

1.需求

- a) 输入框 展示查询结果

2.前期准备

- a) 寻找第三方 API
- b) https://tcc.taobao.com/cc/json/mobile_tel_segment.htm?tel=18565695864

3.功能

- a) 实现实时查询号码信息
- b) 建立号码信息库
 - i. 减少对数据提供方的依赖
 - ii. 减少 HTTP 请求提高响应速度

4.开发环境及使用工具

5.项目框架结构搭建

- a) 框架结构
 - i. MVC
 - ii. Index.html 访问入口文件
 - iii. api.php 处理入口请求，返回响应数据，前期数据调试
 - iv. app/ 具体业务处理模块
 - v. libs/ 扩展库（与业务层没有直接关系、公共调用）
 - vi. static/ 静态文件（存放 css , js 等静态文件）
 - vii. model （提供数据源）
- b) 使用 PHP 内置 web 服务器
 - i. `php --help` //查看帮助
 - ii. `php -S <addr>:<port>` Run with built-in web server.

```
[root@localhost ~]# php -S 0.0.0.0:90
PHP 5.5.31 Development Server started at Fri Jul 8 01:19:15 2016
Listening on http://0.0.0.0:90
Document root is /root
```

注意：项目根目录（您在所属的目录下执行命令就把当前目录作为根目录）

6.自动加载类

a) 传统方式使用 require_once 包含文件

b) autoload.php 自动加载

```
<?php
class autoload
{
    //定义加载类文件规则
    public static function load($className)
    {
        //var_dump($className);
        $fileName = str_replace('\\','/', $className);
        $classFile = sprintf('%s.php', $fileName);
        if(is_file($classFile)){
            require_once $classFile;
        }
    }
}

spl_autoload_register(['autoload','load']);//注册到内置自动加载
```

c) 包含 autoload.php 文件使用类

```
<?php

//require_once "app/QueryPhone.php";//传统加载类模块方式
require_once "autoload.php";

//当包含的文件中使用命名空间时不能包含此文件

app\QueryPhone::query('18565695464');

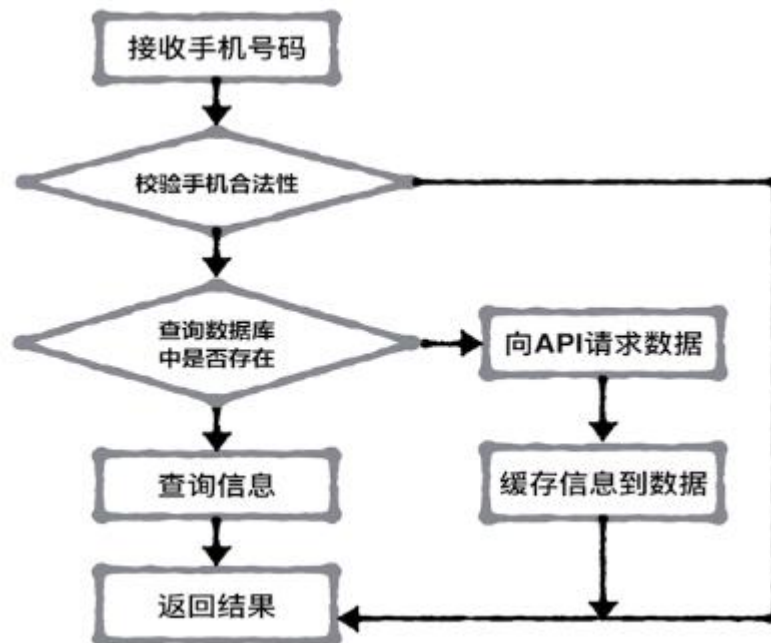
/* 当使用 app\QueryPhone 类中方法，必须先加载类则在使用类时首先会执行

spl_autoload_register(['autoload','load']);即先实例化 autoload 并

执行 load 方法，此方法负责定义规则包含类文件，然后才能执行类中的方法。

*/
```

7.信息查询流程解析



8.校验手机号码合法性

9.API 请求数据

+ 在 libs 定义请求数据的模块

```
<?php
namespace libs;
class HttpRuquest
{
    public static function request($url,$params,$method="GET")
    {
        $response = false;
        if($url){
            $method = strtoupper($method);
            if($method == 'GET'){
                if(is_array($params) and count($params)){
                    if(strrpos($url,'?')){
                        $url = $url .'&'.http_build_query($params);
                    }else{
                        $url = $url .'?.http_build_query($params);
                    }
                }
            }
            $response = file_get_contents($url);
        }
    }
}
```

```
        return $response;
    }
}
```

10.格式化数据

```
/**
 * 格式化数据
 * @author muller
 * @param INT $data
 */
public static function formatData($data = null)
{
    $ret=false;
    if($data){
        preg_match_all("#(\w+):'([^']+)#", $data, $res);
        $ret = array_combine($res[1], $res[2]);
    }
    return $ret;
}
```

11.缓存数据到数据库

12.渲染数据