

前言

主要代码(里面注释要细看!)

主要流程介绍

概述

leanCloud介绍

基本使用

第一步:首先创建一个应用

第二步:引入 av-min.js, 得到 window.AV对象

第三步:

测试代码的详细解读

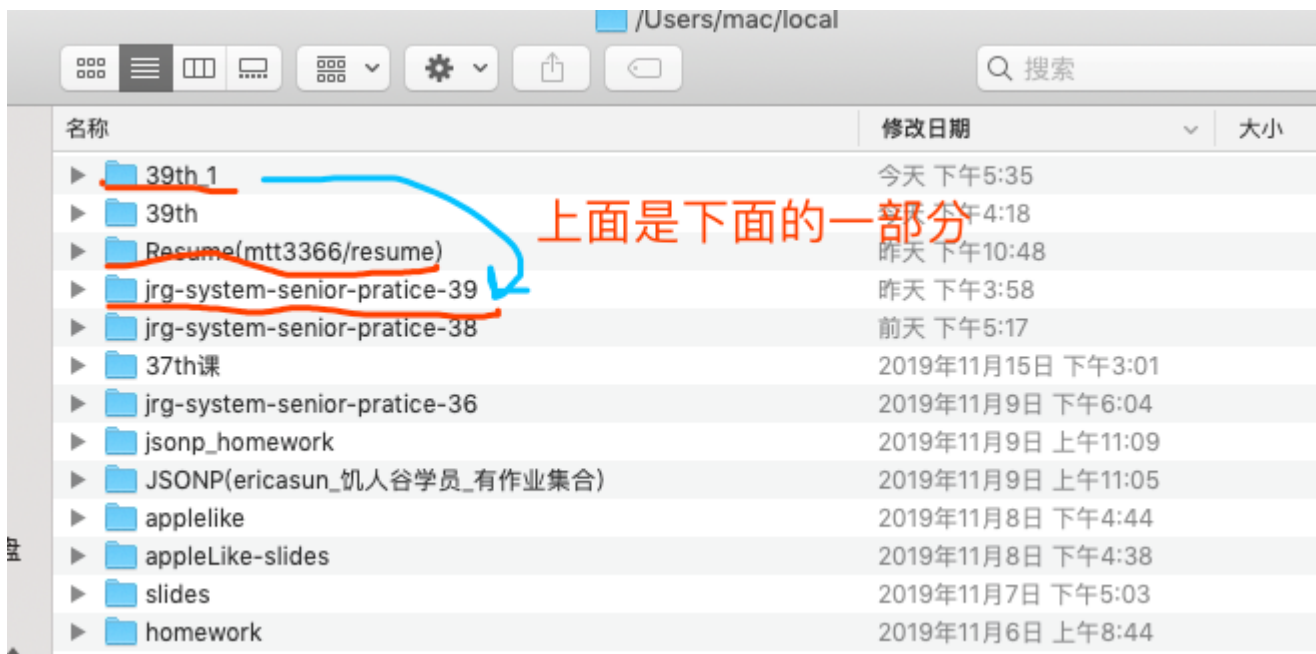
完成简历的留言功能

添加留言

显示留言的代码:

## 前言

这三个里面都是源代码:



<https://segmentfault.com/a/1190000015990451> 这篇文章不错

全部源代码:

<https://github.com/richard1230/39th>

## 主要代码(里面注释要细看!)

message.js

```
1 /**
2  * Created by Administrator on 2018/1/31 0031.
```

```
3  */
4
5  //初始化,这里是原来的
6  // var APP_ID = 'FqPJXxCv1gabU5JMeHrved6l-gzGzoHsz';
7  // var APP_KEY = 'Eevd4W01afMxVUg9s2pDDTr4';
8
9  // AV.init({
10 // appId: APP_ID,
11 // appKey: APP_KEY
12 // });
13 //下面的初始化是我自己的=在Learncloud上面添加的
14 AV.init({
15   appId: "yqzSiRS4ad0tl1IE2jvEBxsC-gzGzoHsz",
16   appKey: "sLwxvuc5C2s9XFSzeo18MRff",
17   // serverURLs: "api-uccloud.leancloud.cn",//最新版的里面这一行不用加,不然有错误!!
18 });
19 // console.log("运行到这里没有报错!!")
20
21 // //创建TestObject表
22 // var TestObject = AV.Object.extend('TestObject');
23
24 // //在表中创建一行数据
25 // var testObject = new TestObject();
26
27 // //数据内容是words: 'Hello World!'
28 // //如果保存成功,则运行alert('')
29 // testObject.save({
30 //   words: 'Hello World!'
31 // }).then(function(object) {
32 //   alert('LeanCloud Rocks!');
33 // })
34
35
36
37
38 //下面暂时先注释!!!
39 var query = new AV.Query('Message3');
40 query.find()
41   .then(
42     function (messages) {
43       let array = messages.map((item)=> item.attributes)
44       array.forEach((item)=>{
45         let li = document.createElement('li')
46         li.innerHTML = `${item.name}: ${item.content}`
47         let messageList = document.querySelector('#messageList')
48         messageList.appendChild(li)
```

```

49 })
50 }
51 )
52
53
54 let myForm = document.querySelector('#postMessage')
55
56 myForm.addEventListener('submit',function(e){
57 console.log(e)
58 e.preventDefault()
59 let name = myForm.querySelector('input[name=name]').value
60 let content = myForm.querySelector('input[name=content]').value
61 var Message = AV.Object.extend('Message3');
62 var message = new Message();
63
64 message.save({
65   'name': name,
66   'content':content
67 }).then(function(object){
68   let li = document.createElement('li')
69   li.innerHTML = `${object.attributes.name}: ${object.attributes.content}`
70   let messageList = document.querySelector('#messageList')
71   messageList.appendChild(li)
72   myForm.querySelector('input[name=content]').value = ''//清空表单里面的内容
73   console.log(object)
74 })
75 })
76
77
78
79
80
81
82
83
84

```

index.html

```

1 <!DOCTYPE html>
2 <head lang="en">
3 <meta charset="UTF-8">
4 <title>方方的个人简历</title>
5 <script src="//at.alicdn.com/t/font_451045_32zbfqegtq3t0529.js"></script>
6 <!--03-无刷新留言-->

```

```
7 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Swiper/4.1.0/css/
8 <link rel="stylesheet" href="main.css">
9 </head>
10
11 <body>
12 <div id="topNavBar" class="topNavBar">
13 <div class="topNavBar-inner clearfix">
14 <a class="logo" href="#" alt="logo" style="float: left;">
15 <span class="rs">RS</span>
16 <span class="card">card</span>
17 </a>
18 <nav class="menu" style="float: right;">
19 <ul class="clearfix">
20 <li>
21 <a href="#siteAbout">关于</a>
22 </li>
23 <li>
24 <a href="#siteSkills">技能</a>
25 </li>
26 <li>
27 <a href="#siteWorks">作品</a>
28 <ul class="submenu">
29 <li>作品1</li>
30 <li>作品2</li>
31 <li>作品3</li>
32 </ul>
33 </li>
34 <li>
35 <a href="#">博客</a>
36 <ul class="submenu">
37 <li>博客1</li>
38 <li>博客2</li>
39 <li>博客3</li>
40 </ul>
41 </li>
42 <li>
43 <a href="#">日历</a>
44 </li>
45 <li>
46 <a href="#">联系方式</a>
47 </li>
48 <li>
49 <a href="#">其他</a>
50 </li>
51 </ul>
52 </nav>
```

```
53 </div>
54 </div>
55 <div class="banner">
56 <div class="mask"></div>
57 </div>
58 <main>
59
60 <div data-x id="siteAbout" class="userCard">
61 <div class="pictureAndText clearfix">
62 <div class="picture">
63 
64 </div>
65 <div class="text">
66 <span class="welcome">Hello
67 <span class="triangle"></span>
68 </span>
69 <h1>richard</h1>
70 <p>前端开发工程师</p>
71 <hr>
72 <dl>
73 <dt>年龄</dt>
74 <dd>18</dd>
75 <dt>所在城市</dt>
76 <dd>北京</dd>
77 <dt>邮箱</dt>
78 <dd>huyongjinnuaa@163.com</dd>
79 <dt>手机</dt>
80 <dd>13812345678</dd>
81 </dl>
82 </div>
83 </div>
84 <footer class="media">
85 <a href="#">
86 <svg class="icon" aria-hidden="true">
87 <use xlink:href="#icon-github"></use>
88 </svg>
89 </a>
90 <a href="#">
91 <svg class="icon" aria-hidden="true">
92 <use xlink:href="#icon-Ankerwebicon-"></use>
93 </svg>
94 </a>
95 <a href="#">
96 <svg class="icon" aria-hidden="true">
97 <use xlink:href="#icon-weibo"></use>
98 </svg>
```

```
99 </a>
100 </footer>
101 </div>
102
103 <p class="downloadResume-wrapper">
104 <a class="downloadResume" href="./resume.pdf" target="_blank" download>下载 PDF 简历</a>
105 </p>
106
107 <p class="selfIntroduction">
108 richard, 资深前端工程师, 资深前端讲师, 现在在 饥人谷 教前端课程。
109 <br> 技能: 前端开发, Rails 开发, Node.js 开发
110 </p>
111
112 </main>
113 <section data-x id="siteSkills" class="skills">
114 <h2>技能</h2>
115 <ol class="clearfix">
116 <li>
117 <h3>
118 HTML 5 & CSS 3
119 </h3>
120 <div class="progressBar">
121 <div class="progress" style="width: 10%;"></div>
122 </div>
123 </li>
124 <li>
125 <h3>
126 JavaScript
127 </h3>
128 <div class="progressBar">
129 <div class="progress" style="width: 20%;"></div>
130 </div>
131 </li>
132 <li>
133 <h3>
134 jQuery
135 </h3>
136 <div class="progressBar">
137 <div class="progress" style="width: 30%;"></div>
138 </div>
139 </li>
140 <li>
141 <h3>
142 Vue.js
143 </h3>
144 <div class="progressBar">
```

```
145 <div class="progress" style="width: 40%"></div>
146 </div>
147 </li>
148 <li>
149 <h3>
150 React.js
151 </h3>
152 <div class="progressBar">
153 <div class="progress" style="width: 50%"></div>
154 </div>
155 </li>
156 <li>
157 <h3>
158 Node.js
159 </h3>
160 <div class="progressBar">
161 <div class="progress" style="width: 60%"></div>
162 </div>
163 </li>
164 </ol>
165 </section>
166 <section data-x class="portfolio" id="siteWorks">
167 <h2>作品集</h2>
168
169 <!--这就是view,看得见的东西-->
170 <!-- Slider main container -->
171 <div id="mySlides">
172 <div class="swiper-container">
173 <!-- Additional required wrapper -->
174 <div class="swiper-wrapper">
175 <!-- Slides -->
176 
177 
178 
179 </div>
180 <!-- If we need pagination -->
181 <div class="swiper-pagination"></div>
182
183 <!-- If we need scrollbar -->
184 <!--<div class="swiper-scrollbar"></div>-->
185 </div>
186
187 <!-- If we need navigation buttons -->
188 <div class="swiper-button-prev"></div>
189 <div class="swiper-button-next"></div>
190 </div>
```

```
191
192 </section>
193
194 <section class="message">
195 <h2>留言</h2>
196 <ol id="messageList"></ol>
197 <form id="postMessage" style="width:700px; margin:50px auto">
198 <label>姓名</label>
199 <input type="text" name="name">
200 <label>内容</label>
201 <input type="text" name="content">
202 <input type="submit" value="提交">
203 </form>
204 </section>
205
206 <!--模块化-->
207 <script src="https://cdnjs.cloudflare.com/ajax/libs/tween.js/17.1.1/Tween.min.js"></s
208 <script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/4.1.0/js/swiper.min.js"></
209 <!-- <script src="//cdn1.lnclld.net/static/js/3.5.0/av-min.js"></script> //这是原来的 -->
210 <script src="//cdn.jsdelivr.net/npm/leancloud-storage@3.15.0/dist/av-min.js"></script
211 <!-- 上面一行是自己加的 -->
212
213
214 <script src="js/init-swiper.js"></script>
215 <script src="js/auto-slide-up.js"></script>
216 <script src="js/sticky-topbar.js"></script>
217 <script src="js/smoothly-navigation.js"></script>
218 <script src="js/message.js"></script>
219 </body>
220
221 </html>
222
223
224
225
```

## 主要流程介绍

使用leanCloud给简历留言

### 概述

在本地预览项目的时候用的 `http-server`

前端的两大块:

操纵DOM:DOM API/jQuery



Ajax:http请求和响应/XMLHttpRequest的API

但是无法存储数据!

aliyun这种服务器太贵;

## leanCloud介绍

使用leancloud,提供简易有廉价的服务器服务.

这个leancloud服务会提供给你api,调用API可以简单操作后台,包括数据库和服务器.但是,只能用他的api.不能自己写后端代码.不需要在服务器部署数据库,省去在服务器上创建表等一切繁琐的后台操作.直接调用API就可以操纵数据库.

用它提供的API可以非常简单得创建表,保存数据等等,操作如后台一样.只是数据库和服务器都是这个平台帮你设置好的,你可以在前台直接发请求,操纵他的数据库.

只要会基本的js语法,了解http.使用他提供的API,就可以操作它提供的服务器了.是一个非常适用前端人员练习的产品

缺点:不安全!!!只能当练手用.前台就可以修改数据库代码.而且知道id之后,任何人任何地方都可以修改数据库

leanCloud

一个自带数据库和增删改查（CRUD）功能的后台系统。

拥有：

登录注册、手机验证码功能（收费）

存储任意信息

读取任意信息

搜索任意信息

删除任意信息

更新任意信息

等功能。

基本的增删改查功能

[javascript SDK 安装指南](#) 或者看这个:[js sdk 安装指南](#)

[数据存储开发指南. javascript](#)

## 基本使用

我在此演示一下基本的使用.

下面演示如何使用 LeanCloud 存一个 Hello World到他的后台数据库上.

分为下面几步:

创建一个应用 resume

引入 av-min.js, 得到 window.AV

初始化 AV 对象（代码直接拷）

新建一条数据来测试（代码直接拷,测试）

详细截图预与演示:

下面主要是看的这个:

[js sdk 安装指南](#)

## 第一步:首先创建一个应用



进入后:

resume-2018-01 开发版

结构化数据

用户

文件

LiveQuery

应用内搜索

离线分析

导入导出

用量统计

服务设置

创建 Class

Filter Classes

TestTable 4 objects

添加行 添加列 刷新 按条件过滤 批量操作

objectId	ACL	words	createdAt	updatedAt
5dd2a1a4844bb400753307e4	{\"a\":{\"read\":true,\"write\":true}}	Hello World!	2019-11-18 21:50:28	2019-11-18 21:50:28
5dd2a0e67d577441a5d8b063	{\"a\":{\"read\":true,\"write\":true}}	Hello World!	2019-11-18 21:47:18	2019-11-18 21:47:18
5dd29ecc21b47e00698a1822	{\"a\":{\"read\":true,\"write\":true}}	Hello World!	2019-11-18 21:38:20	2019-11-18 21:38:20
5dd29bdcfc36ed007162f199	{\"a\":{\"read\":true,\"write\":true}}	Hello World!	2019-11-18 21:25:48	2019-11-18 21:25:48

这里相当于数据库的表

这里代表4条记录

代表里面的记录数

界面的基础用途如图所示

注意创建的应用要等几分钟才能部署好,等几分钟,刷新页面知道应用出现在进行下面的步骤

第二步:引入 **av-min.js** , 得到 **window.AV** 对象

```
</section>

<!--模块化-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/tween.js/17.1.1/Tween.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/4.1.0/js/swiper.min.js"></script>
<!-- <script src="//cdn1.lnclld.net/static/js/3.5.0/av-min.js"></script> //这是原来的 -->
<script src="//cdn.jsdelivr.net/npm/leancloud-storage@3.15.0/dist/av-min.js"></script>
<!-- 上面一行是自己加的 -->

<script src="js/init-swiper.js"></script>
<script src="js/auto-slide-up.js"></script>
<script src="js/sticky-topbar.js"></script>
```

### JavaScript SDK 安装指南

- 获取 SDK
  - Web
    - 安装与引用 SDK
  - Node.js
  - 微信小程序
  - 微信小游戏
  - CocosCreator
  - LayaAir
  - Egret (白鹭引擎)
  - React Native
  - Electron
  - 初始化
  - 开启调试日志
  - 验证
  - 问题排查

展开所有 返回顶部

```
# 存储服务 (包括直播)
$ npm install leancloud-storage --save --registry=https://registry.npm.taobao.org
# 即时通讯服务
$ npm install leancloud-realtime --save --registry=https://registry.npm.taobao.org
```

然后在代码中通过 `require` 获得 SDK 的引用:

```
// 存储服务
var AV = require('leancloud-storage');
var { Query, User } = AV;
// 即时通讯服务
var { Realtime, TextMessage } = require('leancloud-realtime');
```

如果需要使用存储服务的 `LiveQuery` 功能, 需要按如下做法引入存储 SDK:

```
var AV = require('leancloud-storage/live-query');
// 有了这一行, 无需另外 require('leancloud-storage')
```

### CDN

你也可以直接在页面中通过 `script` 标签引入我们的 SDK:

```
<!-- 存储服务 -->
<script src="//cdn.jsdelivr.net/npm/leancloud-storage@3.15.0/dist/av-min.js"></script>
<!-- 即时通讯服务 -->
<script src="//cdn.jsdelivr.net/npm/leancloud-realtime@5.0.0-beta.0/dist/realtime.browser.min.js"></script>
```

## 引入存储服务的SDK

### 第三步:

拷贝初始化的代码:

### JavaScript SDK 安装指南

- 获取 SDK
- 初始化
- 开启调试日志
- 验证
- 问题排查

展开所有 返回顶部

### § 初始化

首先进入 [控制台](#) > [设置](#) > [应用 Key](#) 来获取 **App ID**, **App Key** 以及服务器地址。

如果是在前端项目里面直接通过 CDN 加载的 LeanCloud JavaScript SDK, 那么可以在页面加载的时候调用初始化的函数 `AV.init`:

```
AV.init({
  appId: "yqzSiRS4ad0t1lIE2jvEBxsC-gzGzoHsz",
  appKey: "sLwxvuc5C2s9XF5zeo18MRff",
  serverURLs: "https://xxx.example.com"
});
```

这行不要添加

选择应用

如果是使用 `npm` 安装的 SDK, 用法是一样的, 但是注意需要通过 `require` 获得 SDK 的引用:

```
12 // });
13 //下面的初始化是我自己的=在Learnc cloud上面添加的
14 AV.init({
15   appId: "yqzSiRS4ad0t1lIE2jvEBxsC-gzGzoHsz",
16   appKey: "sLwxvuc5C2s9XF5zeo18MRff",
17   // serverURLs: "api-uccloud.leancloud.cn", //最新版的里面这一行不用加, 不然有错误!!
18 });
19 // console.log("运行到这里没有报错!!")
20
21 // //创建TestObject表
```

这里的id和key每个应用都不一样.是专属的.

### 第四步:

测试,看看是否能够成功存储到服务器中的数据库.这一步仍然拷贝文档中的代码,按照文档中的提示做

```
root@user37:~/mac/local/55th_1
$PING api-ucloud.leancloud.cn
PING api-ucloud.leancloud.cn (106.75.75.67): 56 data bytes
64 bytes from 106.75.75.67: icmp_seq=0 ttl=48 time=40.323 ms
64 bytes from 106.75.75.67: icmp_seq=1 ttl=48 time=27.720 ms
64 bytes from 106.75.75.67: icmp_seq=2 ttl=48 time=50.527 ms
^Z
[4]+  Stopped                  PING api-ucloud.leancloud.cn
$
```

## JavaScript SDK 安装指南

获取 SDK

初始化

开启调试日志

验证

问题排查

展开所有 返回顶部

如果当前网路正常将会得到如下响应:

```
PING api-ucloud.leancloud.cn (123.59.41.31): 56 data bytes
64 bytes from 123.59.41.31: icmp_seq=0 ttl=51 time=9.032 ms
64 bytes from 123.59.41.31: icmp_seq=1 ttl=51 time=7.290 ms
64 bytes from 123.59.41.31: icmp_seq=2 ttl=51 time=8.131 ms
64 bytes from 123.59.41.31: icmp_seq=3 ttl=51 time=9.689 ms
64 bytes from 123.59.41.31: icmp_seq=4 ttl=51 time=6.559 ms
64 bytes from 123.59.41.31: icmp_seq=5 ttl=51 time=8.665 ms
64 bytes from 123.59.41.31: icmp_seq=6 ttl=51 time=8.041 ms
64 bytes from 123.59.41.31: icmp_seq=7 ttl=51 time=8.203 ms
64 bytes from 123.59.41.31: icmp_seq=8 ttl=51 time=6.288 ms
64 bytes from 123.59.41.31: icmp_seq=9 ttl=51 time=7.938 ms

--- api-ucloud.leancloud.cn ping statistics ---
 10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 6.288/7.984/9.689/0.997 ms
```

然后在项目中编写如下测试代码:

```
var TestObject = AV.Object.extend('TestObject');
var testObject = new TestObject();
testObject.set('words', 'Hello world!');
testObject.save().then(function (testObject) {
  console.log('保存成功。')
})
```

保存后运行程序

创建 Class	存储 / CLASS	数据	权限	性
Filter Classes	TestObject 1 object			
	添加行 添加列 刷新 按条件过滤 批量操作 ...			
	<input type="checkbox"/> objectId STRING	ACL ACL	words STRING	createdAt DATE
	<input type="checkbox"/> 5dd262cb7d577441a5d614d9	({"*":{"read":true,"write":true}})	Hello World!	2019-11-18 17:22:19
				2019-11-18 17:22:19
_Conversation	0			
_File	0			
_Followee	0			
_Follower	0			
_Installation	0			
_Role	0			
_User	0			
Message	0			
Message3	3			
TestObject	1			

Hello Word!就是在数据库中新插入的数据

## 测试代码的详细解读

CRM学习代码法中C(拷贝)和R(运行)完毕,接下来实行M(修改),修改代码

下面用我的项目再重新看一下整个过程:

首先引入SDK,再引入一个message.js模块用来写添加数据到数据库的代码

```
</section>

<!--模块化-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/tween.js/17.1.1/Tween.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/4.1.0/js/swiper.min.js"></script>
<!-- <script src="//cdn1.lnld.net/static/js/3.5.0/av-min.js"></script> //这是原来的 -->
<script src="//cdn.jsdelivr.net/npm/leancloud-storage@3.15.0/dist/av-min.js"></script>
<!-- 上面一行是自己加的 -->

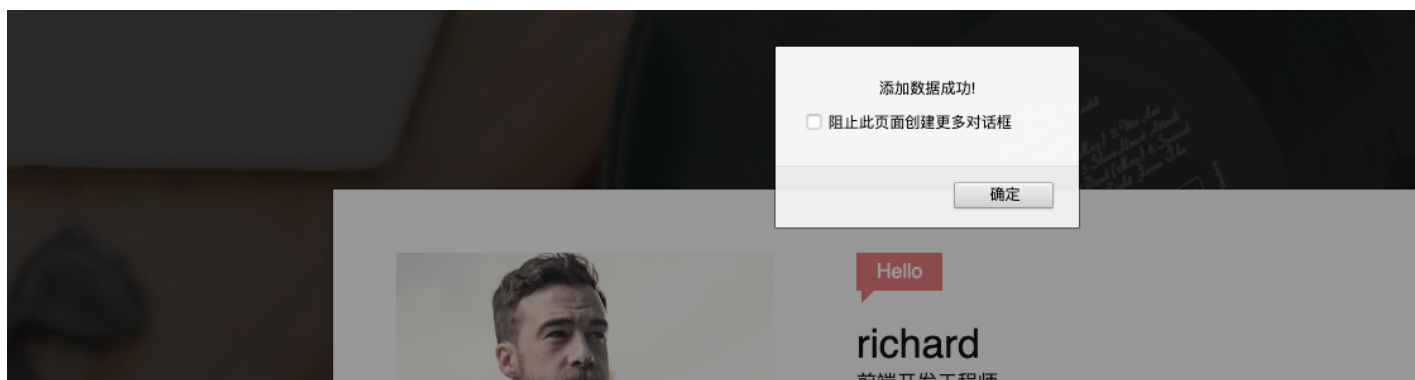
<script src="js/init-swiper.js"></script>
<script src="js/auto-slide-up.js"></script>
<script src="js/sticky-topbar.js"></script>
<script src="js/smoothly-navigation.js"></script>
<script src="js/message.js"></script>
</body>

</html>
```

```

9 // AV.init({
10 //     appId: APP_ID,
11 //     appKey: APP_KEY
12 // });
13 //下面的初始化是我自己的=在Learncoud上面添加的
14 AV.init({
15     appId: "yqzSiRS4ad0tl1IE2jvEBxsC-gzGzoHsz",
16     appKey: "sLwxvuc5C2s9XFSzeo18MRff",
17     serverURLs: "api-ucloud.leancloud.cn", //最新版的里面这一行不用加, 不然有错误!!
18 });
19 console.log("运行到这里没有报错!!")
20
21 //创建TestObject表
22 var TestObject = AV.Object.extend('TestTable'); //TestTable 是表的名字
23
24 //在表中创建一行数据, 创建表TestObject的一个对象testObject,
25 var testObject = new TestObject();
26
27 //数据内容是words: 'Hello World!'
28 //如果保存成功, 则运行alert('')
29 testObject.save({
30     words: 'Hello World!'
31 }).then(function(object) {
32     alert('LeanCloud Rocks!');
33     alert('添加数据成功!');
34 });
35
36 console.log("运行到这里没有报错!!")

```



网络
 样式编辑器
 性能
 内存
 存储
 无障碍环境

message.js X

```

1 /**
2  * Created by Administrator on 2018/1/31 0031.
3  */
4
5 //初始化, 这里是原来的
6 // var APP_ID = 'FqPJXxCv1gabU5JMeHrved6l-gzGzoHsz';
7 // var APP_KEY = 'Eevd4W01afMxVUg9s2pDDTr4';
8
9 // AV.init({
10 //     appId: APP_ID,
11 //     appKey: APP_KEY
12 // });
13 //下面的初始化是我自己的=在Learncoud上面添加的
14 AV.init({
15     appId: "yqzSiRS4ad0tl1IE2jvEBxsC-gzGzoHsz",
16     appKey: "sLwxvuc5C2s9XFSzeo18MRff",
17     serverURLs: "api-ucloud.leancloud.cn", //最新版的里面这一行不用加, 不然有错误!!
18 });
19 console.log("运行到这里没有报错!!")
20
21 //创建TestObject表
22 var TestObject = AV.Object.extend('TestTable'); //TestTable 是表的名字
23
24 //在表中创建一行数据, 创建表TestObject的一个对象testObject,
25 var testObject = new TestObject();
26
27 //数据内容是words: 'Hello World!'
28 //如果保存成功, 则运行alert('')
29 testObject.save({
30     words: 'Hello World!'
31 }).then(function(object) {
32     alert('LeanCloud Rocks!');
33     alert('添加数据成功!');
34 });
35
36 console.log("运行到这里没有报错!!")

```

创建 Class

IF

▼

Filter Classes

\_Conversation

0

\_File

0

\_Followee

0

\_Follower

0

\_Installation

0

\_Role

0

\_User

0

Message

0

Message3

3

TestObject

1

TestTable

2

存储 / CLASS

TestTable 2 objects

数据 权限 性能与索引

添加行

添加列

刷新

按条件过滤

批量操作

...

objectid	STRING	ACL	ACL	words	STRING	createdAt	DATE	updatedAt	DATE
5dd29ecc21b47e00698a1822		{*:*:{read:true,write:true}}		Hello World!		2019-11-18 21:38:20		2019-11-18 21:38:20	
5dd29bdcfc36ed007162f199		{*:*:{read:true,write:true}}		Hello World!		2019-11-18 21:25:48		2019-11-18 21:25:48	

可以发现:  
这里的object其实就是上面的testObject

```

30 words: 'Hello World!'
31 }).then(function(object) {
32   alert('LeanCloud Rocks!');
33   alert(object);
34
35 })
36 console.log(object);
37
38
39
40
41
42 //下面暂时
43 // var query = new LeanCloud.Query(TestTable);
44 // query.equalTo('words', 'Hello World!');
45 // query.find().then(function(results) {
46 //   //
47 //   //
48 //

```

▶ \_allPreviousSaves: {...}  
 ▶ \_changing: false  
 ▶ \_escapedAttributes: {}  
 ▶ \_flags: {}  
 ▶ \_hasData: true  
 ▶ \_hashedJSON: {}  
 ▶ \_opSetQueue: (1) [...]  
 ▶ \_pending: {}  
 ▶ \_previousAttributes: {...}  
 ▶ \_saving: 0  
 ▶ \_serverData: {...}  
 ▶ \_silent: {}  
 ▶ attributes: {...}  
 ▶ changed: {}  
 ▶ cid: "c1"  
 ▶ createdAt: Date Mon Nov 18 2019 21:50:28 GMT+0800 (中国标准时间)

```

28 // 如果保存成功, 则运行alert('');
29 testObject.save({
30   words: 'Hello World!'
31 }).then(function(object) {
32   alert('LeanCloud Rocks!');
33   alert('添加数据成功!');
34
35 })

```

▶ \_allPreviousSaves: {...}  
 ▶ \_changing: false  
 ▶ \_escapedAttributes: {}  
 ▶ \_flags: {}  
 ▶ \_hasData: true  
 ▶ \_hashedJSON: {}  
 ▶ \_opSetQueue: (1) [...]  
 ▶ \_pending: {}  
 ▶ \_previousAttributes: {...}  
 ▶ \_saving: 0  
 ▶ \_serverData: {...}  
 ▶ \_silent: {}  
 ▶ attributes: {...}  
 ▶ changed: {}  
 ▶ cid: "c1"  
 ▶ createdAt: Date Mon Nov 18 2019 21:50:28 GMT+0800 (中国标准时间)

## 完成简历的留言功能

两个功能:  
用户可以添加留言

留言可以显示

批量获取对象的api(一个对象指的就是数据库里面的一条记录)

## 添加留言

姓名 Terry Zhang 内容 you are great 提交

调试器 网络 样式编辑器 性能 内存 存储 无障碍环境

message.js

```
43 var query = new AV.Query('Message3');
44 query.find()
45 .then(
46   function (messages) {
47     let array = messages.map((item) => item.attributes)
48     array.forEach((item) => {
49       let li = document.createElement('li')
50       li.innerText = `${item.name}: ${item.content}`
51       let messageList = document.querySelector('#messageList')
52       messageList.appendChild(li)
53     })
54   }
55 )
56
57
58 let myForm = document.querySelector('#postMessage')
59
60 myForm.addEventListener('submit', function(e) {
61   console.log(e)
62   e.preventDefault()
63   let name = myForm.querySelector('input[name=name]').value
64   let content = myForm.querySelector('input[name=content]').value
65
66   var Message = AV.Object.extend('Message3');
67   var message = new Message();
68
69   message.save({
70     'name': name,
71     'content': content
72   }).then(function(object) {
73     let li = document.createElement('li')
74     li.innerText = `${object.attributes.name}: ${object.attributes.content}`
75     messageList.appendChild(li)
76     myForm.querySelector('input[name=content]').value = '' //清空表单里面的内容
77     console.log(object)
78   })
79 })
```

留言

richard: hello,world

richard: hello,world

Ron Chan: hehe

Terry Zhang: you are great

姓名 Terry Zhang 内容 you are great 提交

调试器 网络 样式编辑器 性能 内存 存储 无障碍环境

message.js

```
62 e.preventDefault()
63 let name = myForm.querySelector('input[name=name]').value
64 let content = myForm.querySelector('input[name=content]').value
65
66 var Message = AV.Object.extend('Message3');
67 var message = new Message();
68
69 message.save({
70   'name': name,
71   'content': content
72 }).then(function(object) {
73   let li = document.createElement('li')
74   li.innerText = `${object.attributes.name}: ${object.attributes.content}`
75   let messageList = document.querySelector('#messageList')
76   messageList.appendChild(li)
77   myForm.querySelector('input[name=content]').value = '' //清空表单里面的内容
78   console.log(object)
79 })
```

留言就出现了

这一行之后

显示留言的代码:



# 留言

richard: hello,world

姓名 frank

内容

提交

调试器

网络

样式编辑器

性能

内存

存储

无障碍环境

大纲

message.js x

```
36 //console.log("运行到这里没有报错!! ")
37
38
39
40
41
42 //从数据库批量读取数据,并显示出来
43 let myForm = document.querySelector('#postMessage')//找到选择器postMessage,其实就是表单
44 var query = new AV.Query('Message3');
45 query.find().then(function (messages) {
46     let array = messages.map(item=> item.attributes)
47     array.forEach(item=>{
48         let li = document.createElement('li')
49         li.innerText = `${item.name}: ${item.content}`
50         let messageList = document.querySelector('#messageList')
51         messageList.appendChild(li)
52     })
53 })
54
55 }
```

# 留言

richard: hello,world

richard: hello,world

样式编辑器

性能

内存

存储

无障碍环境

message.js x

```
6 //console.log("运行到这里没有报错!! ")
7
8
9
0
1
2 //从数据库批量读取数据,并显示出来
3 let myForm = document.querySelector('#postMessage')//找到选择器postMessage,其实就是表单
4 var query = new AV.Query('Message3');
5 query.find().then(function (messages) {
6     let array = messages.map(item=> item.attributes)
7     array.forEach(item=>{
8         let li = document.createElement('li')
9         li.innerText = `${item.name}: ${item.content}`
10        let messageList = document.querySelector('#messageList')
11        messageList.appendChild(li)
12    })
13 })
14
15 }
```

完整代码:

```
1 /**
2  * Created by Administrator on 2018/1/31 0031.
3  */
4
5 //初始化,这里是原来的
6 // var APP_ID = 'FqPJXxCv1gabU5JMeHrved6l-gzGzoHsz';
7 // var APP_KEY = 'Eevd4W01afMxVUg9s2pDDTr4';
8
9 // AV.init({
10 // appId: APP_ID,
```

```
11 // appKey: APP_KEY
12 // });
13 //下面的初始化是我自己的=在Learncloud上面添加的
14 AV.init({
15   appId: "yqzSiRS4ad0t1lIE2jvEBxsC-gzGzoHsz",
16   appKey: "sLwxvuc5C2s9XFSzeo18MRff",
17   // serverURLs: "api-uccloud.leancloud.cn",//最新版的里面这一行不用加，不然有错误！！
18 });
19 // // console.log("运行到这里没有报错！！")
20
21 // //创建TestObject表
22 // var TestObject = AV.Object.extend('TestTable');//TestTable 是表的名字
23
24 // //在表中创建一行数据，创建表TestObject的一个对象testObject,后面可以增删改查
25 // var testObject = new TestObject();
26
27 // //数据内容是words: 'Hello World!'
28 // //如果保存成功，则运行alert('')
29 // testObject.save({
30 //   words: 'Hello World!'
31 // }).then(function(object) { //这里
32 //   alert('LeanCloud Rocks!');
33 //   alert('添加数据成功!');
34 //
35 // })
36 //console.log("运行到这里没有报错！！")
37
38
39
40
41
42 //从数据库批量读取数据,并显示出来;就是把留言显示出来
43 let myForm = document.querySelector('#postMessage')//找到选择器postMessage,其实就是表单
44 var query = new AV.Query('Message3');
45 query.find()//批量找到所有留言记录
46 .then(
47   function (messages) {
48     let array = messages.map((item)=> item.attributes)
49     array.forEach((item)=>{//这里用forEach遍历
50       let li = document.createElement('li')
51       li.innerHTML = `${item.name}: ${item.content}`
52       let messageList = document.querySelector('#messageList')
53       messageList.appendChild(li)
54     })
55   }
56 )
```

```

57
58
59
60
61
62
63 myForm.addEventListener('submit',function(e){
64 console.log(e)
65 e.preventDefault();//阻止默认的提交事件,即阻止网页刷新
66 let name = myForm.querySelector('input[name=name]').value//获取留言的者的name(姓名)
67 let content = myForm.querySelector('input[name=content]').value//获取留言者写的内容
68 //创建一个名字为Message3的表
69 var Message = AV.Object.extend('Message3');
70 // //在表中创建一行数据, 创建表Message的一个对象message,后面可以增删改查,其实这里的message就是产生
71 var message = new Message();
72
73 message.save({
74 'name': name,
75 'content':content
76 }).then(function(object){
77 let li = document.createElement('li')//新建Li元素
78 li.innerHTML = `${object.attributes.name}: ${object.attributes.content}`//li里面添加留言
79 let messageList = document.querySelector('#messageList')//找到选择器messageList
80 messageList.appendChild(li)//添加li这个元素
81 myForm.querySelector('input[name=content]').value = ''//清空表单里面的内容
82 console.log(object)
83 })
84 })
85
86
87
88
89
90
91
92

```

注意:

要监听form表单的submit事件,不要监听提交按钮的点击事件.因为如果用户输入了信息,点击回车,那么用户的意愿还是提交,但是他并没有点击按钮,所以不会触发提交事件.

这个form表单的submit事件包含submit按钮被点击和在任意input打回车.用户提交有可能打回车,有可能点按钮

阻止提交的默认事件,因为点了提交按钮就会刷新当前页

注意map()和forEach()的使用

注意promise中

例如

```
1 query.find().then(fn1,fn2)
2   .then(fn3.fn4)
3
```

含义是:

如果成功执行回调函数fn1,如果失败执行回调函数fn2.

如果fn1和fn2都没有报错,那么执行fn3,如果fn1 和fn2中有一个报错,那么执行fn4