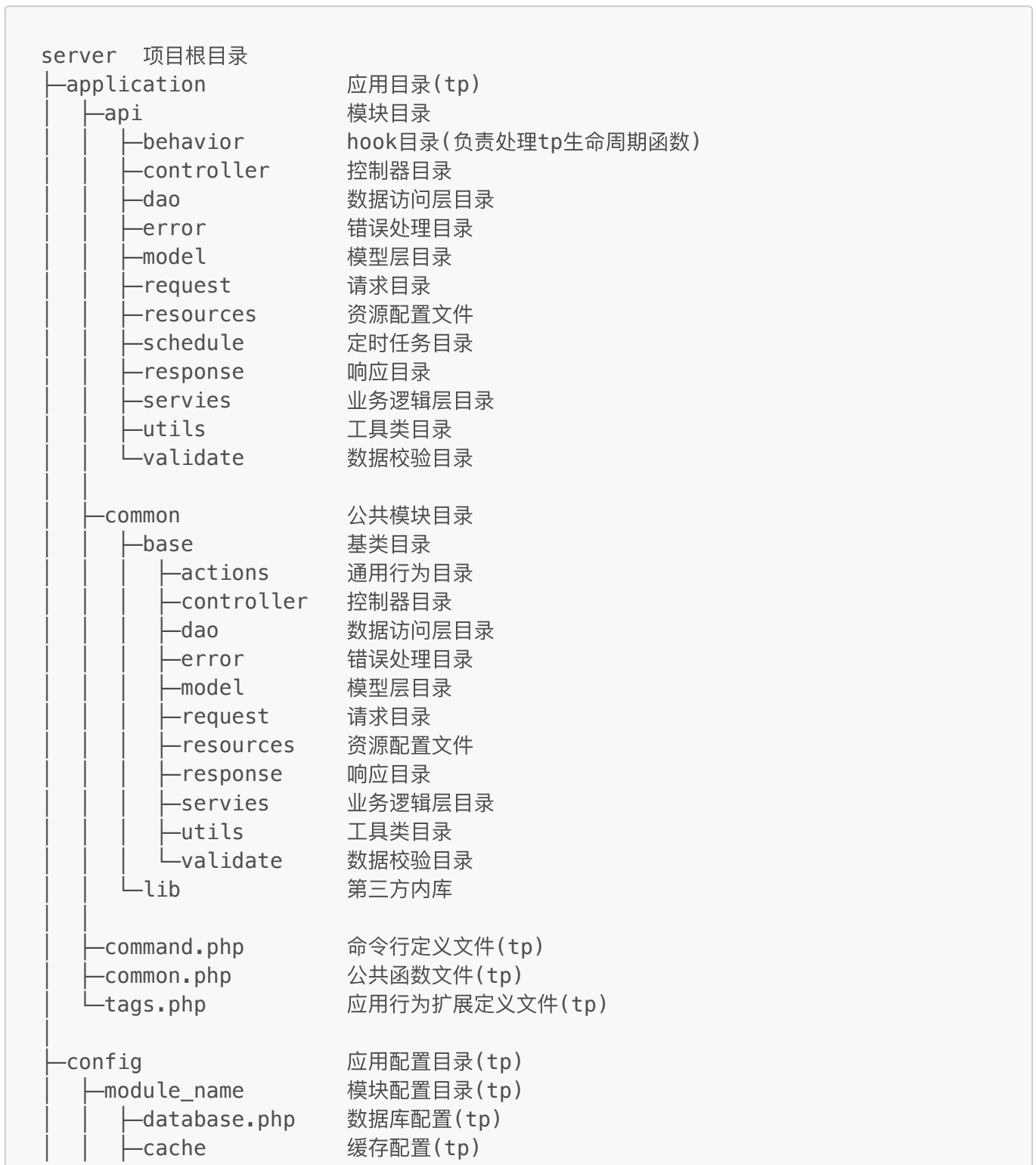


中世康恺RIS3.0文档说明

1. 架构说明

该项目使用PHP7.1,框架使用TPv5.1.37 LTS,数据使用的是Mysql5.6+,缓存使用Redis+文件缓存,服务器部署在centos7+的linux环境下,通过php-fpm启动使用nginx对外提供接口

- 1. ThinkPHP [github地址](#) [中文文档](#)
- 2. 单元测试使用的是TP的think-testing=2.0.*拓展,它是基于PHPUnit([中文文档](#))的
- 3. 项目目录层次图如下



└─...	
└─app.php	应用配置(tp)
└─cache.php	缓存配置(tp)
└─cookie.php	Cookie配置(tp)
└─database.php	数据库配置(tp)
└─log.php	日志配置(tp)
└─session.php	Session配置(tp)
└─template.php	模板引擎配置(tp)
└─trace.php	Trace配置(tp)
└─public	WEB目录 (对外访问目录) (tp)
└─index.php	入口文件(tp)
└─router.php	快速测试文件(tp)
└─.htaccess	用于apache的重写(tp)
└─route	路由定义目录(tp)
└─route.php	路由定义(tp)
└─...	更多(tp)
└─extend	扩展类库目录(tp)
└─runtime	应用的运行时目录 (可写, 可定制) (tp)
└─thinkphp	框架系统目录(tp)
└─lang	语言文件目录(tp)
└─library	框架类库目录(tp)
└─think	Think类库包目录(tp)
└─traits	系统Trait目录(tp)
└─tpl	系统模板目录(tp)
└─base.php	基础定义文件(tp)
└─convention.php	框架惯例配置文件(tp)
└─helper.php	助手函数文件(tp)
└─logo.png	框架LOGO文件(tp)
└─vendor	第三方类库目录 (Composer依赖库) (tp)
└─build.php	自动生成定义文件 (参考) (tp)
└─composer.json	composer 定义文件(tp)
└─LICENSE.txt	授权说明文件(tp)
└─README.md	README 文件
└─think	命令行入口文件(tp)

- 4. 项目调用流程图如下

2. 基类描述

- 1. 文件目录图谱

common	公共模块目录
└─base	基类目录
└─actions	通用行为目录
└─IBaseAction.php	通用行为接口



- 2. common

common目录下的base目录里面的各类负责定义每层的行为,和一些通用的实现,lib目录负责存放第三方库并封装方法由base下utils目录里面的工具类封装调用方法抛出给实现类调用

- 3. base

base目录下的各类是所有实现类的基类,base类分为三种:行为定义接口,继承类,自定义类

base类的继承类基本都重写了父类的构造方法,并调用父类构造,只是加上了父类构造调用前后的钩子(beforeConstruct和afterConstruct)以供子类有需要时重写

base类的自定义类构造也加上了初始化钩子(beforeInit和afterInit)以供子类有需要时重写

base类都是抽象类,尽管有些可能没有抽象方法

- 4. actions 通用行为定义

actions定义了实现类需要的各个行为,比如IBaseAction.php中的抛出异常,token校验失败异常,参数校验失败异常,中断操作返回前端成功响应等操作和IBaseLog.php的记录日志操作,该接口只定义行为,具体实现有实现类完成

- 5. controller 控制器层

controller下的BaseController.php为所有控制器实现类的祖父类

它定义了控制器层的一下通用行为并实现了一部分,剩下的由子类去实现或者重写已实现的行为

定义的行为有 设置请求包装对象,校验(获取)token,获取请求参数,成功返回,异常失败,异常成功等方法

- 6. dao 数据访问层

dao下的BaseDao.php为所有数据访问层实现类的祖父类

- 7. error

error\model 下的 BaseErrorModel.php是所有异常模型实现类的基类

BaseError.php为异常基类继承自PHP原生Exception

BaseHttpError.php为Http异常基类继承自tp的think\exception\HttpException

BaseHttpResponseError.php为Http响应异常基类继承自tp的think\exception\HttpResponseException

- 8. model

model下的BaseModel.php为所有数据模型层实现类的祖父类

BaseModel.php继承自tp的think\Model具体可用方法可以通过[TP关于模型层的文档查看](#)

- 9. request

request下的BaseRequest.php为所有请求对象实现类的祖父类

BaseRequest.php继承自tp的think\Request具体可用方法可以通过[TP关于请求的文档查看](#)

- 10. response

- BaseResponse.php

BaseResponse.php为所有响应对象的基类,它封装了响应的一系列方法

- BaseThinkphpResponse.php

BaseThinkphpResponse.php为tp响应基类继承自tp的think\Response具体可用方法可以通过[TP关于响应的文档查看](#)

- 11. servies

services下的BaseService.php为所有业务逻辑层实现类的祖父类

- 12. utils

utils目录下的一系列工具类封装了一些工具方法供其他类调用

BaseUUID.php封装了生成和解析uuid的方法

TimeUtils.php封装了一下时间函数方法

- 13. validate

validate下的BaseValidate.php为所有数据校验层实现类的祖父类

BaseRequest.php继承自tp的think\Validate具体可用方法可以通过[TP关于验证器文档](#)查看

- 14. lib

lib放置第三方库的源码或者调用方式,由base/utils下的类实现基本调用方法,并由最终的的工具实现类实现调用方法供其他类调用,建议可以用composer形式导入的类都使用composer形式引入,原则上不允许直接修改第三方类的源码,如果需要修改应当继承改类后重写

api_sdk是阿里云短信sdk库

Message.php 和 AliyunSms.php封装了发送短信的函数

2. 远程诊断实现类描述

- 1. 文件目录图谱

-



		自定义请求头校验
		参数错误
		抛出成功
		Token校验失败
		异常基类
		Http异常基类
		Http响应异常
model		模型层目录
		模型层基类
		模型层实现类目录
		模型层实现类(公有云数据库)
		模型层实现类(私有云数据库)
resources		资源目录
		资源配置
request		请求目录
		请求基类
response		响应目录
		响应基类
		自定义请求头校验响应
		默认响应/成功响应
		tp响应对象继承类
		异常响应
schedule		定时任务
servies		业务逻辑层目录
		业务逻辑层基类
		公共业务逻辑层实现类目录
		公共业务逻辑层实现类
		业务逻辑层实现类目录
		业务逻辑层实现类
utils		工具类目录
		发送短信工具类
		uuid工具类
validate		数据校验目录
		数据校验基类
		数据校验实现类目录
		数据校验工具类
		数据校验规则
version		
		日志目录
		1.0日志
		x版本日志
		sql目录
		1版本目录

```

|  |  | build.sql      数据库构建语句
|  |  | build.php     数据库构建语句
|  |  |
|  |  | └ vx         x版本目录
|  |  |   | build.sql  数据库构建语句
|  |  |   | build.php  数据库构建语句
|  |  |   | up.sql     数据库构建语句
|  |  |   | up.php     数据库更新语句
|  |  |
|  |  | └ version.php

```

- 2. api

api为中世康恺RIS3.0+远程诊断项目目录,对外抛出去的接口全部由该目录下的文件甩出

基类名称基本都是有Zskk开头的类

- 3. actions

actions 是common/base基类定义的一些通用行为实现类

actions 实现了在控制器层,业务逻辑层,数据访问层,数据模型层等抛出各种异常以及日志打印的方法

由于PHP语法机制 actions 中类并不是通过 implements 的方式继承接口 而是使用了 trait 关键字定义 其他层基类调用时要通过use该类实现行为接口,由于没有不是通过实现接口方式去做的,需要这边定性的trait类里面的方法和接口方法,方法名和参数一致

ZskkCache.php 为缓存实现类通过封装tp的Cache类提供file和两级redis的缓存

ZskkDefaultAction.php 抛出异常直接由ZskkDefaultResponse等类实现

ZskkDefaultLog日志打印使用tp自定义日志的方法,日志等级为 logName,子类需要重写

- 4. behavior

behavior 下存放监听tp生命周期函数的类

- 5. controller

controller为控制器层 所有控制器都存放在该目录下

ZskkDefaultController继承自common/base的控制器基类,是项目所有控制器实现类的父类

ZskkDefaultController 实现了基本的设置请求对象,设置参数,token校验,抛出各种异常,成功返回,打印控制器层日志等方法

xxx为具体控制器目录 理论上下面应该只有一个XxxController.php文件继承自ZskkDefaultController

XxxController.php为实际业务的控制器类,由于是框架直接调用,所以方法可以依赖注入,一般注入service层对象

XxxController.php原则上只允许调用service层对象

XxxController.php原则上只允许定义被框架调用的方法,其他方法根据其通用性等级,可以沉淀到对应service层或者公共service层或直接抽取到控制器层基类

- 6. dao

dao 为数据访问层

dao层被且只被service层调用,对service提供数据

目前dao层只调用model层

预计dao层还需要调用缓存层获取数据(可能是file缓存或者redis/codis)

dao层调用的model(或者以后的cache)对象 由构造时依赖注入, 具体实现类负责定义成员变量和注入逻辑

dao层通过use ZskkCache 的 getCache 和 setCache的方法获取和设置缓存

设置缓存默认先设置文件缓存再设置redis缓存

读取缓存默认先从文件缓存中读 没有再去redis缓存中读并存入文件缓存

缓存默认保存时间86400秒就是一天

通过重写redis_name属性改变缓存redis的配置

- 7. error

error 封装了对数据错误的处理

ZskkDefaultError继承自BaseError类(祖父类是Exception)用于抛出系统异常 基本应该用不上

ZskkHttpDefaultError继承自BaseHttpError类(祖父类是think\exception\HttpException)用于抛出http异常 基本应该用不上

ZskkHttpResponseDefaultError继承自BaseHttpResponseError类(祖父类是 think\exception\HttpResponseException)用于抛出http响应异常 基本应该用不上

error下model存放了错误处理的模型类

ZskkErrorModel继承自BaseErrorModel为所有错误处理模型基类

Token为token验证失败的错误返回模型类 后续错误码和错误提示可能需要封装到resources

Params为参数校验失败的错误返回模型类 后续错误码和错误提示可能需要封装到resources 该错误应当多数情况下由校验工具类抛出

Cros为请求自定义头验证类(OPTIONS Http探路)返回 由控制器基类调用

- 8. model

model 为数据模型层 复制请求数据库并返回结果

一般情况下由且只由dao层调用

ZskkDefaultModel 为模型层基类继承自BaseModel(祖父类为tp的Model)

xxx为数据模型层具体实现一般情况下一个数据库表对应一个xxx目录

根据业务特殊情况xxx目录下有PublicXxxModel.php和XxxModel.php XxxModel.php负责私有云(一般是代码运行环境)下的数据库表数据操作 PublicXxxModel.php负责公有云下的数据库表数据操作

多数据库配置在/server/config/database.php中配置 model使用中通过继承类重写connection字段修改数据库连接配置

目前基类的是connection字段为db_config_private代表私有云配置,XxxModel.php也是连接私有云不需要重写配置,PublicXxxModel.php需要连接公有云数据库,需要重写connection字段为null代表默认配置(后期可能增加db_config_xxx(xxx为lc jsd zskk zgpa等等))

后期可能会把配置名称统一抽取出来

- 9. request

request目录存放封装后的请求对象

ZskkDefaultRequest继承自BaseRequest(祖父类是tp的Request类)

- 10. resources

resources目录主要存放一些资源类比如各种状态的定义已经错误的定义,后期可能会把所有返回的提示放这方便以后国际化的处理

- 11. response

response目录存放封装后的响应对象

ZskkDefaultThinkphpResponse继承自BaseThinkphpResponse(祖父类是tp的Response类)

ZskkBaseResponse继承自BaseResponse是http响应的工具类

ZskkCrosResponse继承自ZskkBaseResponse是自定义头验证类(OPTIONS Http探路)的响应类

ZskkErrorResponse继承自ZskkBaseResponse是错误返回的响应类

ZskkDefaultResponse继承自ZskkBaseResponse是默认响应类被控制器层success方法调用抛出完成成功的响应结果

- 12. schedule

schedule目录想存放一些定时任务 php可能支持不好 后续通过linux服务器定时任务调用curl接口实现会比较好?

- 13. servies

业务逻辑层 负责处理业务的具体逻辑

ZskkDefaultService继承自BaseService是所有service类的基类

common文件夹存放公共业务逻辑层代码,该层文件命名规则应该是CommonXXXService

xxx为业务逻辑实现类目录 理论上说 应该是一个控制器层(controller)的xxx目录对应一个业务逻辑层(service)的xxx目录对应一个数据访问层(dao)的xxx目录可能对应多个数据模型层(model)目录或缓存层(fiel/redis)

原则上servies层只允许被controller层调用,servies也只能调用dao层,但是考虑到业务可能重复比较多,service层可以调用commonservice(公共业务逻辑)层

servies需要使用的model层或commonservice层实例也是在实现类重写构造方法类注入

- 14. utils

utils主要为了提供一些公共工具的调用以及封装一些第三方插件提供调用方法

MsgUtils.php为短信发送工具类封装了一些短信发送的逻辑, 它的配置在app.php的msg_options中, accessKeyld和accessKeySecret为阿里云身份标识product和domain以及region和endPointName为域名相关基本不用修改,sign为短信签名,目前也不用修改,templateCode中的属性为对应短信模板的模板号

sendSms为短信发送实际方法,他调用了AliyunSms.php的发送短信方法,该方法只供内部调用

BaseUUID.php封装了uuid相关一些操作, 它的配置在app.php的uuid_options中, flag为位数标识只传1, env 为环境标识 1-7,db为数据库标识1-255, server为服务标识1-255, startTime为时间戳开始统计时间1558713600000代表从2019/5/25 0:0:0开始算起

uuid规则 82位的二进制转换的16位36进制字符串, 前20位(4位)代表位数标识+环境标记+数据库标识+服务标识;中间31位(8位)代表时间戳部分标识;后31位(8位)代表时间戳部分标识+随机数+数据库表标识

flag(1)(一位):位数标识占一位只传1

env(1-7)(三位):环境标识 1.开发环境 2.测试环境 3.演示环境 4.正式环境 其他预留

db(1-255)(八位):数据库标识 1.开发数据库 2.测试数据库 3.演示数据库 4+.其他正式数据库

server(1-255)(八位):服务标识 1.开发服务 2.测试服务 3.演示服务 4+.其他正式服务

以上组成20位2进账(4位36进制),为uuid前四位

timestamp(0-4398046511103)(四十二位): 当前时间毫秒值与startTime(1558713600000->2019/5/25 0:0:0)时间戳值差,可以表示时间范围2019/5/25 0:0:0(1558713600000) - 2158/10/6 7:35:11(5956760111103)

timestamp二进制前31位组成8位的36进账,为uuid第五-十二位

random(0-2047)(十二位): 随机数

table(0-255)(八位): 预留字段,分表标识,标识最多可以分256张表

uuid()为生成uuid的方法

pasre()为解析uuid的方法,可以获取flag,env,db,server,timestamp,random,table等字段

pasreTable()为解析uuid获取分表字段的方法

注意,由于分表字段的存在,尽量使有关联关系的数据的\$table字段一致,远程诊断可能数据量不太大,举个dicom存储的例子,默认uuid()随机生成了患者id,考虑以后可能存在分表,通过患者id pasreTable拿到table字段并调用uuid(table)生成检查id,以此类推,一个患者下面,他的检查id,系列id,影像id的分表标识都是一样的,以后如果分库就可以去同一个库去查

- 15. validate

validate为数据验证目录

ZskkDefaultValidate继承自BaseValidate(祖父类为tp的Validate)

xxx为具体校验的目录 一般和model层xxx的对应

Xxx.php 为具体校验规则

XxxValidate 是封装的这些验证规则并提供静态的方式供其他类调用,该类一般对调用 ZskkErrorResponse的抛出参数异常的方法

- 16. version

version位版本记录目录

version.php暴露当前版本号

log目录下为版本更新描述,包括功能描述和数据库修改描述,bug维护等

sql目录下为项目初始化需要的SQL

sql/v1为项目最初版本SQL创建语句,其中只有build.php和build.sql,build.sql为数据库初始创建语句,仅供参考,build.php抛出数据库初始化具体执行语句

sql/vx为项目升级到x版本所需要的SQL语句,起目录下有build和up语句,同样.sql仅供参考,具体执行语句由.php封装抛出build.php提供当时数据库没有初始化的情况下,直接升级到vx版本的数据库SQL,up.php提供从vx-1版本升级到vx版本的数据库SQL,在数据库已经被初始化并且版本落后(vn)的情况下,需要从数据库当前版本依次逐步执行vn+1,vn+2,...vmax目录下的up.php

4. 其他

- 单元测试

单元测试使用的是TP的think-testing=2.0.*拓展,它是基于PHPUnit([中文文档](#))的 单元测试配置文件是/server/phpunit.xml 测试完成后会在/server/tmp下生成HTML和TXT格式的日志报告 单元测试命令为 php think unit /server/test.sh封装了该方