

LAB2 文档

11302016001 郑禧灿

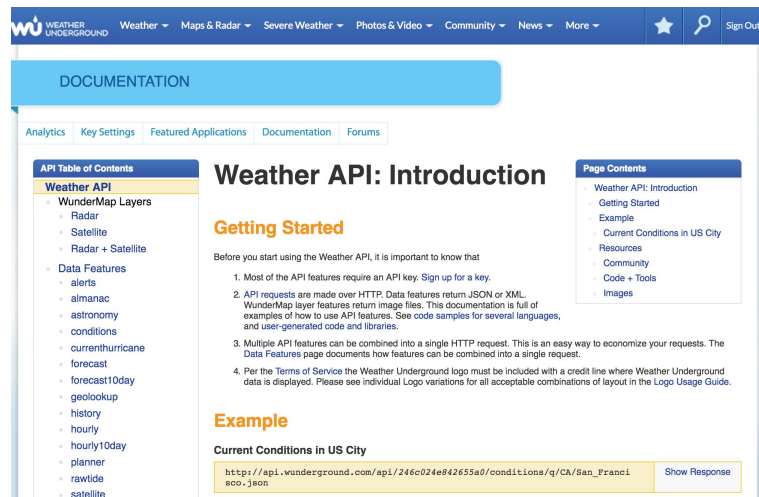
所选题目：WEB API 的开发实践

我选的题目是 WEB API 的开发实践，调用 API 来实现网页的开发。做了一个天气查询功能的一个网页。

首先我调用了天气查询功能的 API 和谷歌 API（微信平台开发等的 API 因为要注册审核等的原因用不了。因为我是外国人，外国人的护照等的信息不能通过审核，所以放弃了一些中国的开发平台 API，采用了国外开放的 API 平台）



下面就是我调用的 weather API。



1、首先大概的制作了网页的页面架构。

```
table#main{
  background-color: white;
  width: 600px;
  border: 1px solid #CCC;
  margin: 150px auto 0px auto;
  height: 250px;
}
input#searchInput, #stateInput{
  padding: 10px;
  margin: auto;
  width: 80%;
}
td#searchCol{
  text-align: center;
  border-right: 1px solid #CCC;
}
td#resultCol{
  padding-left: 30px;
}
label{
  font-weight: bold;
}
```

2. 重点在调用 API，调用了天气查询的 API。

```
// search handler
$("#search").click(function() {
  // get search field
  var city = $("#citiesSelect").val();
  var state = states[cities.indexOf(city)];
  var apiUrl = "http://api.wunderground.com/api/246c024e842655a0/conditions/q/" + state + "/" + city + ".json";

  // use weather API
  $.ajax({
    dataType: 'json',
    url: apiUrl
  }).then(function(data) {
    console.log(data);
  });
});
```

3. 实现了查询天气功能之后另外增加了查询的当地天气对应的城市地图，这时候又调用了谷歌的地图 API。

Like our APIs? Check out our infrastructure. Sign up to get \$300 in credit and 60 days to explore Google Cloud Platform. [Learn more](#)

Google APIs

API Manager

Overview

Credentials

Create credentials Delete

Create credentials to access your enabled APIs. [Refer to the API documentation](#) for details.

API keys

Name	Creation date	Type	Key
Server key 1	Jun 10, 2016	Server	AlzaSyBjgclCpFpOta8aU7NpY8PhPmktLptPSdg
Browser key 1	Jun 10, 2016	Browser	AlzaSyDsdpaFhpCGwZa1G3iRa59AYQ50qlcFGPQ

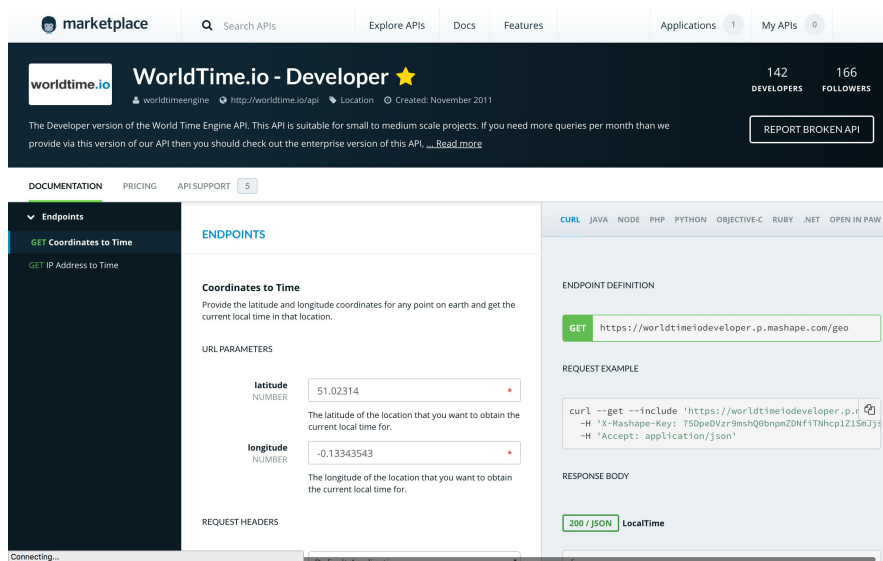
申请 API KEY 有点麻烦，调用一个 API 要申请一次 API KEY，而且申请的速度也比较慢。

```
// get timezone
var timezoneApiUrl = "https://maps.googleapis.com/maps/api/timezone/json?location=" +

$.ajax({
  url: timezoneApiUrl
}).then(function(data) {
  $("#current_timezone span").html(data.timeZoneName);
});
```

这个是调用了谷歌的 timezone API. 能显示对应城市的 timezone。

最后调用了 timeworld 的 API 来实现查询当地城市的时间和我们的时间。



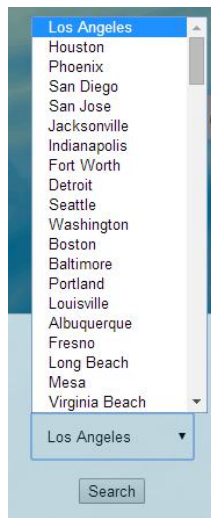
```
// convert to selected city's time
var cityTimeApiUrl = "https://worldtimeiodeveloper.p.mashape.com/geo?latitude=51.02314&longitude=-0.13343543";

$.ajax({
  url: cityTimeApiUrl,
  type: 'GET',
  beforeSend: function(xhr) {
    xhr.setRequestHeader("X-Mashape-Authorization", "75DpeDVzr9mshQ0bnpmZDNfiTNhpc1Z1SmJjsnUYhb5Ngif2Ea");
    xhr.setRequestHeader("X-Mashape-Key", "75DpeDVzr9mshQ0bnpmZDNfiTNhpc1Z1SmJjsnUYhb5Ngif2Ea");
    xhr.setRequestHeader("Accept", "application/json");
  }
}).then(function(data) {
  console.log(data);
  // set time offset
  $("#utc_offset span").html(data.current.utcOffset);
  $("#city_time span").html(data.summary.local);

  // var date_ms = new Date();
  var current_date = new Date();
  current_date.setHours(current_date.getHours() + 8);
  var month = current_date.getMonth() + 1;
  var date = current_date.getUTCDate();
  var hours = current_date.getUTCHours();
  var minutes = current_date.getUTCMinutes();
  var seconds = current_date.getUTCSeconds();

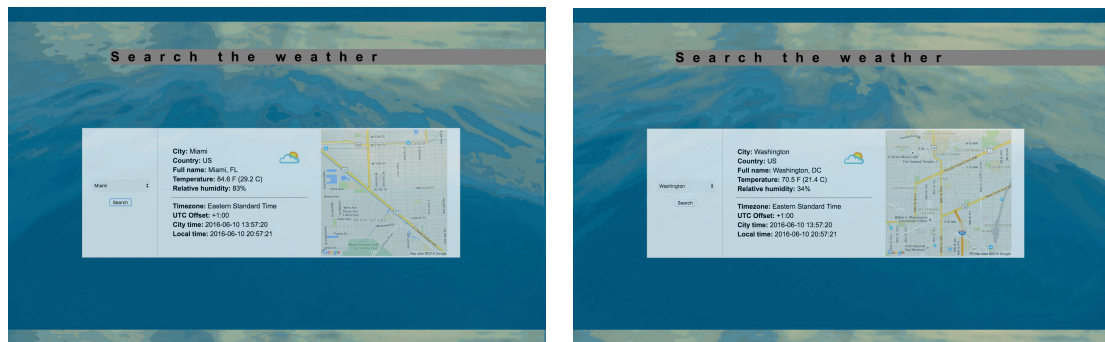
  var local_time = current_date.getFullYear() + "-"
    + ((month < 10) ? "0" + month : month) + "-"
    + ((date < 10) ? "0" + date : date)
    + " "
    + ((hours < 10) ? "0" + hours : hours) + ":"
    + ((minutes < 10) ? "0" + minutes : minutes) + ":"
    + ((seconds < 10) ? "0" + seconds : seconds);
  $("#local_time span").html(local_time);
});
```

选择城市的 `select`, 本来是要把美国 300 多个城市全部输入进去的, 但是这个方法比较笨, 所以用 `JAVA` 程序把网页里面的所有美国城市抽了出来拿来调用。



这样美国所有的 300 多个城市都可以准确的查询了。

最后的结果实现是这样子的



虽然外观设计上我个人比较缺少审美意识，所以不能弄的更漂亮一些，但主要 API 的调用非常顺利的完整的实现出了功能。

要注意的是需要翻墙 VPN 才可以实现这个功能。因为用到了国外的 API，翻墙才可以使用。

心得：

通过这个练习，觉得这种 API 的包装服务的架构非常方便，前端可以完全不知道后端的逻辑去使用，调用的方法也非常的明确和方便，这个良好的架构让我亲身的体验到前后端可以达到非常好的分离度，也让我感觉到这些架构设计上的重要性。将来我写一个 API 的话肯定非常的有帮助。