**1.概 述**

* 名词解释：内部API，内部（可以理解为内部IP）才能调用的API；因此，我们在调用内部API的时候，需要进行权限校验，只有允许的IP才能调用
* 作用：在原有的架构中，我们会从dao层取数据；但是在公司新的架构中，所有数据都需要用内部API从service中取得数据。

**2.**内部API的制作

**2.1设计原则**

* 外部数据-**不信任性**：内部API对于所有外部传入的数据都应该采用不信任性原则，需要做完整的服务端数据格式校验和自身相关的业务逻辑校验，才能进行对应的操作
* 查询数据-**原子性**：这是我们查询数据的基本原则，不能因为各种很妖的要求而改变（内部API提供的数据一般需要使用者进行处理之后，才能在前端展示，而不是作为前端展示的直接提供者）

|  |
| --- |
| //原子性的返回结果示例  {      "bSuccess": true,      "aData": {          "iAutoID": 20,          "sName": "张三",          "iAge": 18,          "iSex": 1      }  }    //非原子性的返回结果示例  {      "bSuccess": true,      "aData": {          "iAutoID": 20,          "sName": "张三",          "iAge": 18,          "sSex": "男",          "aFriends": {              "iAutoID": 21,              "sName": "李四"          }      }  } |

* 增删改操作-**事务性**：接口对数据进行操作时，需要完成一个完整的操作逻辑，而不是只完成一部分

**2.2接口Controller文件**

* 接口文件路径(以新房为例)：/xf-web/app/Http/Controllers/V1\_0
* 文件命名：xxxx.php（首字母大写驼峰式命名）
* 类命名：和文件名一致
* 父类：继承Controller

**2.3接口方法**

 接口方法名称，仿照laravel中的restful方法命名，详细如下：

* 新增：命名-function store()
* 删除：命名-function destroy()
* 修改：命名-function update()
* 取得单条：命名-function show()
* 取得列表：命名-function index()

    注意：以上只是列举了最常见的方法命名，如果需要用到除此之外的方法，请按照代码基础规范来命名方法（即首字母小写驼峰式）

|  |
| --- |
| namespace App\Http\Controllers;    class User extends Controller  {      public function store()      {      }        public function destroy()      {      }        public function update()      {      }        public function show()      {      }        public function index()      {      }    } |

**2.4返回结果（目前只以JSON格式返回为示例，以后会增加其他返回格式）**

* 返回结果：返回结果格式是一串**JSON**格式
* 字段解析-bSuccess:接口调用是否成功；true-成功，false-失败
* 成功返回结果（以下列表为各情景返回结果，如需增加，请联系我或者自行添加，以便完善）：

|  |
| --- |
| //新增成功  {      "bSuccess": true,      "aData": {          "iAutoID":123      }  }  //删除成功  {      "bSuccess": true,      "aData": {          "iAutoID":123      }  }    //修改成功  {      "bSuccess": true,      "aData": {          "iAutoID":123      }  }    //取得总数返回  {      "bSuccess": true,      "aData": {          "iTotal": 10      }  }    //取得详情返回  {      "bSuccess": true,      "aData": {          "iAutoID": 20,          "sName": "张三",          "iAge": 18,          "iSex": 1      }  }    //取得列表返回  {      "bSuccess": true,      "aData": {          "iTotal": 100,      //总数          "iPerPage":1,       //每页数量          "iCurrentPage":1,   //当前页码          "iLastPage":50,     //最后一页          "iFrom":1,          //本次取值开始位置          "iTo":2,            //本次取值结束位置          "aList": [              {                  "iAutoID": 20,                  "sName": "张三",                  "iAge": 18,                  "iSex": 1              },              {                  "iAutoID": 21,                  "sName": "李四",                  "iAge": 19,                  "iSex": 0              }          ]      }  } |

* 错误返回：

|  |
| --- |
| //操作失败返回  // 验证错误  {      "bSuccess": false,      "sErrorType":"validation",  //错误类型，validation-验证错误；logic-手动抛出错误；system-系统异常错误      "aErrors": {          "sName": { // 错误字段              "iCode": 123, // 错误码              "sMsg": "BALABALA\_ERROR", // 错误信息              "sRule": "required|max:5", // 导致错误的规则              "mValue": "MyName", // 字段对应的值          },          "iAge": {              "iCode": 233,              "sMsg": "BALABALA\_ERROR",              "sRule": "required|max:999|min:1",              "mValue": 999999,          }      }  }    // 手动抛出  {      "bSuccess": false,      "sErrorType":"logic",      "aErrors": [          {              "iCode": 1,              "sMsg": "Error Processing Request",              "sRule": "",              "mValue": ""          }      ]  }    // 系统异常  {      "bSuccess": false,      "sErrorType":"system",      "aErrors": {          {              "iCode": 1,              "sMsg": "RuntimeException", // 系统异常sMsg为异常类型, 不传出异常信息以免外部不做处理直接抛出              "sRule": "",              "mValue": ""          }      }  }      // 表单验证自动抛出异常并进行捕获  // 其他错误手动抛出异常由顶部自动捕获  // 手动抛出异常时错误字段可以使用导致错误的变量名  // mValue字段使用该变量的值 |

* sErrorType目前有3种类型:
* validation: 验证错误(通常由表单验证自动抛出)
* logic: 逻辑错误(通常手动抛出, 例如除数为0)
* system: 系统错误(其他没有捕获的由系统自动抛出的异常, 该类异常应当极力避免, 出现此类异常通常来说就是代码存在BUG)

**2.5SDK文件的提供**

* SDK清单文件：会根据此文件统一生成SDK文件，目录结构：app\Services\SDK\V1\_0\VerifyService.php，文件示例：

|  |
| --- |
| <?php  namespace App\Services\SDK\V1\_0;  class VerifyService  {      /\*\*       \* 测试用例       \*/      public function getDetail($id)      {          return ['GET', '/detail/{$id}', ['id']];      }  } |

* SDK文件自动生成：php artisan make::sdk VerifyService 1.0
* SDK文件：根据清单文件自动生产，目录结构：app\Services\SDK\V1\_0\VerifyServiceSDK.php，文件示例：

|  |
| --- |
| <?php  /\*\*   \* 由SDKCreator自动生成   \*   \* @date 2015-04-30   \*/  class VerifyServiceSDK  {      //一大波自动生产的公用方法        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* VerifyServiceSDK v1.0 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/        /\*\*       \* 测试文件       \*/      public function getDetail($id)      {          return $this->request('GET', "/detail/{$id}", compact('id'));      }  } |

**2.6日志记录**

* 日志保存路径：/storage/logs/laravel-2015-04-15.log（暂时存放在当前项目文件下，之后会接入统一的日志服务）
* 日志记录点：共两次，即访问时一次（request），返回时一次（response）
* 日志保存结构：

|  |
| --- |
| //request日志记录格式：IP+追踪ID+传参方式+URL+URL路径+request详情  [2015-05-14 09:16:10] dev.INFO: 10.0.2.2 C59EBC02-8839-011E-BDC1-794A15908742 GET  http://localhost/1.0/check/123 1.0/check/123 request {"method":"GET","url":"[http://localhost/1.0/check/123",](http://localhost/1.0/check/123)  "path":"1.0/check/123","query":[],"ajax":false,"pjax":false,"secure":false,"ip":"10.0.2.2","ips":["10.0.2.2"],  "userAgent":"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135  Safari/537.36","request":[]}      //response日志记录格式：IP+追踪ID+传参方式+URL+URL路径+response详情  [2015-05-14 09:16:11] dev.INFO: 10.0.2.2 C59EBC02-8839-011E-BDC1-794A15908742 GET  http://localhost/1.0/check/123 1.0/check/123 response {"status":200,"response":  "{\"bSuccess\":false,\"aErrors\":{\"sBcode\":{\"iCode\":1,\"sMsg\":\"s bcode \\u81f3\\u5c11\\u4e3a 4 \\u4e2a\\u5b57\\u7b26\\u3002\",\"sRule\":\"required|min:4|max:50\",\"mValue\":\"123\"},\"sVcode\":  {\"iCode\":1,\"sMsg\":\"s vcode \\u4e0d\\u80fd\\u4e3a\\u7a7a\\u3002\",\"sRule\":\"required|min:1|max:50\",\"mValue\":null},\"sClientID\":{\"iCode\":1,\"sMsg\":\"s clientid \\u4e0d\\u80fd\\u4e3a\\u7a7a\\u3002\",\"sRule\":\"required|min:4|max:50\",\"mValue\":null},\"sClientIP\":  {\"iCode\":1,\"sMsg\":\"s client i p \\u4e0d\\u80fd\\u4e3a\\u7a7a\\u3002\",\"sRule\":\"required|ip\",\"mValue\":null}}}"} |

**2.7接口URL的命名规则**

* 新增URL：/1.0/xq；传参方式-POST
* 列表URL：/1.0/xq；传参方式-GET
* 删除URL：/1.0/xq/{$pk} ；传参方式-DELETE
* 更新URL：/1.0/xq/{$pk} ；传参方式-PUT/PATCH
* 查询单条记录URL：/1.0/xq/{$pk} ；传参方式-GET
* replace接口URL：/1.0/xq/replace;传参方式-POST  
    
  注意：$pk规定必须为主键

3. 内部API的调用

**3.1内部API的调用**

* SDK文件,由接口提供者统一提供，文件结构：/xf-web/app/Http/Services/SDK/V1\_0/VerifyServiceSDK.php，  文件内容大致如下：

|  |
| --- |
| <?php  /\*\*   \* 由SDKCreator自动生成   \*   \* @date 2015-04-30   \*/  class VerifyServiceSDK  {      //一大波自动生产的公用方法        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* VerifyServiceSDK v1.0 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/        /\*\*       \* 测试文件       \*/      public function getDetail($id)      {          return $this->request('GET', "/detail/{$id}", compact('id'));      }  } |

* 调用内部接口

|  |
| --- |
| //通用的调用方法  $oVerifyManageSDK = new VerifyManageSDK;  $oVerifyManageSDK->setDomain('<http://verify.sinute.s.dev1.anhouse.com.cn/'>);  $oVerifyManageSDK->setFrom('sdk-test');  $oVerifyManageSDK->setKey('123456');  $oVerifyManageSDK->setTrackId(uniqid('', true));  $oVerifyManageSDK->getDetail(100);    //在新框架里会有统一的调用方法，暂时不提 |

**3.2统一传参**

get统一传参：

* \_format（暂时未启用）：返回结果格式，可选择json、xml，默认为json格式；
* \_from：当前调用内部API的域名，如：xf、esf等；
* \_token：用来校验时间有效性的加密串
* \_trackid:当做日志记录追踪标签
* \_requesttime:请求时间

**3.3日志记录**

* 日志保存路径：/storage/logs/laravel-2015-04-15.log（暂时存放在当前项目文件下，之后会接入统一的日志服务）
* 日志记录点：共两次，即访问时一次（request），返回时一次（response）
* 日志保存结构：

|  |
| --- |
| //request日志记录格式：IP+追踪ID+传参方式+URL+URL路径+request详情  [2015-05-14 09:16:10] dev.INFO: 10.0.2.2 C59EBC02-8839-011E-BDC1-794A15908742 GET  http://localhost/1.0/check/123 1.0/check/123 request {"method":"GET","url":"[http://localhost/1.0/check/123",](http://localhost/1.0/check/123)  "path":"1.0/check/123","query":[],"ajax":false,"pjax":false,"secure":false,"ip":"10.0.2.2","ips":["10.0.2.2"],  "userAgent":"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135  Safari/537.36","request":[]}    //response日志记录格式：IP+追踪ID+传参方式+URL+URL路径+response详情[2015-05-14 09:16:11] dev.INFO: 10.0.2.2 C59EBC02-8839-011E-BDC1-794A15908742 GET  http://localhost/1.0/check/123 1.0/check/123 response {"status":200,"response":  "{\"bSuccess\":false,\"aErrors\":{\"sBcode\":{\"iCode\":1,\"sMsg\":\"s bcode \\u81f3\\u5c11\\u4e3a 4 \\u4e2a\\u5b57\\u7b26\\u3002\",\"sRule\":\"required|min:4|max:50\",\"mValue\":\"123\"},\"sVcode\":  {\"iCode\":1,\"sMsg\":\"s vcode \\u4e0d\\u80fd\\u4e3a\\u7a7a\\u3002\",\"sRule\":\"required|min:1|max:50\",\"mValue\":null},\"sClientID\":{\"iCode\":1,\"sMsg\":\"s clientid \\u4e0d\\u80fd\\u4e3a\\u7a7a\\u3002\",\"sRule\":\"required|min:4|max:50\",\"mValue\":null},\"sClientIP\":  {\"iCode\":1,\"sMsg\":\"s client i p \\u4e0d\\u80fd\\u4e3a\\u7a7a\\u3002\",\"sRule\":\"required|ip\",\"mValue\":null}}}"} |