# 项目开展

## 地址 .

完整domo 地址:

https://github.com/itcastWsy/MyLeTao.git

上课demo 地址

<https://github.com/itcastWsy/letao14.git>

## 安装phpStudy

如果已经安装了mysql 可以忽略本步骤

双击打开即可

## 安装数据库工具 nativecat

双击打开即可(如已经安装则可以忽略)

## 安装nodejs

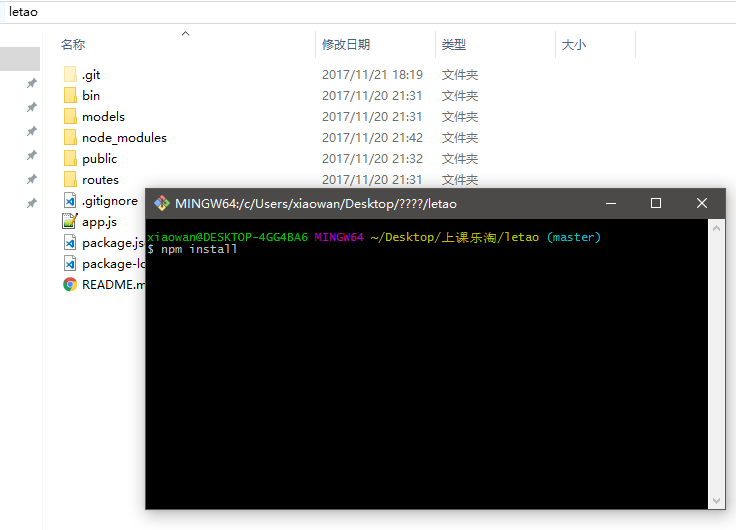
右键 nodejs.exe 以管理员身份安装 不要更改其他选项,一直点击下一步即可.

安装完成后, 打开命令行窗口 输入 node –v 如出现版本号 v8.9.1 即成功安装

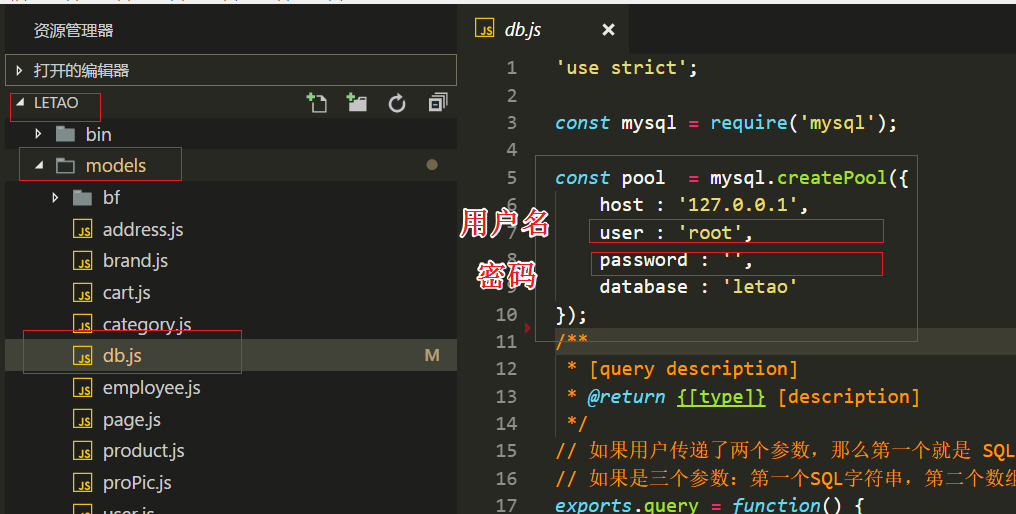
## 安装项目

进入项目根目录 打开命令行工具 输入

npm install

如 

## 连接数据库代码



此时需要修改为 你的电脑上的数据库的用户名和密码

## 项目启动的端口号代码



根据自己需求修改即可.

建议将完整demo的端口设置为3000

上课demo的端口为3001

自己代码的端口为3002

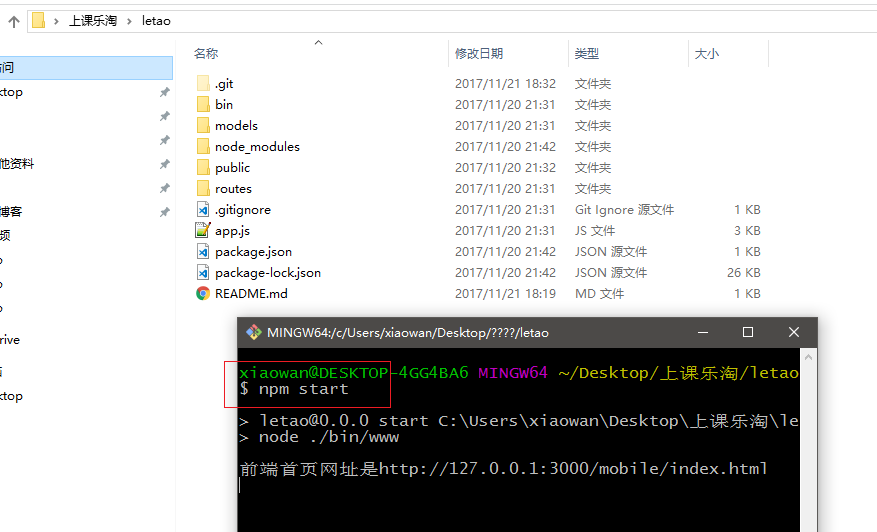
因为一个网站同时只能占用一个端口,所以需要设定其他的端口!

## 启动项目

在根目录 命令行 输入 npm start

如 在浏览器中输入 网址:

<http://127.0.0.1>:端口号/mobile/index.html



# 技术在线

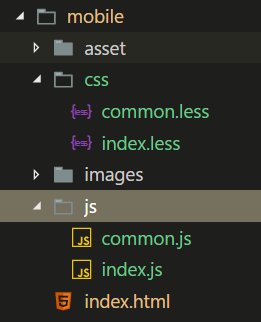
Mui : <http://dev.dcloud.net.cn/mui/>

Mui-demo: <http://www.dcloud.io/hellomui/>

Font-awe some : http://fontawesome.dashgame.com

# 前端web展示

项目结构



## 首页展示 index.html



### Mui 区域滚动

#### Html

<div class="mui-scroll-wrapper">

  <div class="mui-scroll">

    <!--这里放置真实显示的DOM内容-->

  </div>

</div>

#### Js

/\* 初始化区域滚动 \*/

mui('.mui-scroll-wrapper').scroll({

deceleration: 0.0005 //flick 减速系数，系数越大，滚动速度越慢，滚动距离越小，默认值0.0006

});

### Mui 轮播图

#### Html

<div class="mui-slider">

<div class="mui-slider-group mui-slider-loop">

<!--支持循环，需要重复图片节点-->

<div class="mui-slider-item mui-slider-item-duplicate"><a href="#"><img src="4.jpg" /></a></div>

<div class="mui-slider-item"><a href="#"><img src="1.jpg" /></a></div>

<div class="mui-slider-item"><a href="#"><img src="2.jpg" /></a></div>

<div class="mui-slider-item"><a href="#"><img src="3.jpg" /></a></div>

<div class="mui-slider-item"><a href="#"><img src="4.jpg" /></a></div>

<!--支持循环，需要重复图片节点-->

<div class="mui-slider-item mui-slider-item-duplicate"><a href="#"><img src="1.jpg" /></a></div>

</div>

<!—索引器-->

<div class="mui-slider-indicator">

<div class="mui-indicator mui-active"></div>

<div class="mui-indicator"></div>

<div class="mui-indicator"></div>

<div class="mui-indicator"></div>

</div>

</div>

#### Js

/\* 轮播图 \*/

mui('.mui-slider').slider({

interval: 1000//自动轮播周期，若为0则不自动播放，默认为0；

});

### 辅助类

mui-clearfix

mui-pull-left

mui-pull-right

mui-text-left

## 搜索中心 search.html



功能:

1 点击搜索按钮-跳转页面

2 存贮搜索历史

3 对搜索历史做修改 本地存贮的技术 localstorage

4 点击返回->返回上一页面

## 关键代码段:

// 有数据就获取数据 无 就获取空数组 !!

*var* arr = (ls.getItem("LT\_his") && JSON.parse(ls.getItem("LT\_his"))) || [];

// 判断 有没有数据

if(arr.length<1){

$(".history\_list").html('');

return;

}

// 去掉空格

if(!$.trim(val)){

return false;

}

// 要做去重的处理

for (*var* i = 0; i < arr.length; i++) {

if(arr[i]==val){

// 删除旧的 添加新的到最开头

// (要删除的值的索引,要删除几个)

arr.splice(i,1);

}

}

// unshift 往数组的头部加数据

// push 数组的尾部加数据!!

arr.unshift(val);

// 事件的委托

$("body").on("tap",".item\_close",*function* () {

// console.log(1234);

// 获取父元素的索引

*var* index=$(this).parent().index();

// console.log(index);

*var* ls = localStorage;

*var* arr = (ls.getItem("LT\_his") && JSON.parse(ls.getItem("LT\_his"))) || [];

// 删除数组中的元素

arr.splice(index,1);

// 存值

ls.setItem("LT\_his",JSON.stringify(arr));

// 重新渲染

// (注意 自己删除自己 又自己绑定 容易出事 )

loadHistory();

})