```
i5ting ztree toc:memo
Table of Content
1. ivdata-common
  1.1. —、jy_code
                                 1. jydata-common
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
                                   .1. —, jy_code
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
                                   .2. 二、jy_conf 公共配置
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
                                 1.3. 三、jy_crypt 加解密
    1.4.1. 1. 获取数据库对象
      1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
      T.2. 2. 蚁鸡芹ANSADI
1.4.2.1. a. select: 查询,适用于Oracle/3.1. 1. base_bit_num 基础数位
        1.4.2.1.1. 方法定义
        1.4.2.1.2. 示例
      1.4.2.2. b. update: 增删改, 适用于Ora3.1.1. a. 类定义
        1.4.2.2.1. 方法定义
        1.4.2.2.2. 示例
                                   :lass BaseBitNum(object):
      1.4.2.3. c. update batch: 批量更新, 一
        1.4.2.3.1. 方法定义
                                     def __init__(self, num_bit=10):
        1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
                                         初始化
  1.6. 六、jy_factory 工厂类
                                         :param num_bit: 数位 2/8/10/16/32...
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库
                                      @property
      1.8.1.1. a.设置对象属性
                                     def num_bit(self):
    1.8.2. 2. calender_utils: 日历工具方法库
                                         """ 属性方法: 获取数位 """
      1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库 @property
      1.8.4.1. a. 常量定义
                                      def bit_chars(self):
```

## 1.3.1.2. b. 已定义对象

1.8.4.2. b. 格式化时间元素数字 1843 c 附间戳线datetime

""" 属性方法: 字符序列 """

```
from jy crypt.base bit num import BaseBitNum
BINARY = BaseBitNum(num_bit=2) # 二进制
OCTAL = BaseBitNum(num_bit=8) # 8进制
NUM = BaseBitNum(num_bit=10) # 10进制
HEX = BaseBitNum(num bit=16) # 16进制
BASE32 = BaseBitNum(num_bit=32) # 32进制
BASE64 = BaseBitNum(num_bit=64) # 64进制
```

## 1.3.2. 2. bit num 进制数

#### 1.3.2.1. a. 类定义

```
from jy_crypt.base_bit_num import BaseBitNum, NUM
class BitNum(object):
   def __init__(self, val, bit=NUM):
       初始化
       :param val: 值
       :param bit: 数位
   def convert to bit(self, aim bit: BaseBitNum):
       进制转换
       :param aim_bit: 目标数位
       :return:
   @property
   def val_num(self) -> int:
       对应的十进制数值
```

```
:return:
Table of Content
1. jydata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                         def get_bit_char(self, m):
  1.3. 三、 jy_crypt 加解密
     1.3.1. 1. base_bit_num 基础数位
                                             数值转字符
       1.3.1.1. a. 类定义
                                             :param m: 数值
       1.3.1.2. b. 已定义对象
                                             :return:
     1.3.2. 2. bit_num 进制数
       1.3.2.1. a. 类定义
     1.3.3. 3. 佳缘线上加解密
                                         def get_bit_val(self, ch):
       1.3.3.1. a. 身份证号加密
       1.3.3.2. b. 身份证号解密
                                             字符转数值
       1.3.3.3. c. 姓名加解密
                                             :param ch: 字符
    1.3.4. 4. 佳缘vip加解密
                                             :return:
       1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
       1.4.1.1. a. 查看数据库配置信息
       1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
       4.2. 2. 数据年対象apı :rom jy_crypt.base_bit_num import BASE32
1.4.2.1.a. select: 査询, 适用于Oracle/l
'rom jy_crypt.bit_num import BitNum
         1.4.2.1.1. 方法定义
                                      32 bit num = BitNum(245, BASE32)
          1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac
         1.4.2.2.1. 方法定义
                                      .3.3. 3. 佳缘线上加解密
         1.4.2.2.2. 示例
       1.4.2.3. c. update batch: 批量更新, 一
         1.4.2.3.1. 方法定义
                                      3.3.1. a. 身份证号加密
          1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                      def encrypt_idc(idc: str):
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
                                         加密线上身份证号
     1.8.1. 1. attr_utils: 对象属性工具方法库
                                         :type idc: str 待加密的身份证号
       1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
       1.8.2.1. a.获取给定日期所在月份的工作日
     1.8.3. 3. condition_utils: 条件类工具方渍域列
     1.8.4. 4. date_utils: 日期时间类工具方法库
       1.8.4.1. a. 常量定义
                                      'rom jy_crypt.jyol_crypt import encrypt_idc
       1.8.4.2. b. 格式化时间元素数字
                                      !ncrypt_idc('350122198910116847')
       1843 c 时间戳转datetime
                                      ### 'fhcdeedlkldzddikgj'
                                    1.3.3.2. b. 身份证号解密
                                      def decrypt_idc(en_idc: str):
                                         解密线上身份证号
                                         :param en_idc: 待解密的身份证号密文
                                         :return:
                                    示例
                                     from jy_crypt.jyol_crypt import decrypt_idc
                                      decrypt_idc('fhcdeedlkldzddikgj')
                                      ### '350122198910116847'
                                    1.3.3.3. c. 姓名加解密
                                      def crypt_name(name: str):
                                         加解密姓名
                                         :param name: 待加(解)密的姓名 (密文)
                                         :return:
                                    示例
                                      from jy_crypt.jyol_crypt import crypt_name
                                      crypt_name('诸葛亮')
                                      ### '讏蒵仪'
```

crypt\_name('讏蒵仪') ### '诸葛亮'

```
Table of Content
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
       1.3.3.1. a. 身份证号加密
       1.3.3.2. b. 身份证号解密
       1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
       1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
       1.4.1.1. a. 查看数据库配置信息
       1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
         1.4.2.2.1. 方法定义
         1.4.2.2.2. 示例
       1.4.2.3. c. update batch: 批量更新. —
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
```

## 1.3.4. 4. 佳缘vip加解密

## 1.3.4.1. a. 佳缘vip数据加解密

```
lef convert(in str: str, convert type: int, dept=None):
  加解密 (数位转换)
  :param in_str: 待加 (解) 密字符串
  :param convert_type: 1:解密, 0:加密
  :param dept:
  :return:
```

#### 示例

```
'rom jy_crypt.jyvip_convert import convert
                                  # 解察
                                  :onvert('1j1m1i1k1j1g1h1p1p1j1g1h1h1g1g1g1h1p', 1)
1.4.2.1. a. select: 查询,适用于Oracle## '362430199301100019'
                                  # 加密
                                  :onvert('362430199301100019', 0)
1.4.2.2. b. update: 增删改,适用于Ora<sub>f##</sub> '1j1mli1k1j1g1h1p1p1j1g1h1h1g1g1g1h1p'
```

# .4. 四、jy\_db 数据库

## 1.4.1. 1. 获取数据库对象

#### 1.8.1. 1. attr\_utils: 对象属性工具方法库 4.1.1. a. 查看数据库配置信息

```
1.8.2. 2. calender_utils: 日历工具方法库
```

1.8.2.1. a.获取给定日期所在月份的工作日

1.8.3. 3. condition\_utils: 条件类工具方法库om jy\_db import show\_db\_confs

1.8.4. 4. date\_utils: 日期时间类工具方法库how\_db\_confs(db\_type='Orcl') # db\_type: Oracle/Mysql/Hive/Impala

1.8.4.1. a. 常量定义

1.8.1.1. a.设置对象属性

1.8.4.2. b. 格式化时间元素数字 1843 c 时间戳线datetime

## 4.1.2. b. 获取数据库对象

```
from jy_db import get_db
orcl_dc = get_db(db_type='Orcl', key='dc') # db_type同上, key为数据库配置信息中的键
orcl_huawei = get_db(db_type='Orcl', key='huawei')
mysql_crm = get_db(db_type='Mysql', key='crm')
hive_cloud = get_db(db_type='Hive', key='cloud')
impala_cloud = get_db(db_type='Impala', key='cloud')
```

## 1.4.2. 2. 数据库对象api

## 1.4.2.1. a. select: 查询,适用于Oracle/Mysql/Hive/Impala

#### 1.4.2.1.1. 方法定义

```
def select(self, sql, cursor=None, size=-1, params: tuple = None, key_pre='#', **kw):
   杏油粉据
   :type size: int 抓取结果条数, 默认取全部查询结果
   :param sql: select sql
   :param cursor:
   :param params: params of select sql
   :param key_pre: 命名参数标识符, 默认为#
   :param kw: 命名参数
   :return:
```

#### 1.4.2.1.2. 示例

```
from jy_db import get_db
orcl_dc = get_db('Orcl', 'dc')
mysql_crm = get_db('Mysql', 'crm')
# 完整sal, 无参数
data = orcl_dc.select("select * from city where city_new = '北京'")
```

```
注意事项:
Table of Content
                                                        1> 查询sql结尾不添加分号(;),下同。
1. ivdata-common
   1.1. —、jy_code
   1.2. 二、jy_conf 公共配置
   1.3. 三、 jy_crypt 加解密
       1.3.1. 1. base_bit_num 基础数位
                                                         params -- 参数列表, 必须是tuple类型: (param1, param2, ...)
          1.3.1.1. a. 类定义
                                                         lata = orcl_dc.select("select * from city where city_new = '%s' and id < %s", params=('北京', 11
          1.3.1.2. b. 已定义对象
       1.3.2. 2. bit_num 进制数
                                                        主意事项:
          1.3.2.1. a. 类定义
                                                        .> 查询sq1所有参数占位符必须是%s(int/float类型的参数也不例外,不采用%d/%f形式占位符),如果查询参数类型员
       1.3.3. 3. 佳缘线上加解密
                                                        !> 如果参数较多(>4)时,建议使用关键字参数形式。
          1.3.3.1. a. 身份证号加密
                                                        > 如果查询sql中有'%',不建议使用参数列表(params),可改用关键字参数形式
          1.3.3.2. b. 身份证号解密
                                                        ⇒ 特殊参数(系统定义, 在执行时会被转化成特定的值,同样适应于命名参数形式),要传确定的参数时应避开这些析
          1.3.3.3. c. 姓名加解密
                                                             特殊参数格式:类型(非必须)#可执行方法名?k=v(可执行方法参数,是否必须要看可执行方法名对应的方法是否有
      1.3.4. 4. 佳缘vip加解密
                                                              I: 日期时间字符串(datestr#{func_name}?k=v)
           1.3.4.1. a. 佳缘vip数据加解密
                                                                   指定日期(默认今天)开始时间: datestr#day_begin[?day={datetime}|year=%dmonth&month=%d&days=
    1.4. 四、jy_db 数据库
                                                                   指定日期(默认今天)结束时间: datestr#day_end[?day={datetime}|year=%dmonth&month=%d&days=%d
       1.4.1. 1. 获取数据库对象
                                                                   指定日期(默认今天)的昨天开始时间: datestr#yesterday_begin[?day={datetime}|year=%dmonth&mon
           1.4.1.1. a. 查看数据库配置信息
                                                                   指定日期(默认今天)的昨天结束时间: datestr#yesterday_end[?day={datetime}|year=%dmonth&mont|
          1.4.1.2. b. 获取数据库对象
                                                                   指定日期(默认今天)的月初开始时间: datestr#month_begin[?day={datetime}|year=%dmonth&month=
       1.4.2. 2. 数据库对象api
                                                                  指定日期(默认今天)的月末结束时间: datestr#month end[?day={datetime}|year=%dmonth&month=%d
           1.4.2.1. a. select: 查询,适用于Oracle/I
                                                                  指定日期(默认今天)的年初开始时间: datestr#year_begin[?day={datetime}|year=%dmonth&month=%
              1.4.2.1.1. 方法定义
              1.4.2.1.2. 示例
                                                                   指定日期(默认今天)的年末结束时间: datestr#year_end[?day={datetime}|year=%dmonth&month=%d&
           1.4.2.2. b. update: 增删改, 适用于Orac Ⅱ: 日期时间时间戳(datets#{func_name}?k=v)
              1.4.2.2.1. 方法定义
                                                                   指定日期(默认今天)开始时间: datets#day_begin[?day={datetime}|year=%dmonth&month=%d&days=%
              1.4.2.2.2. 示例
                                                                   指定日期(默认今天)结束时间: datets#day end[?day={datetime}|year=%dmonth&month=%d&days=%d]
           1.4.2.3. c. update_batch: 批量更新. —
                                                                  指定日期(默认今天)的昨天开始时间:datets#yesterday_begin[?day={datetime}|year=%dmonth&mon
              1.4.2.3.1. 方法定义
                                                                   指定日期(默认今天)的昨天结束时间: datets#yesterday_end[?day={datetime}|year=%dmonth&month
              1.4.2.3.2. 示例
                                                                   指定日期(默认今天)的月初开始时间: datets#month_begin[?day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month=%day={datetime}|year=%dmonth&month&month=%day={datetime}|year=%dmonth&month&month&month=%day={datetime}|year=%dmonth&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month&month
   1.5. 五、jy_entity 实体基类
                                                                   指定日期(默认今天)的月末结束时间: datets#month_end[?day={datetime}|year=%dmonth&month=%d&
    1.6. 六、jy_factory 工厂类
                                                                   指定日期(默认今天)的年初开始时间: datets#year_begin[?day={datetime}|year=%dmonth&month=%datetime}
    1.7. 七、jy_result 返回结果
                                                                   指定日期(默认今天)的年末结束时间: datets#year_end[?day={datetime}|year=%dmonth&month=%d&d
    1.8. 八、jy_util 工具/方法
                                                             Ⅲ: 无类型({func name}?k=v)
       1.8.1. 1. attr_utils: 对象属性工具方法库
                                                                   暂无, 待添加
          1.8.1.1. a.设置对象属性
       1.8.2. 2. calender_utils: 日历工具方法库
          1.8.2.1. a.获取给定日期所在月份的工作日
       1.8.3. 3. condition_utils: 条件类工具方法 kw -- 命名参数
       1.8.4. 4. date_utils: 日期时间类工具方法摩ata = orcl_dc.select("select * from city where city_new = '#name' and id < #id", name='北京', ic
          1.8.4.1. a. 常量定义
                                                        ‡意事项:
          1.8.4.2. b. 格式化时间元素数字
          1843 c 时间戳转datetime
                                                        _> 查询sql的查询参数类型是varchar/string时,对应的占位参数名应包含在引号内。
                                                        2> 命名参数可以和params同时使用, eg:
                                                        data = orcl_dc.select("select * from city where city_new = '%s' and id < #id", params=('北京',),
                                                        3> 如果默认命名参数标识符(#)和查询参数冲突时可以替换命名参数标识符(传入新的key_pre), eg:
                                                        data = orcl_dc.select("select * from city where city_new = '?name' and id < ?id", key_pre='?', n</pre>
                                                        # size -- 抓取结果条数
                                                        data = mysql_crm.select('select con_id, contract_num from contract order by con_id desc', size=5
                                                        注意事项:
                                                        1> 该参数不适用于Oracle
                                                        2> 该参数可以和以上参数同时使用。
```

## 1.4.2.2. b. update: 增删改, 适用于Oracle/Mysql/Hive/Impala(前提是有相应的 数据库权限)

#### 1.4.2.2.1. 方法定义

```
def update(self, sql, params=None, cursor=None, **kw):
   insert/update/delete data
   :param sql: insert/update/delete sql
    :param params: params of update sql
   :naram cursor:
   :return:
```

#### 1.4.2.2.2. 示例

```
# update
```

1.4.2.3. c. update\_batch: 批量更新, 一般用于批量Insert

lef update\_batch(self, sql, params=None, batch\_num=10000, cursor=None):

:param batch num: batch num, the num of per commit, default 10000

```
Table of Content
1. ivdata-common
  1.1. —、jy_code
```

1.2. 二、jy\_conf 公共配置

1.3. 三、 jy\_crypt 加解密

1.3.1. 1. base\_bit\_num 基础数位

1.3.1.1. a. 类定义 1.3.1.2. b. 已定义对象

1.3.2. 2. bit\_num 进制数 1.3.2.1. a. 类定义

1.3.3. 3. 佳缘线上加解密 1.3.3.1. a. 身份证号加密

1.3.3.2. b. 身份证号解密 1.3.3.3. c. 姓名加解密

1.3.4. 4. 佳缘vip加解密

1.3.4.1. a. 佳缘vip数据加解密

1.4. 四、jy\_db 数据库

1.4.1. 1. 获取数据库对象

1.4.1.1. a. 查看数据库配置信息 1.4.1.2. b. 获取数据库对象

1.4.2. 2. 数据库对象api

1.4.2.1. a. select: 查询,适用于Oracle/I

1.4.2.1.1. 方法定义

.5. 五、jy\_entity 实体基类

1.4.2.1.2. 示例 1.4.2.2. b. update: 增删改, 适用于Orac

1.4.2.2.1. 方法定义

1.4.2.2.2. 示例

1.4.2.3. c. update\_batch: 批量更新, 一**6.** 六、jy\_factory 工厂类 1.4.2.3.1. 方法定义

1.4.2.3.2. 示例

1.5. 五、jy\_entity 实体基类

1.6. 六、jy\_factory 工厂类

1.7. 七、jy\_result 返回结果

1.8. 八、jy\_util 工具/方法

1.8.1. 1. attr\_utils: 对象属性工具方法库

1.8.1.1. a.设置对象属性

1.8.2. 2. calender\_utils: 日历工具方法库 1.8. 八、jy util 工具/方法

1.4.2.3.1. 方法定义

:return:

1.4.2.3.2. 示例

batch insert/update/delete data

:param params: batch params(list)

:param sql: insert/update/delete sql

1.8.2.1. a.获取给定日期所在月份的工作日 1.8.3. 3. condition utils: 条件类工具方法库

1.8.4.2. b. 格式化时间元素数字

1843 c 时间戳线datetime

# 1.7. 七、jy\_result 返回结果

# 

## 1.8.1.1. a.设置对象属性

#### 方法定义

```
def set_attrs(o, **kwargs):
   设置对象属性,用于将dict赋值给对象属性
   :param o: 待设置属性对象
   :param kwargs: 需要设置的属性键值对
   :return: None
```

#### 示例

```
from jy_utils.attr_utils import set_attrs
class A:
   attr 1 = 0
    attr_2 = 0
a = A()
a.attr_1, a.attr_2
### (0, 0)
set_attrs(a, **{'attr_1': 1, 'attr_2': 2})
a.attr_1, a.attr_2
### (1, 2)
```

## 1.8.2. 2. calender\_utils: 日历工具方法库

## 1.8.2.1. a.获取给定日期所在月份的工作日序号list和休息日序号list

```
from datetime import datetime
def month work days(day: datetime, weekends=(5, 6), holidays: list = None, workdays: list = None
```

```
Table of Content
                                      获取给定日期所在月份的工作日序号list和休息日序号list
1. ivdata-common
                                      :param day: 给定日期
  1.1. —、jy_code
                                      :param weekends: 固定周休序号,默认为(5,6), 对应为(周六,周日)
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
                                      :param holidays: 自定义节假日(list)
    1.3.1. 1. base_bit_num 基础数位
                                      :param workdays: 自定义工作日(list),调班
      1.3.1.1. a. 类定义
                                      :return:(工作日序号list,休息日序号list)
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
                                   例
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
      1.3.3.3. c. 姓名加解密
                                   rom datetime import datetime
    1.3.4. 4. 佳缘vip加解密
                                   'rom jy_utils.calender_utils import month_work_days, month_nrd_work_day
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
                               # 无自定义节假日和自定义调休日
    1.4.1. 1. 获取数据库对象
                                   # 默认周休(周六、日双休)
      1.4.1.1. a. 查看数据库配置信息
                                   'esult = month_work_days(datetime.now())
      1.4.1.2. b. 获取数据库对象
                                   : result: (
    1.4.2. 2. 数据库对象api
                                      [2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, 30, 31],
      1.4.2.1. a. select: 查询,适用于Oracle/[ [1, 7, 8, 14, 15, 21, 22, 28, 29]
        1.4.2.1.1. 方法定义
                                   † )
         1.4.2.1.2. 示例
      fresult: (
         1.4.2.2.2. 示例
      1.4.2.3. c. update_batch: 批量更新, 上 [1, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29],
                                      [2, 3, 9, 10, 16, 17, 23, 24, 30, 31]
         1.4.2.3.1. 方法定义
                                   † )
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                  # 自定义节假日和调休日
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
                                   result = month_work_days(datetime(2021, 10, 1), workdays=(9,), holidays=(1,2,3,4,5,6,7))
    1.8.1. 1. attr_utils: 对象属性工具方法库 # result: (
      1.8.1.1. a.设置对象属性
                                   [8, 9, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29],
    1.8.2. 2. calender_utils: 日历工具方法库 # [1, 2, 3, 4, 5, 6, 7, 10, 16, 17, 23, 24, 30, 31]
      1.8.2.1. a.获取给定日期所在月份的工作日)
    1.8.3. 3. condition utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库
      1.8.4.1. a. 常量定义
```

## .8.3. 3. condition\_utils: 条件类工具方法库

## 1.8.4. 4. date\_utils: 日期时间类工具方法库

## 1.8.4.1. a. 常量定义

1.8.4.2. b. 格式化时间元素数字 1 8 4 3 c. 时间戳转datetime

```
from datetime import datetime
# 单位值
MONTHS = 12
HOURS = 23
MINUTES = 59
SECONDS = 59
# 日期时间格式
DATE_FMT = '%Y-%m-%d'
DATE_FMT0 = '\%y-\%m-\%d'
DATE FMT HMS = '%Y-%m-%d %H:%M:%S'
DATE_FMT_IMS = '%Y-%m-%d %I:%M:%S'
DATE_FMT0_HMS = '\%y-\%m-\%d \%H:\%M:\%S'
DATE_FMT0_IMS = '%y-%m-%d %I:%M:%S'
DATE_FMT_Ymd = '%Y%m%d'
DATE FMT ymd = '%y%m%d'
DATE_FMT_Ymd_HMS = '%Y%m%d %H:%M:%S'
DATE_FMT_Ymd_IMS = '%Y%m%d %I:%M:%S'
DATE_FMT_ymd_HMS = '%y%m%d %H:%M:%S'
DATE_FMT_ymd_IMS = '%y%m%d %I:%M:%S'
DAY_BEGIN = '00:00:00'
DAY BEGIN12 = '00:00:00 am'
DAY_END = '23:59:59
DAY_END12 = '23:59:59'
# 特殊时间点定义
ZERO DATETIME = datetime(1970, 1, 1)
DATE_20190101 = datetime(2019, 1, 1)
DATE 20210201 = datetime(2021, 2, 2)
```

```
Table of Content
                                  1.8.4.2. b. 格式化时间元素数字
1. ivdata-common
  1.1. —、jy_code
                                  方法定义
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
                                    lef format_numbers(*numbers: tuple, min_len=2, max_len=0, fill_num=0, fill_loc=1, sub_loc=-1):
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
                                       格式化时间元素数字
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                       :param numbers: 带格式化数字组
      1.3.2.1. a. 类定义
                                       :param min len: 最小长度
    1.3.3. 3. 佳缘线上加解密
                                       :param max_len: 最大长度
      1.3.3.1. a. 身份证号加密
                                       :param fill_num: 不足最小长度时填充的数字
      1.3.3.2. b. 身份证号解密
                                       :param fill loc: 填充位置: 1: 左填充; -1: 右填充
      1.3.3.3. c. 姓名加解密
                                       :param sub_loc: 超出长度截取位置: 1: 左截取; -1: 右截取
    1.3.4. 4. 佳缘vip加解密
                                       :return:
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
                                    個
       1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
                                    From jy utils.date utils import format numbers
      1.4.2.1. a. select: 查询,适用于Oracle/brmat_numbers(2021, 9, 23, 1, 29, 6)
         1.4.2.1.1. 方法定义
                                    ### ('2021', '09', '23', '01', '29', '06')
         1.4.2.1.2. 示例
                                    Format_numbers(2021, 9, 23, 1, 29, 6, max_len=2)
      Format_numbers(2021, 9, 23, 1, 29, 6, max_len=2, sub_loc=1)
         1.4.2.2.2. 示例
      1.4.2.3. c. update_batch: 批量更新, 一## ('20', '09', '23', '01', '29', '06')
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
                                   ..8.4.3. c. 时间戳转datetime
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                  方法定义
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库 def from_timestamp(timestamp):
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
                                       时间戳转datetime
      1.8.2.1. a.获取给定日期所在月份的工作日 :param timestamp: 时间戳
    1.8.3. 3. condition utils: 条件类工具方法库
                                       :return:
    1.8.4. 4. date_utils: 日期时间类工具方法库
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳转datetime
                                  ...例
                                    from jy_utils.date_utils import from_timestamp, now_timestamp
                                    from timestamp(0)
                                    ### datetime.datetime(1970, 1, 1, 8, 0)
                                    now timestamp()
                                    ### 1632374848.482117
                                    from_timestamp(1632374848.482117)
                                    ### datetime.datetime(2021, 9, 23, 13, 27, 28, 482117)
                                  1.8.4.4. d. 当前时间
                                  方法定义
                                    def now():
                                       """获取当前时间"""
                                  示例
                                    from jy_utils.date_utils import now
                                    now()
                                    ### datetime.datetime(2021, 9, 23, 13, 35, 56, 752159)
                                  1.8.4.5. e. 当前时间戳
                                  方法定义
                                    def now_timestamp():
                                       """获取当前时间戳"""
                                  示例
                                    from jy_utils.date_utils import now_timestamp
                                    now timestamp()
```

```
### 1632375409.143204
Table of Content
1. ivdata-common
  1.1. —、jy_code
                                   1.8.4.6. f. 当前时间字符串
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
                                    法定义
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
                                    lef now_str(fmt):
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
                                       获取当前时间字符串
    1.3.3. 3. 佳缘线上加解密
                                       :param fmt: 默认DATE_FMT_HMS
      1.3.3.1. a. 身份证号加密
                                       :return:
      1.3.3.2. b. 身份证号解密
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
                                    例
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
                                    From jy_utils.date_utils import now_str
      1.4.1.1. a. 查看数据库配置信息
                                    iow_str()
      1.4.1.2. b. 获取数据库对象
                                    ### '2021-09-23 13:39:10'
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询,适用于Oracle) OW_str(fmt='%Y%m%d')
                                    1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
      1.4.2.2. b. update: 增删改, 适用于Ora8.4.7. g. 当前日期字符串
         1.4.2.2.2. 示例
      1.4.2.3. c. update batch: 批量更新, 一法定义
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
                                    lef cur_day_str(fmt):
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                       获取当前日期字符串
  1.7. 七、jy_result 返回结果
                                       :param fmt: 默认DATE_FMT
  1.8. 八、jy_util 工具/方法
                                       :return:
    1.8.1. 1. attr_utils: 对象属性工具方法库
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
      1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库<sub>rom jy_utils.date_utils import cur_day_str</sub>
      1.8.4.1. a. 常量定义
                                    :ur_day_str()
      1.8.4.2. b. 格式化时间元素数字
                                    ## '2021-09-23'
      1843 c 时间戳转datetime
                                   cur_day_str(fmt='%Y%m%d')
                                    ### '20210923'
                                   1.8.4.8. h. 获取给定时间的时间戳
                                   方法定义
                                    def day_ts(day=None):
                                       获取给定时间的时间戳
                                       :param day: 默认为当前时间
                                       :return:
```

```
from jy_utils.date_utils import day_ts, datetime
day_ts()
### 1632375914.375838
day_ts(day=datetime(2021, 1, 1))
### 1609430400.0
```

#### 1.8.4.9. i. 获取给定日期开始时间

```
def day_begin(day=None, year=None, month=None, days=None):
   获取给定日期开始时间
   :param day: 默认为当前时间
   :param year:
   :param month:
   :param days:
```

```
:return:
Table of Content
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                   示例
  1.3. 三、jy_crypt 加解密
     1.3.1. 1. base_bit_num 基础数位
                                     'rom jy_utils.date_utils import day_begin
       1.3.1.1. a. 类定义
                                     lay_begin()
      1.3.1.2. b. 已定义对象
                                     ### datetime.datetime(2021, 9, 23, 0, 0)
     1.3.2. 2. bit_num 进制数
                                     lay_begin(year=2021, month=1, days=1)
      1.3.2.1. a. 类定义
                                     ### datetime.datetime(2021, 1, 1, 0, 0)
     1.3.3. 3. 佳缘线上加解密
       1.3.3.1. a. 身份证号加密
       1.3.3.2. b. 身份证号解密
                                     8.4.10. j. 获取给定日期结束时间
       1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
       1.3.4.1. a. 佳缘vip数据加解密
                                     法定义
  1.4. 四、jy_db 数据库
     1.4.1. 1. 获取数据库对象
                                     lef day_end(day=None, year=None, month=None, days=None):
       1.4.1.1. a. 查看数据库配置信息
       1.4.1.2. b. 获取数据库对象
                                        获取给定日期结束时间
    1.4.2. 2. 数据库对象api
       1.4.2.1. a. select: 查询,适用于Oracle/I :param day: 默认为当前时间
                                        :param year:
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac :param days:
                                        :return:
         1.4.2.2.1. 方法定义
         1.4.2.2.2. 示例
       1.4.2.3. c. update batch: 批量更新, 一
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
                                     from jy_utils.date_utils import day_end
  1.6. 六、jy_factory 工厂类
                                     day_end()
  1.7. 七、jy_result 返回结果
                                     ### datetime.datetime(2021, 9, 23, 23, 59, 59)
  1.8. 八、jy_util 工具/方法
     1.8.1. 1. attr_utils: 对象属性工具方法库 day_end(year=2021, month=1, days=1)
      1.8.1.1. a.设置对象属性
                                     ## datetime.datetime(2021, 1, 1, 23, 59, 59)
    1.8.2. 2. calender_utils: 日历工具方法库
       1.8.2.1. a.获取给定日期所在月份的工作日
     1.8.3. 3. condition_utils: 条件类工具方法 3.4.11. k. 获取给定日期昨天开始时间
     1.8.4. 4. date_utils: 日期时间类工具方法库
       1.8.4.1. a. 常量定义
                                     法定义
       1.8.4.2. b. 格式化时间元素数字
       1843 c 时间戳线datetime
                                     def yesterday_begin(day=None, year=None, month=None, days=None):
                                        获取给定日期昨天开始时间
                                        :param day: 默认为当前时间
                                        :param year:
                                        :param month:
                                        :param days:
                                   示例
                                     from jy_utils.date_utils import yesterday_begin
                                     yesterday_begin()
                                     ### datetime.datetime(2021, 9, 22, 0, 0)
                                     yesterday_begin(year=2021, month=1, days=1)
                                     ### datetime.datetime(2020, 12, 31, 0, 0)
                                   1.8.4.12. I. 获取给定日期昨天结束时间
                                   方法定义
                                     def yesterday_end(day=None, year=None, month=None, days=None):
                                        获取给定日期昨天结束时间
                                        :param day: 默认为当前时间
                                        :param year:
                                        :param month:
                                        :param days: 加减天数
```

```
Table of Content
                                      from jy utils.date utils import yesterday end
1. ivdata-common
                                      yesterday_end()
  1.1. —、jy_code
                                      ### datetime.datetime(2021, 9, 22, 23, 59, 59)
  1.2. 二、jy_conf 公共配置
                                      yesterday_end(year=2021, month=1, days=1)
  1.3. 三、 jy_crypt 加解密
                                      ### datetime.datetime(2020, 12, 31, 23, 59, 59)
     1.3.1. 1. base_bit_num 基础数位
       1.3.1.1. a. 类定义
                                       8.4.13. m. 获取给定日期月初时间
       1.3.1.2. b. 已定义对象
     1.3.2. 2. bit_num 进制数
       1.3.2.1. a. 类定义
                                       法定义
     1.3.3. 3. 佳缘线上加解密
       1.3.3.1. a. 身份证号加密
       1.3.3.2. b. 身份证号解密
                                       lef month begin(day=None, year=None, month=None, months=0):
       1.3.3.3. c. 姓名加解密
                                          获取给定日期月初时间
    1.3.4. 4. 佳缘vip加解密
       1.3.4.1. a. 佳缘vip数据加解密
                                          :param day: 默认为当前时间
  1.4. 四、jy_db 数据库
                                          :param year:
     1.4.1. 1. 获取数据库对象
                                          :param month:
       1.4.1.1. a. 查看数据库配置信息
                                          :param months:
       1.4.1.2. b. 获取数据库对象
                                          :return:
    1.4.2. 2. 数据库对象api
       1.4.2.1. a. select: 查询,适用于Oracle/L
         1.4.2.1.1. 方法定义
          1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac
         1.4.2.2.1. 方法定义
                                      From jy_utils.date_utils import month_begin
         1.4.2.2.2. 示例
                                       nonth begin()
       1.4.2.3. c. update_batch: 批量更新, __## datetime.datetime(2021, 9, 1, 0, 0)
         1.4.2.3.1. 方法定义
                                      nonth_begin(year=2021, month=1)
          1.4.2.3.2. 示例
                                      ### datetime.datetime(2021, 1, 1, 0, 0)
  1.5. 五、jy_entity 实体基类
                                      month_begin(months=-3)
  1.6. 六、jy_factory 工厂类
                                     ### datetime.datetime(2021, 6, 1, 0, 0)
  1.7. 七、jy_result 返回结果
                                      month_begin(year=2021, month=1, months=1)
  1.8. 八、jy_util 工具/方法
     1.8.1. 1. attr_utils: 对象属性工具方法库 ### datetime.datetime(2021, 2, 1, 0, 0)
       1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法梅 8.4.14. n. 获取给定日期月末时间 1.8.2.1. a.获取给定日期所在月份的工作日
     1.8.3. 3. condition utils: 条件类工具方法库
     1.8.4. 4. date_utils: 日期时间类工具方法方法定义
       1.8.4.1. a. 常量定义
       1.8.4.2. b. 格式化时间元素数字
                                       lef month_end(day=None, year=None, month=None, months=0):
       1843 c 时间戳线datetime
                                          获取给定日期月末时间
                                          :param day: 默认为当前时间
                                          :param year:
                                          :param month:
                                          :param months:
                                          :return:
                                     示例
                                      from jy_utils.date_utils import month_end
                                      month_end()
                                      ### datetime.datetime(2021, 9, 30, 23, 59, 59)
                                      month end(year=2021, month=1)
                                      ### datetime.datetime(2021, 1, 31, 23, 59, 59)
                                      month end(months=-1)
                                      ### datetime.datetime(2021, 8, 31, 23, 59, 59)
                                      month_end(year=2021, month=1, months=-1)
                                      ### datetime.datetime(2020, 12, 31, 23, 59, 59)
                                     1.8.4.15. o. 获取给定日期上个月月初时间
                                     方法定义
                                      def last_month_begin(day=None, year=None, month=None):
                                          获取给定日期上个月月初时间
                                          :param day: 默认为当前时间
                                          :param year:
                                          :param month:
                                          :return:
```

```
Table of Content
                                     from jy_utils.date_utils import last_month_begin
1. ivdata-common
                                     last_month_begin()
  1.1. —、jy_code
                                     ### datetime.datetime(2021, 8, 1, 0, 0)
  1.2. 二、jy_conf 公共配置
                                     last_month_begin(year=2021, month=1)
  1.3. 三、 jy_crypt 加解密
                                     ### datetime.datetime(2020, 12, 1, 0, 0)
    1.3.1. 1. base_bit_num 基础数位
       1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
                                     8.4.16. p. 获取给定日期上个月月末时间
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
                                     法定义
    1.3.3. 3. 佳缘线上加解密
       1.3.3.1. a. 身份证号加密
       1.3.3.2. b. 身份证号解密
                                     lef last_month_end(day=None, year=None, month=None):
       1.3.3.3. c. 姓名加解密
                                        获取给定日期上个月月末时间
    1.3.4. 4. 佳缘vip加解密
       1.3.4.1. a. 佳缘vip数据加解密
                                        :param day: 默认为当前时间
  1.4. 四、jy_db 数据库
                                        :param year:
    1.4.1. 1. 获取数据库对象
                                        :param month:
       1.4.1.1. a. 查看数据库配置信息
                                        :return:
       1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
       1.4.2.1. a. select: 查询,适用于Oracle/
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改,适用于Ora c_{COM} jy_utils.date_utils import last_month_end
         1.4.2.2.1. 方法定义
                                     last_month_end()
         1.4.2.2.2. 示例
                                     ### datetime.datetime(2021, 8, 30, 23, 59, 59)
       1.4.2.3. c. update_batch: 批量更新, —ast_month_end(year=2021, month=1)
         1.4.2.3.1. 方法定义
                                     ### datetime.datetime(2020, 12, 31, 23, 59, 59)
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                   1.8.4.17. q. 获取给定日期年初时间
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库方法定义
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库 def year_begin(day=None, year=None):
       1.8.2.1. a.获取给定日期所在月份的工作日 """
    1.8.3. 3. condition_utils: 条件类工具方法库 获取给定日期年初时间
    1.8.4. 4. date_utils: 日期时间类工具方法库 :param day: 默认为当前时间
       1.8.4.1. a. 常量定义
                                        :param year:
       1.8.4.2. b. 格式化时间元素数字
                                        :return:
       1843 c 时间戳转datetime
```

```
from jy_utils.date_utils import year_begin
year_begin()
### datetime.datetime(2021, 1, 1, 0, 0)
year_begin(year=2020)
### datetime.datetime(2020, 1, 1, 0, 0)
```

#### 1.8.4.18. r. 获取给定日期年末时间

#### 方法定义

```
def year_end(day=None, year=None):
"""

获取给定日期年末时间
:param day: 默认为当前时间
:param year:
:return:
"""
```

## 示例

```
from jy_utils.date_utils import year_end
year_end()
### datetime.datetime(2021, 12, 31, 23, 59, 59)
year_end(year=2020)
### datetime.datetime(2020, 12, 31, 23, 59, 59)
```

## 1.8.4.19. s. 当前日期开始时间字符串

```
i5ting ztree toc:memo
Table of Content
                                   def cur_day_begin_str():
1. ivdata-common
                                      """当前日期开始时间字符串"""
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                  1.8.4.20. t. 当前日期结束时间字符串
  1.3. 三、 jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
                                   法定义
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                   lef cur_day_end_str():
      1.3.2.1. a. 类定义
                                      """当前日期结束时间字符串"""
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
                                   8.4.21. u. 获取给定时间的字符串
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
                                   法定义
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
                                   lef day_str(day, fmt):
      1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
                                      获取给定时间的字符串
    1.4.2. 2. 数据库对象api
                                      :param day:
      1.4.2.1. a. select: 查询, 适用于Oracle/I
                                      :param fmt: 时间格式,默认DATE_FMT_HMS
        1.4.2.1.1. 方法定义
                                      :return:
         1.4.2.1.2. 示例
      1.4.2.2. b. update: 增删改, 适用于Orac
         1.4.2.2.1. 方法定义
      1.4.2.3. c. update_batch: 批量更新, 一 8.4.22. v. 获取当前日期开始时间
         1.4.2.3.1. 方法定义
                                   法定义
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                   def cur_day_begin():
  1.7. 七、jy_result 返回结果
                                      """当前日期开始时间"""
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库
                                   8.4.23. w. 获取当前日期结束时间
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
    1.8.2.1. a.获取给定日期所在月份的工作日本定义
1.8.3. 3. condition_utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库
      1.8.4.1. a. 常量定义
                                   lef cur_day_end():
                                      """当前日期结束时间"""
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳线datetime
                                  1.8.4.24. x. 昨天开始时间字符串
                                  方法定义
                                   def yesterday_str(fmt):
                                      昨天开始时间字符串
                                      :param fmt: 格式, 默认DATE_FMT_HMS
                                      :return:
                                  1.8.4.25. y. 给定日期加days天
```

## 方法定义

```
def add_days(day, days):
   加days天
   :param day:
   :param days:
   :return:
```

#### 示例

```
from jy_utils.date_utils import add_days, datetime
add_days(datetime(2021, 9, 1), 11)
### datetime.datetime(2021, 9, 12, 0, 0)
### datetime.datetime(2021, 9, 20, 14, 28, 32, 663742)
```

## 1.8.4.26. z. 给定日期加months月

```
Table of Content
                                方法定义
1. jydata-common
  1.1. —、jy_code
                                  def add_months(day, months, p_day=0):
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
                                     加monthss月
    1.3.1. 1. base_bit_num 基础数位
                                     :param day:
      1.3.1.1. a. 类定义
                                     :param months:
      1.3.1.2. b. 已定义对象
                                     :param p_day:
    1.3.2. 2. bit_num 进制数
                                     :return:
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
                                  例
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
                                  'rom jy_utils.date_utils import add_months, datetime
      1.3.4.1. a. 佳缘vip数据加解密
                               add_months(datetime(2021, 9, 1), 1)
  1.4. 四、jy_db 数据库
                                 ### datetime.datetime(2021, 10, 1, 0, 0)
    1.4.1. 1. 获取数据库对象
                                 ıdd_months(-3)
      1.4.1.1. a. 查看数据库配置信息
                                  ### datetime.datetime(2021, 6, 23, 14, 31, 14)
      1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
      1.4.2.1.2. 示例
      1.4.2.2. b. update: 增删改, 适用于Ora法定义
        1.4.2.2.1. 方法定义
      lef month_first(day, days=1):
        1.4.2.3.1. 方法定义
                                     给定时间的月初
        1.4.2.3.2. 示例
                                     :param day:
  1.5. 五、jy_entity 实体基类
                                     :param days: 日期偏移量
  1.6. 六、jy_factory 工厂类
                                     :return:
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库
    1.8.2. 2. calender_utils: 日历工具方法库 .8.4.28. ab. 获取给定时间的月初字符串
      1.8.1.1. a.设置对象属性
      1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition_utils: 条件类工具方法法定义
    1.8.4. 4. date_utils: 日期时间类工具方法库
      1.8.4.1. a. 常量定义
                                  lef month_first_str(day, fmt, hms=False):
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳转datetime
                                     给定时间的月初字符串
                                     :param day:
                                     :param fmt:
                                     :param hms:
                                     :return:
                                1.8.4.29. ac. 获取给定时间的月末时间
                                方法定义
                                  def month_last(day):
```

```
def month_last(day):
"""
给定时间的月末
:param day:
:return:
"""
```

## 1.8.4.30. ad. 获取给定时间的月末字符串

方法定义

```
def month_last_str(day, fmt, hms=False):
"""
给定时间的月末字符串
:param day:
:param fmt:
:param hms:
:return:
"""
```

## 1.8.4.31. ae. 获取给定日期当月的天数

```
Table of Content
                                    def month_days(day):
1. ivdata-common
  1.1. —、jy_code
                                       获取给定日期当月的天数
  1.2. 二、jy_conf 公共配置
                                       :param day:
  1.3. 三、 jy_crypt 加解密
                                       :return: (当月第一天的星期序号[0:6], 当月总天数)
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
                                    例
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
                                    'rom jy_utils.date_utils import month_days, datetime
      1.3.3.1. a. 身份证号加密
                                    ionth_days(datetime(2021, 8, 23))
      1.3.3.2. b. 身份证号解密
                                    ### (6, 31)
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
                                    8.4.32. af. 字符串转日期
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
                                    法定义
       1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
                                    lef str_to_date(st, fmt):
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询, 适用于Oracle/l """
                                       字符串转日期
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
                                       :param st: 字符串
       1.4.2.2. b. update: 增删改, 适用于Orac :param fmt: 格式, 默认: DATE_FMT
         1.4.2.2.1. 方法定义
                                       :return:
         1.4.2.2.2. 示例
      1.4.2.3. c. update batch: 批量更新, —
         1.4.2.3.1. 方法定义
                                    8.4.33. ag. 判断给定的值是否为数字
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                  方法定义
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库 def is_number(val):
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
                                      判断给定的值是否为数字
      1.8.2.1. a.获取给定日期所在月份的工作日 :param val:
    1.8.3. 3. condition_utils: 条件类工具方法库 :return:
    1.8.4. 4. date_utils: 日期时间类工具方法库 """
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
                                  ..8.4.34. ah. 值转时间
      1843 c 时间戳转datetime
```

#### 方法定义

```
def convert_val_to_datetime(val, fmt):
    """
    值转时间
    :param val: 值
    :param fmt: 格式,默认: DATE_FMT_HMS
    :return:
    """
```

## 1.8.5. 5. dict\_utils: 字典类工具方法库

## 1.8.5.1. a.深度更新dict

## 方法定义

```
def deep_update(d1: dict, d2: dict, distinct=False):
    """

深度更新dict, 如果源字典(d1)key对应的值是集合类对象,则如果d2存在相同的key,则合并d1[v]和d2[v],而2:param d1: 源dict
:param d2:
:param distinct: 合并集合对象是否去重,默认不去重
:return: 更新后的源dict, 即d1
    """
```

#### 示例

## 1.8.5.2. b.左保全合并dict

```
Table of Content
                                  方法定义
1. ivdata-common
  1.1. —、jy_code
                                   def left_union(d1: dict, d2: dict):
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
                                      左保全合并dict
    1.3.1. 1. base_bit_num 基础数位
                                      :param d1: 源dict
      1.3.1.1. a. 类定义
                                      :param d2:
      1.3.1.2. b. 已定义对象
                                      :return: 更新后的源dict, 即d1
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
                                   例
      1.3.3.2. b. 身份证号解密
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
                                  1.8.6. 6. excel utils: excel文件类工具方法库
    1.4.1. 1. 获取数据库对象
      1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
      4.2. 2. 数据库对象api .8.7. 7.file_utils: 文件类工具方法库 1.4.2.1. a. select: 查询,适用于Oracle/l
    1.4.2. 2. 数据库对象api
        1.4.2.1.1. 方法定义
                                   8.7.1. a.从文本文件中读取字符串
         1.4.2.1.2. 示例
      1.4.2.2. b. update: 增删改, 适用于Orac
        1.4.2.2.1. 方法定义
                                   法定义
         1.4.2.2.2. 示例
      1.4.2.3. c. update batch: 批量更新. —
                                   lef read_str(path, size=-1, mode='r', encoding='utf-8'):
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
                                      从文本文件中读取字符串
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                      :param path: 文本文件路径
  1.7. 七、jy_result 返回结果
                                      :param size: 读取大小, -1: 全量读取
  1.8. 八、jy_util 工具/方法
                                     :param mode: 读取模式
    1.8.1. 1. attr_utils: 对象属性工具方法库
                                      :param encoding: 文件编码
      1.8.1.1. a.设置对象属性
                                      :return: str
    1.8.2. 2. calender_utils: 日历工具方法库
      1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法 1.8.7.2. b.从文件中读取每行内容,return list
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
                                   法定义
      1843 c 时间戳线datetime
                                   def read_lines(path, mode='r', encoding='utf-8'):
                                      从文件中读取每行内容, return list
                                      :param path: 文本文件路径
                                      :param mode: 读取模式
                                      :param encoding: 文件编码
                                      :return: list
                                  1.8.7.3. c.获取文件指定行内容
                                  方法定义
                                   def get_line(path, lineno, module_globals=None):
                                      Get the lines for a Python source file from the cache.
                                      :param path: 文件路径
                                      :param lineno: 行号
                                      :param module_globals:
                                      :return:
                                  1.8.7.4. d.分批从大文件中读取
                                  方法定义
                                   def read_block(path, size=-1, mode='r', encoding='utf-8'):
                                      分批从大文件中读取
                                      :param path: 文件路径
```

:param size: 每次读取大小 :param mode: 模式 :param encoding: 编码

```
:return: generator
Table of Content
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                  1.8.7.5. e.按行读取大文件,返回生成器
  1.3. 三、 jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
                                    法定义
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                    lef read_line(path, mode='r', encoding='utf-8'):
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
                                       按行读取大文件
      1.3.3.1. a. 身份证号加密
                                       :param path: 文件路径
      1.3.3.2. b. 身份证号解密
                                       :param mode: 模式
      1.3.3.3. c. 姓名加解密
                                      :param encoding: 编码
    1.3.4. 4. 佳缘vip加解密
                                       :return: generator
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
       1.4.1.1. a. 查看数据库配置信息
                                    8.7.6. f.写入数据到文件
      1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询, 适用于Oracle, 祛定义
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
                                    lef write_file(path, data: str, mode='a+', encoding='utf-8'):
       1.4.2.2. b. update: 增删改, 适用于Orac """
         1.4.2.2.1. 方法定义
                                       写入数据到文件
         1.4.2.2.2. 示例
      1.4.2.2.2. 亦例
1.4.2.3. c. update_batch: 批量更新, 一
         1.4.2.3.1. 方法定义
                                       :param mode: 文件打开模式
         1.4.2.3.2. 示例
                                       :param encoding: 文件编码
  1.5. 五、jy_entity 实体基类
                                       :return:
  1.6. 六、jy_factory 工厂类
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库 8.7.7. g. 写入多行数据到文件
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
      1.8.2.1. a.获取给定日期所在月份的工作法定义
    1.8.3. 3. condition utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库<sub>lef write_file_by_lines(path, lines, mode='a+', encoding='utf-8'):</sub>
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
                                       写入多行数据到文件
      1843 c 时间戳转datetime
                                       :param path: 写入路径
                                       :param lines: 待写入数据
                                       :param mode: 文件打开模式
                                       :param encoding: 文件编码
```

## 1.8.7.8. h.检查数据是否有换行符,如果没有则添加

#### 方法定义

```
def check_linesep(line: str):
    """
    检查数据是否有换行符,如果没有则添加
:param line:
:return:
    """
```

## 1.8.7.9. i.移除换行符

#### 方法定义

```
def rmv_linesep(line: str, only_tail=1):
    """
    移除末尾的换行符
    :param line: 数据行
    :param only_tail: 只移除末尾
    :return:
    """
```

## 1.8.8. 8. func utils:

## 1.8.8.1. a.注册方法

```
Table of Content
                                     方法定义
1. ivdata-common
  1.1. —、jy_code
                                       def regist_func(name=None, app=None, level=1, default_params=None, replace=False):
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
                                          regist func for later use
     1.3.1. 1. base_bit_num 基础数位
                                          :param name:
       1.3.1.1. a. 类定义
                                          :param app:
       1.3.1.2. b. 已定义对象
                                          :param level:
     1.3.2. 2. bit_num 进制数
                                          :param default params:
       1.3.2.1. a. 类定义
                                          :param replace:
     1.3.3. 3. 佳缘线上加解密
                                          :return:
       1.3.3.1. a. 身份证号加密
       1.3.3.2. b. 身份证号解密
       1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
                                       例
       1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
                                       <sup>3</sup>rom jy_utils.func_utils import regist_func
     1.4.1. 1. 获取数据库对象
       1.4.1.1. a. 查看数据库配置信息
                                       # 定义执行方法
       1.4.1.2. b. 获取数据库对象
                                       @regist_func(name="", app=app, )
     1.4.2. 2. 数据库对象api
       1.4.2.1. a. select: 查询, 适用于Oracle/lef method(*args, **kw):
         1.4.2.1.1. 方法定义
                                          pass
          1.4.2.1.2. 示例
       1.4.2.1.2. ハアッ
1.4.2.2. b. update: 増删改,适用于Orac
8.8.2. b.执行任务
          1.4.2.2.2. 示例
       1.4.2.3. c. update_batch: 批量更新, 一法定义
          1.4.2.3.1. 方法定义
          1.4.2.3.2. 示例
                                       def run_task(app, log_model=None):
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                          执行任务
  1.7. 七、jy_result 返回结果
                                          :param app: 任务所在app
  1.8. 八、jy_util 工具/方法
                                          :param log_model: 执行日志数据模型
     1.8.1. 1. attr_utils: 对象属性工具方法库
       1.8.1.1. a.设置对象属性
     1.8.2. 2. calender_utils: 日历工具方法库
       1.8.2.1. a.获取给定日期所在月份的工作日
     1.8.3. 3. condition_utils: 条件类工具方法库列1.8.4. 4. date_utils: 日期时间类工具方法库
       1.8.4.1. a. 常量定义
                                       'rom jy_utils.func_utils import run_task
       1.8.4.2. b. 格式化时间元素数字
       1843 c 时间戳线datetime
                                       ## 定义执行方法
                                       @run_task(app=app, log_model=None)
                                       def task(*args, **kw):
                                     1.8.8.3. c.重试
                                     方法定义
                                       def retry(wait_rule='5:3:15', split='', def_result=None):
```

```
def retry(wait_rule='5:3:15', split='', def_result=None):
    """
    重试
    :param wait_rule: 重试等待时间规则 '%d:%d'% (最多重试次数,等待时间,最大超时时间)
    :param split:
    :param def_result:
    :return:
    """
```

```
from jy_utils.func_utils import run_task

## 定义执行方法
@run_task(app=app, log_model=None)
def task(*args, **kw):
    pass
```

## 1.8.9. 9. gps\_utils:

## 1.8.9.1. a.gps坐标转换类

方法列表

```
Table of Content
                                    def gcj02_to_bd09(self, gcj_lng, gcj_lat):
1. ivdata-common
  1.1. —、jy_code
                                       实现GCJ02向BD09坐标系的转换
  1.2. 二、jy_conf 公共配置
                                       :param lng: GCJ02坐标系下的经度
  1.3. 三、jy_crypt 加解密
                                       :param lat: GCJ02坐标系下的纬度
    1.3.1. 1. base_bit_num 基础数位
                                       :return: 转换后的BD09下经纬度
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                    lef bd09_to_gcj02(self, bd_lng, bd_lat):
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
                                       实现BD09坐标系向GCJ02坐标系的转换
      1.3.3.1. a. 身份证号加密
                                       :param bd_lng: BD09坐标系下的经度
      1.3.3.2. b. 身份证号解密
                                       :param bd_lat: BD09坐标系下的纬度
      1.3.3.3. c. 姓名加解密
                                       :return: 转换后的GCJ02下经纬度
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
                                    lef wgs84_to_gcj02(self, lng, lat):
       1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
                                       实现WGS84坐标系向GCJ02坐标系的转换
    1.4.2. 2. 数据库对象api
                                       :param lng: WGS84坐标系下的经度
       1.4.2.1. a. select: 查询,适用于Oracle/I :param lat: WGS84坐标系下的纬度
         1.4.2.1.1. 方法定义
                                       :return: 转换后的GCJ02下经纬度
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac
         1.4.2.2.1. 方法定义
                                    lef gcj02_to_wgs84(self, gcj_lng, gcj_lat):
         1.4.2.2.2. 示例
       1.4.2.3. c. update batch: 批量更新. —
                                       实现GCJ02坐标系向WGS84坐标系的转换
         1.4.2.3.1. 方法定义
                                       :param gcj_lng: GCJ02坐标系下的经度
         1.4.2.3.2. 示例
                                       :param gcj_lat: GCJ02坐标系下的纬度
  1.5. 五、jy_entity 实体基类
                                       :return: 转换后的WGS84下经纬度
  1.6. 六、jy_factory 工厂类
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库 def bd09_to_wgs84(self, bd_lng, bd_lat):
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
                                       实现BD09坐标系向WGS84坐标系的转换
      1.8.2.1. a. 获取给定日期所在月份的工作日 : param bd_lng: BD09坐标系下的经度
    1.8.3. 3. condition utils: 条件类工具方法库
                                       :param bd_lat: BD09坐标系下的纬度
    1.8.4. 4. date_utils: 日期时间类工具方法库
                                       :return: 转换后的WGS84下经纬度
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
       1843 c 时间戳转datetime
                                    def wgs84_to_bd09(self, lng, lat):
                                       实现WGS84坐标系向BD09坐标系的转换
                                       :param lng: WGS84坐标系下的经度
                                       :param lat: WGS84坐标系下的纬度
                                       :return: 转换后的BD09下经纬度
                                    def wgs84_to_webmercator(self, lng, lat):
                                       实现WGS84向web墨卡托的转换
                                       :param lng: WGS84经度
                                       :param lat: WGS84纬度
                                       :return: 转换后的web墨卡托坐标
                                    def webmercator_to_wgs84(self, x, y):
                                       实现web墨卡托向WGS84的转换
                                       :param x: web墨卡托x坐标
                                       :param y: web墨卡托y坐标
                                       :return: 转换后的WGS84经纬度
```

## 1.8.10. 10. http\_utils: 网络连接类工具方法库

#### 1.8.10.1. a.邮件通知

```
from jy_utils.http_utils import jy_mail
# 定义执行方法
```

```
Table of Content
                                  jy_mail 参数:
1. ivdata-common
                                   subject: 邮件主题, 不传则默认为: 监控情况
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                   _from: 发送人,不传则默认为jf_jk@jiayuan.com
  1.3. 三、 jy_crypt 加解密
                                   passwd: 发送人邮箱密码,默认发送人(jf_jk@jiayuan.com)不需要传此参数
                                   _to: 接收人,多个接收人以','分割
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
                                   file_name: 附件名字,如果不需要发送附件则不传
      1.3.1.2. b. 已定义对象
                                   file_path: 附件对应的文件路径,如果不需要发送附件则不传
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
                                   )jy_mail(subject='test', _to='zhangjiawen@jiayuan.com', file_name='test', file_path='/u01/work/
    1.3.3. 3. 佳缘线上加解密
                                   lef test():
      1.3.3.1. a. 身份证号加密
                                    result = ''
      1.3.3.2. b. 身份证号解密
                                     result = 'test' ### 方法执行内容
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
                                     # 执行结果写入文件
      1.3.4.1. a. 佳缘vip数据加解密
                                     with open('/u01/work/jiayuan/test.txt', 'w') as f:
  1.4. 四、jy_db 数据库
                                         f.write(result)
    1.4.1. 1. 获取数据库对象
      1.4.1.1. a. 查看数据库配置信息
                                     # 返回执行结果,执行结果自动邮件发出
      1.4.1.2. b. 获取数据库对象
                                     return result
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询, 适用于Oracle/I
                                  : 调用方法
        1.4.2.1.1. 方法定义
        1.4.2.1.2. 示例
                                  :est()
      1.4.2.2. b. update: 增删改, 适用于Orac
        1.4.2.2.1. 方法定义
      1.4.2.3. c. update_batch: 批量更新,一样
        1.4.2.2.2. 示例
        1.4.2.3.1. 方法定义
                                   8.10.2. b. 微信通知
        1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                  from jy_utils.http_utils import weixin
  1.7. 七、jy_result 返回结果
                                  # 定义执行方法
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库
                                   reixin参数:
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库 user: 消息接收人(企业微信账号),通知多人以'|'分割
      1.8.2.1. a.获取给定日期所在月份的工作日""
    1.8.3. 3. condition_utils: 条件类工具方法向deixin('zhangjiawen')
    1.8.4. 4. date_utils: 日期时间类工具方法库ef test():
      1.8.4.1. a. 常量定义
                                     result = 'test' ### 方法执行内容
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳转datetime
                                     # 返回执行结果,执行结果自动企业微信通知接收人
                                      return result
                                 jy_weixin_notify
                                 1.8.11. 11. instance utils:
                                 1.8.11.1. a.获取实例
                                 方法定义
                                  def instance(cls):
                                      获取实例
                                     :param cls: 实例类型
                                      :return:
                                 示例
                                 1.8.12. 12. json utils:
```

## 1.8.12.1. a.对象转json字符串

```
def dumps(item):
"""
对象转json字符串
```

```
:param item: 对象
Table of Content
                                       :return:
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
  1.3. 三、jy_crypt 加解密
                                  示例
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
                                    8.12.2. b.json字符串转对象
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
                                    法定义
      1.3.3.2. b. 身份证号解密
      1.3.3.3. c. 姓名加解密
    1.3.4. 4. 佳缘vip加解密
                                    lef loads(json_str, encoding='utf8', **kw):
      1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
                                       json字符串转对象
    1.4.1. 1. 获取数据库对象
                                       :param json_str:
       1.4.1.1. a. 查看数据库配置信息
                                       :param encoding:
      1.4.1.2. b. 获取数据库对象
                                       :param kw:
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询,适用于Oracle/I :return:
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Ora何
         1.4.2.2.1. 方法定义
         1.4.2.2.2. 示例
       1.4.2.3. c. update_batch: 批量更新, 一
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
                                  ..8.12.3. c.从jsonfile加载对象
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
                                  方法定义
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库 def load_json_file(path, encoding='utf8', **kw):
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
                                       从jsonfile加载对象
      1.8.2.1. a.获取给定日期所在月份的工作日 :param path:
    1.8.3. 3. condition_utils: 条件类工具方法库
                                       :param encoding:
    1.8.4. 4. date_utils: 日期时间类工具方法库
                                       :param kw:
      1.8.4.1. a. 常量定义
                                       :return:
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳转datetime
                                  示例
                                  1.8.13. 13. list_utils:
                                  1.8.13.1. a.获取对象集合中符合条件的第一个对象
                                  方法定义
                                   def first(items, condition=None, def_val=None):
                                       获取对象集合中符合条件的第一个对象
                                       :param items: 对象集合
                                       :param condition: 过滤条件
                                       :param def_val: 默认返回值
                                       :return:
                                  示例
                                  1.8.13.2. b.list/tuple转set
                                  方法定义
                                   def to_set(a):
                                       list/tuple转set
```

:param a:

```
:return:
Table of Content
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                  示例
  1.3. 三、jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                    8.13.3. c.list/tuple/set 求并集
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
                                    法定义
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
      1.3.3.3. c. 姓名加解密
                                    lef union(a, b, distinct=False):
    1.3.4. 4. 佳缘vip加解密
      1.3.4.1. a. 佳缘vip数据加解密
                                       list/tuple/set 求并集
  1.4. 四、jy_db 数据库
                                       :param a: 第一个集合
    1.4.1. 1. 获取数据库对象
                                       :param b: 第二个集合
       1.4.1.1. a. 查看数据库配置信息
                                       :param distinct: 结果是否去重
      1.4.1.2. b. 获取数据库对象
                                       :return:
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询, 适用于Oracle/I
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac
         1.4.2.2.1. 方法定义
         1.4.2.2.2. 示例
      1.4.2.3. c. update_batch: 批量更新, 一
         1.4.2.3.1. 方法定义
                                    8.13.4. d.返回第n个元素
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
                                  方法定义
  1.6. 六、jy_factory 工厂类
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
                                    def nrd(items, n, last_replace=False):
    1.8.1. 1. attr_utils: 对象属性工具方法库
      1.8.1.1. a.设置对象属性
                                       返回第n个元素
    1.8.2. 2. calender_utils: 日历工具方法库
                                       :param items:
      1.8.2.1. a.获取给定日期所在月份的工作日 :param n: 第n个
    1.8.3. 3. condition_utils: 条件类工具方法库
                                       :param last_replace: 如果元素个数不足是否用最后一个代替,如果否,元素不足时返回None
    1.8.4. 4. date_utils: 日期时间类工具方法库
                                       :return:
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳线datetime
                                  示例
```

## 1.8.14. 14. log utils

## 1.8.14.1. a.输出日志

方法定义

```
def log(app=None, msg='', level='info'):
   输出日志
   :param app: 所在app(flask启动对象)
   :param msg: 日志内容
   :param level: 日志级别
   :return:
```

## 1.8.15. 15. mobile\_utils

#### 1.8.15.1. a.随机生成1个手机号

```
def random_mobile(pries=None):
   随机生成1个手机号
   :param pries: 指定前缀list
   :return:
```

```
Table of Content
                                  示例
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
  1.3. 三、jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
                                    8.15.2. b.随机生成n个手机号
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
                                   法定义
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
                                   {\tt lef\ random\_mobiles(n,\ pries=None):}\\
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
                                      随机生成n个手机号
      1.3.3.3. c. 姓名加解密
                                      :param n: 生成手机号数量
    1.3.4. 4. 佳缘vip加解密
                                      :param pries: 指定前缀list
      1.3.4.1. a. 佳缘vip数据加解密
                                      :return: 包含n个手机号的生成器
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
      1.4.1.1. a. 查看数据库配置信息
                                    例
      1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询,适用于Oracle/I
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
      1.4.2.2. b. update: 增删改, 适用于Ora%.16. 16. model_utils
         1.4.2.2.1. 方法定义
         1.4.2.2.2. 示例
      1.4.2.3. c. update_batch: 批量更新, 一法定义
         1.4.2.3.1. 方法定义
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
  1.6. 六、jy_factory 工厂类
  1.7. 七、jy_result 返回结果
                                  1.8.17. 17. num utils
  1.8. 八、jy_util 工具/方法
    1.8.1. 1. attr_utils: 对象属性工具方法库
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库 .8.17.1. a. 常量定义
      1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition_utils: 条件类工具方法库x_INT = 2**32-1 # 最大整数
    1.8.4. 4. date_utils: 日期时间类工具方法库
      1.8.4.1. a. 常量定义
                                    法定义
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳线datetime
                                  1.8.18. 18. obj_utils
                                  1.8.18.1. a.根据type_name获取type
                                  方法定义
                                   def name2class(name: str):
                                      根据type_name获取type
                                      :param name: type_name
                                  示例
                                  1.8.18.2. b.对象序列化成key
                                  方法定义
                                   def serializable_key(key, add_cls=False, cls_k='class'):
                                      序列化key
                                      :param key:
                                      :param add_cls:
                                      :param cls k:
                                      :return:
```

```
Table of Content
1. ivdata-common
  1.1. —、jy_code
                                   1.8.18.3. c.对象转dict
  1.2. 二、jy_conf 公共配置
  1.3. 三、 jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
                                     法定义
      1.3.1.1. a. 类定义
      1.3.1.2. b. 已定义对象
                                     fef obj2dict(o, add_cls=False, cls_k='class'):
    1.3.2. 2. bit_num 进制数
      1.3.2.1. a. 类定义
                                        对象转dict
    1.3.3. 3. 佳缘线上加解密
                                        :param o: 待转对象
       1.3.3.1. a. 身份证号加密
                                       :param add_cls: 添加类型
       1.3.3.2. b. 身份证号解密
       1.3.3.3. c. 姓名加解密
                                       :param cls_k: 结果中类型字段对应的k值,仅在默认('cls')与代转对象属性冲突时使用
    1.3.4. 4. 佳缘vip加解密
                                        :return:
       1.3.4.1. a. 佳缘vip数据加解密
  1.4. 四、jy_db 数据库
    1.4.1. 1. 获取数据库对象
                                     例
       1.4.1.1. a. 查看数据库配置信息
       1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
       1.4.2.1. a. select: 查询, 适用于Oracle/I
         1.4.2.1.1. 方法定义
                                     8.18.4. d.dict 转对象
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac
         1.4.2.2.1. 方法定义
                                     法定义
         1.4.2.2.2. 示例
       1.4.2.3. c. update_batch: 批量更新, — lef dict2obj(d: dict, cls_k='class', copy=False, fields_dict=None):
         1.4.2.3.2. 示例
                                        dict 转对象
  1.5. 五、jy_entity 实体基类
                                        :param d: 待转dict
  1.6. 六、jy_factory 工厂类
                                        :param cls k: 类型字段对应的k值
  1.7. 七、jy_result 返回结果
                                        :param copy: 是否全复制对象
  1.8. 八、jy_util 工具/方法
                                        :param fields_dict: 参与创建对象的字段列表{cls: {dict_key: obj_param_name}}}
    1.8.1. 1. attr_utils: 对象属性工具方法库
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
       1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition_utils: 条件类工具方法库
1.8.4. 4. date_utils: 日期时间类工具方法库
       1.8.4.1. a. 常量定义
       1.8.4.2. b. 格式化时间元素数字
       1843 c 时间戳转datetime
                                   1.8.18.5. e.如果对象为None则转为默认值
                                   方法定义
```

```
def none(obj, v):
   如果对象为None则转为默认值
   :param obj: 待转对象
   :param v: 默认值
   :return:
```

#### 1.8.18.6. f.下一个值

#### 方法定义

```
def next_val(obj, sort=False, reverse=False, limit=10000, def_val=None, params=None):
    下一个值
   :param obj:
   :param sort:
   :param reverse:
   :param limit:
   :param def_val:
   :param params:
   :return:
```

示例

```
Table of Content
1. ivdata-common
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                   1.8.19. 19. operation utils
  1.3. 三、 jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
                                    8.19.1. a.追加/合并对象到list/dict
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                    法定义
      1.3.2.1. a. 类定义
    1.3.3. 3. 佳缘线上加解密
      1.3.3.1. a. 身份证号加密
                                    lef add(o1, o2):
      1.3.3.2. b. 身份证号解密
      1.3.3.3. c. 姓名加解密
                                       追加/合并对象到list/dict
    1.3.4. 4. 佳缘vip加解密
                                       :param o1:
      1.3.4.1. a. 佳缘vip数据加解密
                                       :param o2:
  1.4. 四、jy_db 数据库
                                       :return:
    1.4.1. 1. 获取数据库对象
      1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
    1.4.2. 2. 数据库对象api
      .... 2. メスルビティンダペロド
1.4.2.1. a. select: 音询, 适用于Oracle/8.20. 20. param_utils
         1.4.2.1.1. 方法定义
      1.4.2.2. b. update: 增删改, 适用于Orac 8.20.1. a.生成任务参数
         1.4.2.2.1. 方法定义
                                    法定义
         1.4.2.2.2. 示例
      1.4.2.3. c. update batch: 批量更新, 一
         1.4.2.3.1. 方法定义
                                    lef gen_task_params(app, **kwargs):
         1.4.2.3.2. 示例
  1.5. 五、jy_entity 实体基类
                                       生成任务参数
  1.6. 六、jy_factory 工厂类
                                       :param app:
  1.7. 七、jy_result 返回结果
                                       :param kwargs:
  1.8. 八、jy_util 工具/方法
                                       :return:
    1.8.1. 1. attr_utils: 对象属性工具方法库
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
    1.8.2.1. a. 获取给定日期所在月份的工作B.20.2. b. 处理接收参数 1.8.3. 3. condition_utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库
                                    法定义
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳线datetime
                                    def handle_receive_args(args):
                                       处理接收参数
                                       :param args:
                                   1.8.21. 21. str_utils
                                   1.8.21.1. a.打包字符串为dict
                                   方法定义
                                    def zip_str(s: str):
                                       将给定字符串打包成dict(), k, v分别为未知索引和对应的字符
                                       :param s:
                                       :return:
                                   示例
                                    from jy_utils import zip_str
                                    zip str('123')
                                    ### {0: '1', 1: '2', 2: '3'}
                                   1.8.21.2. b.字符串内部排序
                                   方法定义
                                    def sorted_str(s: str, idxes=None, reverse=False):
                                       字符串内部排序, 返回排序后的字符串
```

```
:param s:
Table of Content
                                       :param idxes: 给定的索引顺序
1. ivdata-common
                                       :param reverse: 是否反转
  1.1. —、jy_code
  1.2. 二、jy_conf 公共配置
                                       :return:
  1.3. 三、jy_crypt 加解密
    1.3.1. 1. base_bit_num 基础数位
      1.3.1.1. a. 类定义
                                    例
      1.3.1.2. b. 已定义对象
    1.3.2. 2. bit_num 进制数
                                    'rom jy_utils import sorted_str
      1.3.2.1. a. 类定义
                                    ;orted_str('1d3a2d45')
    1.3.3. 3. 佳缘线上加解密
                                    ## '12345add'
      1.3.3.1. a. 身份证号加密
      1.3.3.2. b. 身份证号解密
                                    sorted_str('1d3a2d45', reverse=True)
      1.3.3.3. c. 姓名加解密
                                    ## 'dda54321'
    1.3.4. 4. 佳缘vip加解密
                                    sorted_str('1d3a2d45', idxes='1042')
      1.3.4.1. a. 佳缘vip数据加解密
                                    ### 'd123'
  1.4. 四、jy_db 数据库
                                  sorted_str('1d3a2d45', idxes='1042', reverse=True)
    1.4.1. 1. 获取数据库对象
                                    ### '321d'
       1.4.1.1. a. 查看数据库配置信息
      1.4.1.2. b. 获取数据库对象
                                    8.21.3. c.有分隔符的字符串转换成dict
    1.4.2. 2. 数据库对象api
      1.4.2.1. a. select: 查询,适用于Oracle/I
         1.4.2.1.1. 方法定义
         1.4.2.1.2. 示例
       1.4.2.2. b. update: 增删改, 适用于Orac
                                    lef str2dict(s: str, sps: list, keys: list = None) -> dict:
         1.4.2.2.1. 方法定义
         1.4.2.2.2. 示例
       1.4.2.3. c. update batch: 批量更新, 一 把有分隔符的字符串转换成dict
         1.4.2.3.1. 方法定义
                                       :param s: 输入字符串
         1.4.2.3.2. 示例
                                       :param sps: 分隔符
  1.5. 五、jy_entity 实体基类
                                       :param keys: 给定key集合,在没有二级分隔符(sp)时,将一级分割结果依次匹配key
  1.6. 六、jy_factory 工厂类
                                       :return: 返回分割成的dict
  1.7. 七、jy_result 返回结果
  1.8. 八、jy_util 工具/方法
    8. 八、Jy_uui ユ桑ノハム
1.8.1. 1. attr_utils: 对象属性工具方法库
— 例
      1.8.1.1. a.设置对象属性
    1.8.2. 2. calender_utils: 日历工具方法库
      1.8.2.1. a.获取给定日期所在月份的工作日
    1.8.3. 3. condition_utils: 条件类工具方法库
    1.8.4. 4. date_utils: 日期时间类工具方法库
                                    8.21.4. d.字符串转值list
      1.8.4.1. a. 常量定义
      1.8.4.2. b. 格式化时间元素数字
      1843 c 时间戳转datetime
                                  万法定义
                                    def str2dict(s: str, sps: list, keys: list = None) -> dict:
                                       把有分隔符的字符串转换成dict
                                       :param s: 输入字符串
                                       :param sps: 分隔符
                                       :param keys: 给定key集合,在没有二级分隔符(sp)时,将一级分割结果依次匹配key
                                       :return: 返回分割成的dict
                                  示例
                                  1.8.21.5. e.根据给定字符集截取字符串
                                  方法定义
                                    def str2dict(s: str, sps: list, keys: list = None) -> dict:
                                       把有分隔符的字符串转换成dict
                                       :param s: 输入字符串
                                       :param sps: 分隔符
                                       :param keys: 给定key集合,在没有二级分隔符(sp)时,将一级分割结果依次匹配key
                                       :return: 返回分割成的dict
                                  示例
```