

公开课报名

1. 创建一个表 verify_code
2. 开发获取短信验证码接口 get /api/code
 - a. 时间格式化 moment

```
const moment = require('moment')
const now = moment();
const timestamp = now.format('YYYYMMDDHHmmss');
```

- b. 加密 md5

```
const md5 = require('md5')
md5(accountSid + authToken + timestamp)
```

- c. 网络请求 axios

```
const axios = require('axios')
//参数1: url, 参数2: 参数, 参数3: config
//返回Promise: 可用于await, 也可以then()中处理结果
axios.post(url,
  qs.stringify({to, accountSid, templateid, param, timestamp, sig}),
  {headers: {'Content-Type': 'application/x-www-form-urlencoded'}}
);
```

第三方接口调用要严格按照对方文档要求来设置, 如果成功, 对结果的处理是大同小异的。

Angular 7的使用 (学习中心开发)

1. angular/vue/react
 - angular: 完整解决方案、超前架构体系、跨平台、模块化、学习曲线陡峭
 - vue: 轻量级、渐进式、易用性强、文档丰富
 - react: 单项数据流, 生态系统强大、跨平台
2. 创建angular项目: <https://angular.cn>
 - 环境准备工作: angular cli: npm i -g @angular/cli
 - 创建项目: ng new kaikeba-student 或 webstorm新建
 - 运行项目: ng serve 或 npm start
3. TypeScript: 强类型的JS, ES6超集
 - 变量、常量 let a; const b=1;
 - 类型注解 const isBoolean:boolean;
 - 元组

```
let x: [string, number];
x = ['hello', 10]; // ok
```

-任意类型any

```
// 任意类型any
let notSure: any;
notSure = 4;
notSure = 'aaaa';

// any也能用于数组
const list: any[] = [1, true, 'aaa'];
list[1] = 100;
```

-枚举enum

```
enum Color {Red = 1, Green = 2, Blue = 3}

const c: Color = Color.Blue;
```

```
// 函数中使用类型约束
function greeting(person: string): string {
    return 'Hello, ' + person;
}
```

-void类型

```
// void类型
function warnUser(): void {
    alert('aaaaaaaa');
}
```

-接口：约束类型的结构

```
interface Person {
    firstName: string;
    lastName: string;
}

function greeting2(person: Person) {
    return 'hello, ' + person.firstName + ' ' + person.lastName;
}

const myname = greeting2({firstName: 'tom', lastName: 'cruise'});
console.log(myname);
```

