

星云链Nebulas——6.开发和部署星云链DApp

经过前面一系列的基础学习，本篇将进入DApp开发阶段。

1、创建星云链NAS钱包

在星云链上，部署智能合约需要星云币，所以需要申请一个星云链的钱包。

1.1 下载官方web钱包

注意：出于安全性的考虑，星云不架设专门Web钱包网站，以免私钥触网导致信息被盗。将钱包下载到自己的计算机上，断开网络，离线执行这些操作更为安全。

GitHub地址为：

```
https://github.com/nebulasio/web-wallet
```

```
wenzildeiMac:study wenzil$ git clone https://github.com/nebulasio/web-wallet
.git
Cloning into 'web-wallet'...
remote: Counting objects: 1592, done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 1592 (delta 34), reused 48 (delta 21), pack-reused 1528
Receiving objects: 100% (1592/1592), 1.93 MiB | 156.00 KiB/s, done.
Resolving deltas: 100% (1119/1119), done.
Checking out files: 100% (54/54), done.
```

web-wallet				
名称	修改日期	大小	种类	
check.html	今天 下午1:33	8 KB	HTML...ocument	
contract.html	今天 下午1:33	34 KB	HTML...ocument	
css	今天 下午1:33	--	文件夹	
img	今天 下午1:33	--	文件夹	
index.html	今天 下午1:33	4 KB	HTML...ocument	
js	今天 下午1:33	--	文件夹	
lib	今天 下午1:33	--	文件夹	
LICENSE	今天 下午1:33	8 KB	文本编辑 文稿	
README.md	今天 下午1:33	2 KB	Markdown	
sendNas.html	今天 下午1:33	15 KB	HTML...ocument	
sendOffline.html	今天 下午1:33	17 KB	HTML...ocument	
server.js	今天 下午1:33	922 字节	JavaScript	
viewWalletInfo.html	今天 下午1:33	7 KB	HTML...ocument	

双击 index.html打开，这时会启动浏览器，打开一个保存在本地的网页，即Web版钱包。



Mainnet 简体中文

[新建钱包](#)

[发送星云币](#)

[离线发送](#)

[钱包信息](#)

[交易状态](#)

[合约](#)

输入密码：（不少于9位数）

别忘了这个！

新建钱包

该密码用于加密您的私钥。

他不做为产生私钥的种子。

您需要该密码 + 您的私钥以解锁您的钱包。

设置密码(密码至少是9位数)，点击创建新钱包。

特别提醒: 请务必保证密钥文件的安全，不要忘记密码（私钥是您数字资产的凭证，请不要告诉任何人，如不慎暴露或丢失，会面临丢失对应数字资产的风险，请谨慎保管）。

右上方可以选择主网、测试网络还是私有网络，为了方便演示，这里选择了测试网络。

[新建钱包](#)[发送星云币](#)[离线发送](#)[钱包信息](#)[交易状态](#)[合约](#)

输入密码：(不少于9位数)

.....

[新建钱包](#)

保存你的 密码库文件。

[下载密码库文件](#)

该密码用于加密您的私钥。

他不做为产生私钥的种子。

您需要该密码 + 您的私钥以解锁您的钱包。

输入密码后，点击"新建钱包"按钮，然后点击"下载密码库文件"，会生成一个"n1"开头的文件，保存到电脑上，这里包含了这个钱包内的私钥信息，因此请妥善保管，避免遗失或被盗。

n1WYN8Nm1bpgBtqSXWh9XC

6PDJ2k5ZpPP6c.json

JSON - 504 字节

创建时间 2018年6月10日 下午1:56

修改时间 2018年6月10日 下午1:56

上次打开时间 --

注意：密码库文件可以用于断开网络，进行离线交易，这个钱包内的秘钥信息都包含在内，请妥善保管，如不慎丢失，会面临丢失对应数字资产的风险。

1.2 使用web钱包

点击"钱包信息"，然后上传钱包文件。

新建钱包

发送星云币

离线发送

钱包信息

交易状态

合约

选择你的钱包文件：

选择钱包文件...

解锁

上传成功后，输入创建钱包时的密码，点击"解锁"。

新建钱包

发送星云币

离线发送

钱包信息

交易状态

合约

选择你的钱包文件：

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c.json

你的钱包加密了，好！请输入密码。

.....

解锁

解锁成功后，会在当前页面出现类似如下页面，可以查看账户的地址、余额、私钥、二维码地址等信息。

你的地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6i

钱包余额

0 NAS

密码库文件 (JSON / Recommended / Encrypted)

下载

私钥 (未加密)

.....

☐ visible

你的地址



发现钱包余额为0，以下介绍如何获取测试网络的星云币。2018年5月22日，星云链推出了星云激励计划，在主网每提交一个有效DApp即可获得100NAS（注：提交的DApp需要官方审核，需要满足一定的条件，可以查看以下链接的内容）。当然，也可以用钱来购买主网的星云币，这里不作介绍，有兴趣的同学自行研究。

[星云激励计划DApp审核规则指南](#)

1.3 领取测试网络星云币

领取测试网络星云币网址：

<https://testnet.nebulas.io/claim/>

每天可以领取1NAS的测试币，领取的星云币只能在测试网使用，没有实际价值，与主网没有任何关系。

打开网址后，输入邮箱和钱包地址，然后"点击按钮进行验证"。



Testnet

Email:

7@qq.com

Wallet:

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

● 点击按钮进行验证



Claim Token (1 NAS per email per day)

Wallet:

Nebulas Wallet Address

Check Balance

然后，在"Check Balance"一栏输入钱包地址，再点击“Check Balance”按钮检查钱包余额。

Wallet:

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

Check Balance

成功的话，页面会弹出这样的提示。

Fetch State Successfully



Balance: 1.0000 Nas, Nonce: 0

说明已经成功领取了一个星云币NAS，接下来就可以用来部署和测试自己编写的智能合约了。

2、编写和部署智能合约

这里就不介绍编写和部署智能合约的内容了，直接进入编写部分。

这里以一个简单的"文章发布"DApp为例来讲解，主要的功能是任何区块链上的用户都可以发表一篇标题不重复的文章，如果文章标题被占用，就只能查看这篇文章。

2.1 编写智能合约

先创建一个该DApp的工程目录，暂且命名为"nebulas_article_publish"，并创建一个名为"article_publish.js"的智能合约文件，终端操作如下：

```
wenzildeiMac:~ wenzil$ cd /Users/wenzil/Desktop/study
wenzildeiMac:study wenzil$ pwd
/Users/wenzil/Desktop/study
wenzildeiMac:study wenzil$ mkdir nebulas_article_publish
wenzildeiMac:study wenzil$ cd nebulas_article_publish/
wenzildeiMac:nebulas_article_publish wenzil$ touch article_publish.js
```

然后用JavaScript编写一个智能合约的代码，整个代码如下：

```
'use strict'
```

```

var Article = function(text){
    if(text){
        var obj = JSON.parse(text);
        // 文章作者
        this.author = obj.author;
        // 文章标题
        this.title = obj.title;
        // 文章内容
        this.content = obj.content;
        // 发布者钱包地址
        this.address = obj.address;
        // 文章发布时间
        this.date = obj.date;
    }
};

Article.prototype = {
    toString : function(){
        return JSON.stringify(this)
    }
};

// 获取当前时间
function getCurrentTime() {
    var date = new Date();
    var seperator1 = "-";
    var seperator2 = ":";
    var month = date.getMonth() + 1;
    var strDate = date.getDate();
    if (month >= 1 && month <= 9) {
        month = "0" + month;
    }
    if (strDate >= 0 && strDate <= 9) {
        strDate = "0" + strDate;
    }
    var currentTime = date.getFullYear() + seperator1 + month + seperator1 + strDate
        + " " + date.getHours() + seperator2 + date.getMinutes()
        + seperator2 + date.getSeconds();
    return currentTime;
};

// 智能合约存储区
var ArticleContract = function () {
    // 使用"LocalContractStorage"绑定Map属性
    LocalContractStorage.defineMapProperty(this, "articleData", {
        stringify: function (obj) {
            return obj.toString();
        }
    });
};

```



```

    },
    parse: function (text) {
        return new Article(text);
    }
});
};

ArticleContract.prototype = {
    init: function(){
        //TODO:
    },

    /**
     * 根据文章标题或者对应的文章
     * author: 文章作者
     * title: 文章标题
     * content: 文章内容
     */
    save: function(author, title, content){
        if(!author || !title || !content){
            throw new Error("文章标题、作者和内容不能为空")
        }

        if(author.length > 20) {
            throw new Error("文章作者名字过长")
        }

        if(title.length > 100) {
            throw new Error("文章标题过长")
        }

        if(content.length > 100000){
            throw new Error("文章内容过长")
        }

        var from = Blockchain.transaction.from;
        var article = this.articleData.get(title);
        if(article){
            throw new Error("该文章已存在，请勿重新发布");
        }

        article = new Article();
        article.author = author;
        article.title = title;
        article.content = content;
        article.address = from;
        article.date = getCurrentTime();
        this.articleData.put(title, article);
    }
};

```

```
    },  
  
    /**  
     * 根据文章标题或者对应的文章  
     * title: 文章标题  
     */  
    get: function(title){  
        if(!title){  
            throw new Error("文章标题为空")  
        }  
        return this.articleData.get(title);  
    }  
}  
  
module.exports = ArticleContract;
```

2.2 部署智能合约

跟以太坊平台一样，在星云链上部署智能合约写入区块链中需要支付一定的费用（即Gas）。支付Gas的目的主要有如下几个：

- 1.保证节点正常运行防止蓄意攻击；
- 2.用于矿工的奖励；
- 3.防止恶意发布代码（如执行死循环操作）。

切换到Web钱包的"合约"选项，然后选择"部署"。

新建钱包

发送星云币

离线发送

钱包信息

交易状态

合约

搜索

部署

执行

部署合约

合约代码：

编程语言：☒ JavaScript ☐ TypeScript （请选择代码类型！）

参数 ?

["arg-1", "arg-2", ...]

选择你的钱包文件：

选择钱包文件...

解锁

来自地址

目的地址

将合约代码粘贴到输入框，点击"选择钱包文件"进行上传，输入钱包密码后再点击"解锁"，会出现如下内容。

选择你的钱包文件：

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c.json

你的钱包加密了，好！请输入密码。

.....

解锁

来自地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

目的地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

余额

1 NAS

要发送的价值 / 金额

0

Gas 限制

200000≈ 200 k

Gas 价格 (1 NAS = 1EWei = 10¹⁸ Wei)

1000000≈ 200 kWei

测试

提交

点击"测试"按钮后，发现没有返回错误信息。

测试

提交

测试结果

```
{ "result": "", "execute_err": "", "estimate_gas": "23197" }
```

提交结果

交易哈希：[\(点击产看交易详情\)](#)↓

48b0e3888e432d589bc7e5876f35a35bc8d7ada9796a563aeab01744a5a3c667

合约地址：

n1yE6rWrGo8yd1E9Wzn6CqZB85gyLF65YdL

然后点击右边的"提交"按钮进行部署，成功后会返回交易的哈希以及合约地址（记得做好保存）。

2.3 测试智能合约

切换到"执行"选项，函数输入要测试的名称，如"save"，参数这里传如下内容：

```
[ "Wenzil", "开发和部署星云链DApp", "经过前面一系列的基础学习，本篇将进入DApp开发阶段。\\n\\n在星云链上，部署智能合约需要星云币，所以需要申请一个星云链的钱包。。。" ]
```

然后跟之前一样，上传钱包文件，输入密码，点击"解锁"、"测试"和"提交"按钮一系列操作。

搜索

部署

执行

输入要测试的函数名

切换到"执行"选项

执行合约

函数 ?

save

参数 ?

["Wenzil","开发和部署星云链DApp","经过前面一系列的基础学习，本篇将进入DApp"]

选择你的钱包文件:

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c.json

你的钱包加密了，好！请输入密码。

.....

上传钱包文件，输入密码，进行解锁

解锁

来自地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

目的地址

n1yE6rWrGo8yd1E9Wzn6CqZB85gyLF65YdL

这里输入合约地址

余额

0.999999886163 NAS

要发送的价值 / 金额

0

Gas 限制

200000

Gas 价格 (1 NAS = 1EWei = 10¹⁸ Wei)

1000000

测试

提交

测试结果

表示测试成功

{ "result": "\\", "execute_err": "", "estimate_gas": "21017" }

提交结果

表示提交成功

交易哈希: (点击查看交易详情)↓

c5539c8452e14bceb14ac704014b6c26cd6bd6e31ba095565e1f048712968dbf

合约地址为：
n1yE6rWrGo8yd1E9Wzn6CqZB85gyLF65YdL

然后，可以测试get函数，如下

执行合约

函数 ?

get

参数 ?

["开发和部署星云链DApp"]

选择你的钱包文件：

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c.json

你的钱包加密了，好！请输入密码。

.....

解锁

来自地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c



目的地址

n1yE6rWrGo8yd1E9Wzn6CqZB85gyLF65YdL



余额

0.999999865145 NAS

要发送的价值 / 金额

0

Gas 限制

200000

Gas 价格 (1 NAS = 1EWei = 10¹⁸ Wei)

1000000

测试

提交

测试结果

```
{ "result": { "author": "Wenzil", "title": "开发和部署星云链DApp", "content": "经过前面一系列的基础学习，本篇将进入DApp开发阶段。\\n\\n在星云链上，部署智能合约需要星云币，所以需要申请一个星云链的钱包。。。", "address": "n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c", "date": "2018-06-10 9:32:30", "execute_err": "", "estimate_gas": "20249" }
```

提交结果

交易哈希：([点击查看交易详情](#))

07d01cbb63436a36ab7ddb5b4e4c575e338ed3b6ded394f28cba8973d4bda389

来到这里，表示已经成功部署合约并测试通过。

3、安装相关文件和插件

前端页面中需要用到两个js文件，分别是neb.js和nebPay.js，用于JavaScript中API的调用和支付接口的调用。

3.1 安装neb.js

neb.js提供了JavaScript开发的API接口，需要从GitHub下载。

1、clone项目neb.js

```
git clone https://github.com/nebulasio/neb.js.git
```

终端操作如下：

```
wenzildeiMac:study wenzil$ pwd
/Users/wenzil/Desktop/study
wenzildeiMac:study wenzil$ git clone https://github.com/nebulasio/neb.js.git
Cloning into 'neb.js'...
remote: Counting objects: 674, done.
remote: Compressing objects: 100% (95/95), done.
remote: Total 674 (delta 54), reused 64 (delta 20), pack-reused 557
Receiving objects: 100% (674/674), 1.05 MiB | 356.00 KiB/s, done.
Resolving deltas: 100% (377/377), done.
```

2、进入到clone项目的根目录，安装所有依赖

```
npm install
```

终端操作如下：

```
wenzildeiMac:study wenzil$ cd neb.js
wenzildeiMac:neb.js wenzil$ npm install
```

3、安装自动化构建工具gulp

```
npm install gulp
```

4、打包生成neb.js等文件

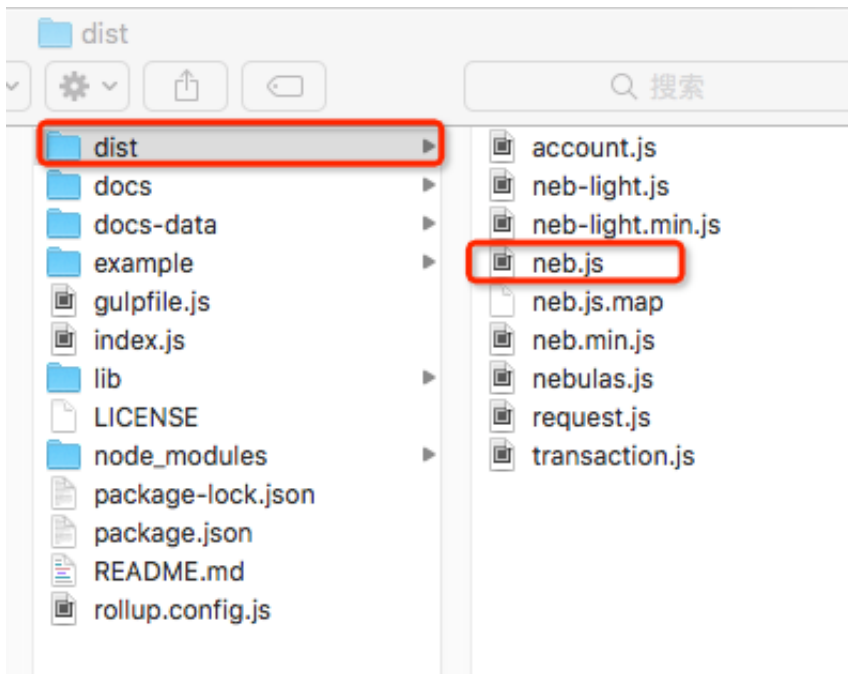
```
gulp
```

如果出现如下错误，用以下另外一种命令执行安装gulp

```
-bash: gulp: command not found

sudo npm install --global gulp
```

5、执行成功会生成"dist"文件夹，文件夹中有前端页面要使用的js文件。



3.2 安装nebPay.js

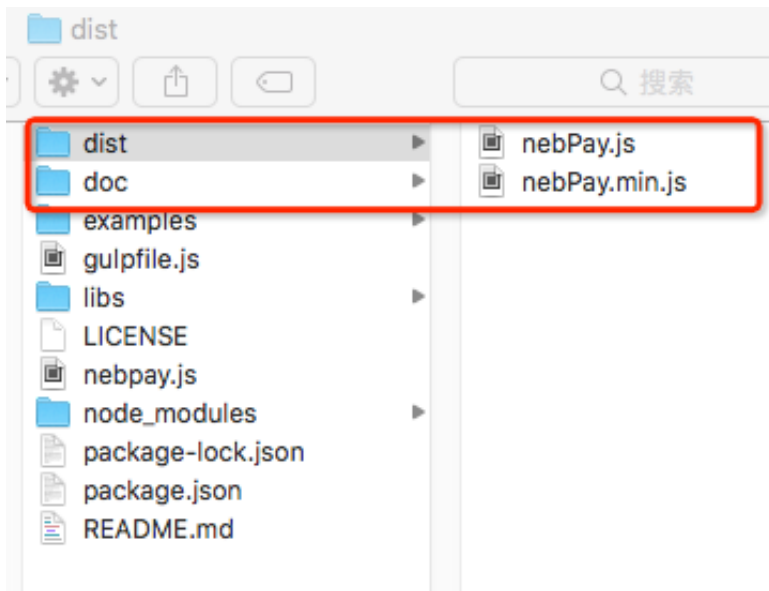
NebPay SDK为不同平台的交易提供了统一的支付接口，开发者在Dapp页面中使用NebPay API可以通过浏览器插件钱包、手机App钱包等实现交易支付和合约调用。

github地址：<https://github.com/nebulasio/nebPay>

安装方法同上，会生成nebPay.js文件

终端操作如下：

```
git clone https://github.com/nebulasio/nebPay
cd nebPay/
npm install
gulp
```

3.3 安装Chrome钱包插件

调用netPay的api需要唤醒Chrome钱包插件进行支付，因此需要安装钱包插件，查看GitHub有详细的安装说明

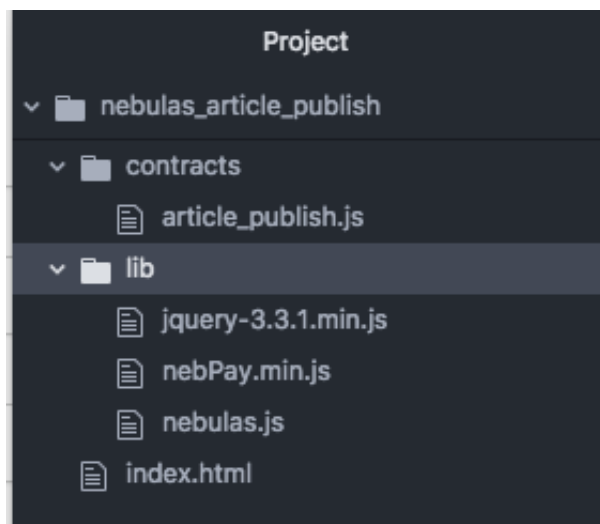
GitHub地址：

<https://github.com/ChengOrangeJu/WebExtensionWallet>

4、创建和运行DApp

4.1 创建DApp项目

前面创建了一个工程名为"my_web_wallet"的文件夹，将打包生成的nebular.js和nebPay.min.js文件放到lib文件夹（jquery-3.3.1.min.js也放到该文件夹），将article_publish.js放到contracts文件夹，新建index.html页面文件，这里用到Bootstrap框架开发简单的前端页面，工程结构如下：



index.html源码如下:

```
<!DOCTYPE html>
<html>
<head>
  <title>文章发布</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstrap.min.css" rel="stylesheet">
</head>

<body>
  <script type="text/javascript" src="./lib/nebulas.js"></script>
  <script type="text/javascript" src="./lib/nebPay.min.js"></script>
  <script type="text/javascript" src="./lib/jquery-3.3.1.min.js"></script>
  <script src="http://cdn.bootcss.com/bootstrap/3.3.1/js/bootstrap.min.js"></script>

  <div class="container">
    <div class="row clearfix">
      <div class="col-md-12 column">

        <h3>发布文章</h3>

        <form>
          <div class="form-group" style="margin-top:20px">
            <label for="author">文章作者: </label>
            <input id="input_author" type="text" class="form-control" id="author" placeholder="输入文章作者">

            <label for="title" style="margin-top:10px">文章标题: </label>
            <input id="input_title" type="text" class="form-control" id="title" placeholder="输入文章标题">

            <label for="content" style="margin-top:10px">文章内容: </label>
            <input id="input_content" type="text" class="form-control" id="content" placeholder="输入文章内容">
          </div>

          <div style="text-align:center;margin-top:20px">
            <button id="post" type="submit" class="btn btn-primary" style="padding:6px 16px">发布</button>
          </div>
        </form>

        <h3 style="margin-bottom:10px">查询文章</h3>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        <div class="input-group">
            <input type="text" class="form-control" placeholder="请输入文章标题"
id="text_search" />
            <span class="input-group-btn">
                <button id="button_search" class="btn btn-primary">搜索</button>
            </span>
        </div>

    </div>
    <div class="row clearfix">
        <div class="col-md-12 column">
            </div>
        </div>

    <div style="text-align:center;margin-top:30px">
        <h4 id="article_title"></h4>
    </div>
    <div style="text-align:center">
        <span id="article_author"></span><span id="article_date" style="margin-left:20px"></span>
    </div>
    <div id="article_content" style="margin-top:10px"></div>
</div>
</body>

<script>
"use strict";

// 智能合约的地址
var contactAddress = "n1yE6rWrGo8yd1E9Wzn6CqZB85gyLF65YdL";
// 访问星云链的API
var nebulas = require("nebulas"), Account = Account, neb = new nebulas.Neb
();
// 设置使用的网络, 主网为"https://mainnet.nebulas.io", 测试网络为
neb.setRequest(new nebulas.HttpRequest("https://testnet.nebulas.io"));

// NebPay SDK 为不同平台的交易提供了统一的支付接口
// 开发者在Dapp页面中使用NebPay API可以通过浏览器插件钱包、手机App钱包等实现交易支付
和合约调用。
var NebPay = require("nebpay");
var nebPay = new NebPay();

// 执行合约返回的交易流水号, 用于查询交易信息
var serialNumber;
$("#button_search").click(function () {
    if (!$("#text_search").val()) {
        alert('搜索文章标题不能为空');
    }
}

```

```

    return;
}
$('#article_content').text("");
var from = contactAddress
var value = "0"; // 金额
var nonce = "0"; // 交易序号
var gas_price = "1000000"; // 手续费价格
var gas_limit = "2000000"; // 手续费限制
var callFunction = "get";
var callArgs = "[" + $('#text_search').val() + "]";
console.log(callArgs);
var contract = { // 合约
    "function": callFunction, // 方法名
    "args": callArgs // 方法的参数
}

// 使用api.call执行合约中的方法
neb.api.call(from, contactAddress, value, nonce, gas_price, gas_limit, c
ontract).then(function (resp) {
    var result = resp.result;

    if (result === 'null') {
        $('#article_content').text("没有找到对应的文章");
        // 清空信息
        $('#article_title').text("");
        $('#article_author').text("");
        $('#article_date').text("");
        return;
    }
    console.log(result);
    result = JSON.parse(result);
    $('#article_title').text(result.title);
    $('#article_content').text(result.content);
    $('#article_author').text("文章作者: " + result.author);
    $('#article_date').text("发布时间: " + result.date);
}).catch(function (err) {
    console.log("error :" + err.message);
})
})
$('#post').click(function () {
    if (!$("#input_author").val() || !$("#input_title").val() || !$("#input_
content").val()) {
        alert('作者、标题、内容输入不能为空');
        return;
    }
    var to = contactAddress;
    var value = "0";
    var callFunction = "save";

```

```

var callArgs = "[\"" + $("#input_author").val() + "\",\"" + $("#input_title").val() + "\",\"" + $("#input_content").val() + "\"]";
console.log(callArgs);
serialNumber = nebPay.call(to, value, callFunction, callArgs, { //使用
nebpay的call接口去调用合约,
  listener: function (resp) {
    console.log("nebPay response:" + resp)
  }
});
})
</script>

</html>

```

4.2 运行DApp

直接打开index.html，界面如下：

发布文章

文章作者：

文章标题：

文章内容：

发布

查询文章

搜索

输入文章作者、标题、内容后，点击"发布"按钮，会弹出Chrome钱包插件。如果之前没有导入钱包文件，需要解锁。

 NEBULAS

Testnet ▾ 简体中文 ▾

新建钱包发送星云币Check-TX

选择你的钱包文件：

选择钱包文件...

解锁

来自地址

余额

≈ 0 NAS

目的地址要发送的价值 / 金额

地址金额≈ 0 NAS

Gas 限制Gas 价格

如果之前导入过钱包，会弹出如下信息，不需要手动填写任何内容。

 NEBULAS

Testnet ▾ 简体中文 ▾

新建钱包发送星云币Check-TX

Your wallet file is imported.

switch to a different wallet

来自地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

余额

0.999999844896

≈ 1 NAS

目的地址

要发送的价值 / 金额

n1yE6rWrGo8yd1l



0

≈ 0 NAS

Gas 限制

Gas 价格

200000

≈ 200 k

1000000

≈ 1 MWei

Nonce

8

Contract args

```
{"function": "save", "args": ["\\Wenzi\\", "\\测试标题\\", "\\测
```

生成交易

取消交易



[主页](#) [技术](#) [社区](#) [团队](#) [资源](#) [博客](#)

点击"生成交易"按钮，会弹出如下内容

交易信息

×

来自地址

n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c

目的地址

n1yE6rWrGo8yd1E9Wzn6CqZB85gyLF65YdL

要发送的价值 / 金额

0 ≈ 0 NAS

关闭

确认

再点击"确认"按钮，会弹出交易状态详情信息

交易状态

交易哈希可以查询交易信息，包含等待上链和已经打包的交易。在交易打包验证时需要多次刷新查询交易的打包状态变更。

fa9d50a65634e5ef327bf106b758672ca2aaf17a1t

关闭

交易详情

TxReceipt Status	success
金额	0 ≈ 0 Wei
Nonce	8
Gas 价格	1000000 1 MWei
Gas 限制	200000 ≈ 200 k
Gas 消耗	20628 21 kWei

过了一会，发现交易成功。
这时回到页面，输入提交时填写的文章标题来查询（即"测试标题"）来检查。

查询文章

测试标题

搜索

测试标题

文章作者：Wenzil 发布时间：2018-06-10 15:14:0

测试内容

发现，跟发布的内容一样，说明数据已经写入了区块链。当然，发布时间不是东八区的时间，时间格式也有点问题（转换时间格式的JS代码有点问题）。

当然，可以查看部署合约时创建的文章，同时也可以查看打印输出的内容。

查询文章

开发和部署星云链DApp

搜索

开发和部署星云链DApp

文章作者: Wenzil 发布时间: 2018-06-10 9:32:30

经过前面一系列的基础学习，本篇将进入DApp开发阶段。在星云链上，部署智能合约需要星云币，所以需要申请一个星云链的钱包。。。

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels Group similar

["src":"content","logo":"nebulas","serialNumber":"DLfSpGkeReqrVck40Z4JfPI5fGC8F99n","resp":{"txhash":"fa9d50a65634e5ef327bf106b758672ca2aaf17a1b2e0bc3d600404ea56c530b","contract_address":""}]

["测试标题"] index.html:154

["author":"Wenzil","title":"测试标题","content":"测试内容","address":"n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c","date":"2018-06-10 15:14:0"} index.html:172

["开发和部署星云链DApp"] index.html:154

["author":"Wenzil","title":"开发和部署星云链DApp","content":"经过前面一系列的基础学习，本篇将进入DApp开发阶段。\\n\\n在星云链上，部署智能合约需要星云币，所以需要申请一个星云链的钱包。。。","address":"n1WYN8Nm1bpgBtqSXWh9XC6PDJ2k5ZpPP6c","date":"2018-06-10 9:32:30"} index.html:172

这里运行的只是本地html文件，需要部署到服务器上，别人才能访问。该DApp功能比较简陋，还有很多需要完善的地方，这里不作继续谈论。

至此，整个星云链DApp开发流程已经走完了，可以去主网提交DApp了，哈哈。