReader()
23              reader.onload = ({ target }) => {
24                  const wordArray = CryptoJs.lib.WordArray.create(target.result)
25                  // 增量更新计算结果
26                  alog.update(wordArray)
27                  resolve()
28              }
29              reader.readAsArrayBuffer(blob)
30          })
31      }
32
33      // 使用promise返回最终的计算结果
34      await promise
35      return encHex.stringify(alog.finalize())
36  }
37
38  /**
39   * 用于计算文件的hash值, sha256
40   */
41  export const calculateSha256 = (file) => {
42      return hashFileInternal(file, CryptoJs.algo.SHA256.create())
```

```
41              file.POST( relativePath: "/upload", Upload)
42          }
43          router.StaticFS( relativePath: "/static", http.Dir("./files"))
44
45          router.Run()
46
47      }
48
49      func Upload(c *gin.Context) { 1 usage  ±snowrain
50          //  获取文件
51          file, err := c.FormFile( name: "file")
52          if err != nil {
53              panic(err.Error())
54              return
55          }
56
57          // 重新命名文件
58          newFileName := strconv.FormatInt(time.Now().Unix(), base: 10) + strconv.Itoa(rand.Intn( n: 999999-100000)+10000) + path.Ext(file.Filename)
59
60          // 保存到服务器
61          err = c.SaveUploadedFile(file, "./files/"+newFileName)
62          if err != nil {
63              c.JSON(http.StatusOK, gin.H{"status": http.StatusOK, "message": "文件上传失败！"})
64              return
65          }
66          fileAddress := "/static/" + newFileName
67          c.JSON(http.StatusOK, gin.H{"status": http.StatusOK, "url": fileAddress})
68
69      }
```

保存募捐纪录到区块链

调用区块链合约方法把数据保存到区块链中，保存成功则跳到我发起的募捐列表

```
const add = () => {
    console.log("add: ", form.value);
    // check form
    if (form.value.name == '') {
        ElMessage.error("项目名称不能为空")
        return
    }
    if (form.value.target == '') {
        ElMessage.error("目标金额不能为空")
        return
    }
    if (form.value.deadline == '') {
        ElMessage.error("募捐截止时间不能为空")
        return
    }
    if (form.value.description == '') {
        ElMessage.error("申请描述不能为空")
        return
    }
    if (form.value.fileHash == '' || form.value.fileUrl == '') {
        ElMessage.error("凭证不能为空")
        return
    }
    const specificDate = new Date(form.value.deadline); // 注意日期格式要符合 ISO 8601 标准
    const timestamp = `${specificDate.getTime() / 1000}`;
    console.log("timestamp: ", timestamp);
    fundrisingContract.methods.createDonation(form.value.name, form.value.target, timestamp, form.value.description, form.value.fileHash, form.value.fileUrl).send({
        from: metamaskStore.getAccount,
        gas: 3000000
    }).then((res) => {
        console.log("res: ", res);
        ElMessage.success("募捐发起成功")
        router.push("/fundrising/initiator")
    }).catch((err) => {
        console.log("err: ", err);
        ElMessage.error("募捐发起失败")
    })
}
```

募捐结束时间要大于当前时间+1小时


```
100          ElMessage.error("募捐发起失败")
101        })
102      }
103
104      const disabledDate = (time) => {
105        // time > now + 1h, use moment.js
106        return time.getTime() < moment().add(1, 'hours').valueOf();
107      }
108    </script>
109    <template>
110      <el-card class="min-h-full">
111        <div class="h-full min-h-full flex justify-center w-full">
112          <div class="w-100 text-center">
113            <h2>发起募捐</h2>
114            <el-form ref="formRef" :model="form" label-width="100px">
```

募捐列表

根据路由名称判断使用那个合约方法获取募捐列表


```
1    <script setup>        You, 2 months ago • feat: complete
43   const activeType = ref(currentRoute.value.meta.type)
44
45   if (activeType.value == 'initiator') {
46     fundrisingContract.methods.getMyInitiatedRecords().call({
47       from: metamaskStore.getAccount,
48     }
49   ).then((res) => {
50     console.log("getMyInitiatedRecords: ", res);
51     tableData.value = res.map((item) => {
52       return {
53         id: item.id,
54         name: item.projectName,
55         initiator: item.initiator,
56         target: item.targetAmount,
57         deadline: item.deadline,
58         donatedAmount: item.donatedAmount,
59
60         // state: item.state,
61       }
62     })
63     console.log("tableData: ", tableData.value);
64   })
65   } else if (activeType.value == 'donate') {
66     fundrisingContract.methods.getMyDonation().call(
67       {
68         from: metamaskStore.getAccount,
69       }
70   ).then((res) => {
71     console.log("getMyDonatedRecords: ", res);
72     tableData.value = res.map((item) => {
73       return {
74         id: item.id,
75         name: item.projectName,
76         initiator: item.initiator,
77         target: item.targetAmount,
78         deadline: item.deadline,
79         donatedAmount: item.donatedAmount,
80         donorAmount: item.donorAmount,
81         // state: item.state,
82       }
83     })
84     console.log("tableData: ", tableData.value);
85   })
86   } else {
87     findAll()
88
89   }
```

跳转到详情界面
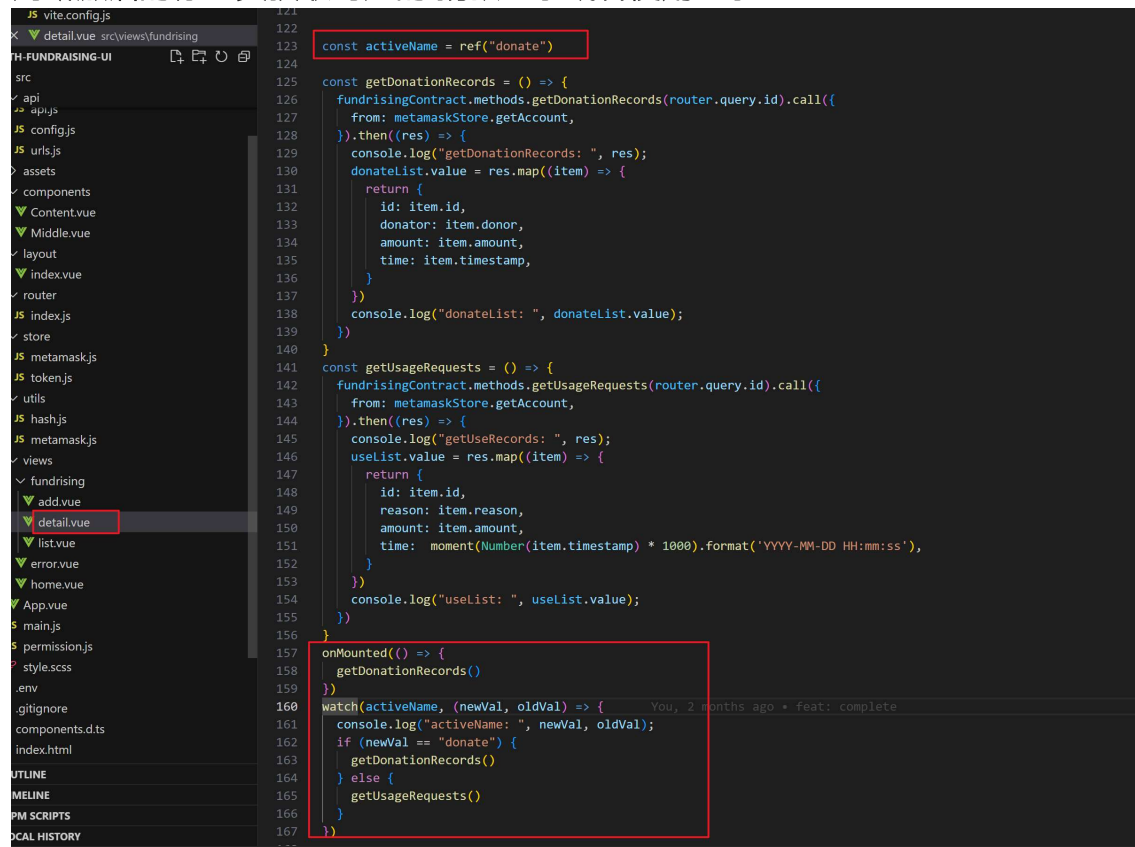
```
104
105
106    const handleDetail = (data) => {
107      console.log(data)
108      console.log(router)
109      router.push({
110        path: '/fundrising/detail',
111        query: {
112          id: data.id
113        }
114      })
115    }
116    </script>
117    <template>
118      <el-card class="min-h-full">
119        <div class="w-full">
120          <el-form :model="form" :inline="true" v-if="activeType == 'all'">
121            <el-form-item label="项目名称">
122              <el-input v-model="form.name" placeholder="项目名称"></el-input>
123            </el-form-item>
124            <el-form-item>
125              <el-button type="primary" @click="findAll">查询</el-button>
126            </el-form-item>
127          </el-form>
128          <div class="mt-5">
129            <el-table :data="tableData" style="width: 100%" :row-class-name="tableRowClassName">
130              <el-table-column prop="name" label="项目名称" align="center" />
131              <el-table-column v-if="activeType != 'initiator'" prop="initiator" label="发起者" align="center" />
132              <el-table-column prop="target" label="目标金额" width="200" align="center" />
133              <el-table-column v-if="activeType == 'initiator'" label="捐款金额" width="200" prop="donatedAmount" align="center" />
134              <el-table-column prop="deadline" label="截止时间" width="200" align="center" >
135                <template #default="scope">
136                  {{ moment(Number(scope.row.deadline) * 1000).format('YYYY-MM-DD HH:mm:ss') }}
137                </template>
138              </el-table-column>
139              <!-- <el-table-column label="状态" width="100" align="center" prop="state" /> -->
140              <el-table-column v-if="activeType == 'donate'" label="我的捐款金额" width="200" prop="donorAmount" align="center" />
141              <el-table-column label="操作" width="200" align="center">
142                <template #default="scope">
143                  <el-button type="primary" @click="handleDetail(scope.row)">查看详情</el-button>
144                </template>
145              </el-table-column>
146            </el-table>
```

募捐详情

初始化时调用合约获取当前募juju情

```
12      console.log('router: ', router.query);
13
14      const detail = ref({
15      })
16
17      const showDonate = ref(false)
18      const showUse = ref(false)
19      const getDetail = () => {
20
21        fundrisingContract.methods.getDontaionById(router.query.id).call({
22          from: metamaskStore.getAccount,
23        }).then(res => {
24          console.log("getDontaionById: ", res);
25          detail.value = {
26            name: res.projectName,
27            initiator: res.initiator,
28            target: res.targetAmount,
29            deadline: res.deadline,
30            description: res.description,
31            fileHash: res.proofHash,
32            fileUrl: res.proofUrl,
33            donatedAmount: res.donatedAmount,
34            donorCount: res.donorCount,
35            usedAmount: res.usedAmount,
36          }
37          showDonate.value = detail.value.initiator.toLowerCase() != metamaskStore.getAccount.toLowerCase() && detail.value.target != detail.value.donatedAmount
38          showUse.value = detail.value.initiator.toLowerCase() == metamaskStore.getAccount.toLowerCase() && detail.value.target != detail.value.usedAmount
39        }).catch(err => {
40          console.log("err: ", err);
41          ElMessage.error("获取详情失败")
42        })
43      }
44      onMounted(() => {
45        getDetail()
46      })
47
48
49      const handleDownload = () => {
```

Files sidebar:
```
src
 api
  api.js
  JS config.js
  JS urls.js
  assets
 components
  V Content.vue
  V Middle.vue
  index.vue
 router
  JS index.js
 store
  JS metamask.js
  JS token.js
 utils
  JS hash.js
  JS metamask.js
 views
  fundrising
   V add.vue
   V detail.vue
   V list.vue
  V error.vue
  V home.vue
 App.vue
 JS main.js
 JS permission.js
 style.scss
 .env
 .gitignore
 components.d.ts
```

判断激活的标签页获取对应的 捐款纪录或者使用纪录

```js
const activeName = ref("donate")

const getDonationRecords = () => {
  fundrisingContract.methods.getDonationRecords(router.query.id).call({
    from: metamaskStore.getAccount,
  }).then((res) => {
    console.log("getDonationRecords: ", res);
    donateList.value = res.map((item) => {
      return {
        id: item.id,
        donator: item.donor,
        amount: item.amount,
        time: item.timestamp,
      }
    })
    console.log("donateList: ", donateList.value);
  })
}
const getUsageRequests = () => {
  fundrisingContract.methods.getUsageRequests(router.query.id).call({
    from: metamaskStore.getAccount,
  }).then((res) => {
    console.log("getUseRecords: ", res);
    useList.value = res.map((item) => {
      return {
        id: item.id,
        reason: item.reason,
        amount: item.amount,
        time: moment(Number(item.timestamp) * 1000).format('YYYY-MM-DD HH:mm:ss'),
      }
    })
    console.log("useList: ", useList.value);
  })
}

onMounted(() => {
  getDonationRecords()
})
watch(activeName, (newVal, oldVal) => {
  console.log("activeName: ", newVal, oldVal);
  if (newVal == "donate") {
    getDonationRecords()
  } else {
    getUsageRequests()
  }
})
```

捐赠

调用合约的捐赠方法把捐赠金额转移到合约账号上

```js
const donaterForm = ref({
  amount: 0,
})

const openDonate = () => {
  donateVisiable.value = true
}

const handleDonate = () => {
  console.log("donate: ", donateForm);
  fundrisingContract.methods.donate(router.query.id).send({
    from: metamaskStore.getAccount,
    value: donateForm.value.amount,
  }).then((res) => {
    console.log("donate: ", res);
    ElMessage.success("捐款成功")
    donateVisiable.value = false
    activeName.value = "donate"
    getDetail()
    getDonationRecords()
  }).catch((err) => {
    console.log("err: ", err);
    ElMessage.error("捐款失败")
  }
  )
}
```

使用

调用合约的使用方法，把金额从合约账号上转到发起募捐人的账号上

```
100      useVisiable.value = true
101  }
102
103  ∨ const handleUse = () => {
104      console.log("use: ", useForm);
105  ∨   fundrisingContract.methods.requestUsage(router.query.id, useForm.value.reason, useForm.value.amount).send({
106          from: metamaskStore.getAccount,
107  ∨   }).then((res) => {
108          console.log("use: ", res);
109          ElMessage.success("申请成功")
110          useVisiable.value = false
111          activeName.value = "use"
112          getDetail()
113          getUsageRequests()
114  ∨   }).catch((err) => {
115          console.log("err: ", err);
116          ElMessage.error("申请失败")
117      }
118      )
119      useVisiable.value = false
120  }
121
122
```

```solidity
40        }
41
42        function requestUsage(
43            uint _donationId,
44            string memory _reason,
45            uint amount
46        ) public {
47            require(
48                msg.sender == donations[_donationId].initiator,
49                "Only initiator can request usage"
50            );
51            require(
52                donations[_donationId].donatedAmount -
53                    donations[_donationId].usedAmount >=
54                    amount,
55                "Donated amount must be greater than usage amount"
56            );
57            usageCount++;
58            donations[_donationId].usageRequestsArray.push(
59                UsageRequest({
60                    id: usageCount,
61                    donationId: _donationId,
62                    recipient: msg.sender,
63                    amount: amount,
64                    reason: _reason,
65                    time: block.timestamp
66                })
67            );
68
69            // 转移资金给使用人
70            payable(msg.sender).transfer(amount);
71            // 更新已使用资金
72            donations[_donationId].usedAmount += amount;
73        }
74
```