**一、redis实例分类**

目前的redis会有两种类型的实例设置：Cache实例和持久化实例。

**纯Cache 实例**： 根据LRU算法进行key 清理，无需设置报警提醒，保证可读即可。

**持久化实例：** 内存+磁盘持久化机制，无清理动作，需要设置阀值报警，人为通知相关业务线人员进行清理。

**二、redis分库**

为了避免各业务之间的相互干扰，每个redis实例内置的16个库（0~15库），根据4个业务线进行划分：

| **业务线** | **redis库** | **备注** |
| --- | --- | --- |
| 其他 | 0库 | 作为预留库，所有业务暂不使用 |
| 房产线 | 1~3库 |  |
| 安安租 | 4~6库 |  |
| 用户中心 | 7~9库 |  |
| 金融中心 | 10~12库 |  |
| 数据中心 | 13库 |  |
| 其他 | 14~15库 | 作为预留库，所有业务暂不使用 |

**三、房产线各个库使用**

| **redis库** | **存放内容** | **key命名规范** | **备注** |
| --- | --- | --- | --- |
| 1库 | 房源相关 | estate:代码仓库名称:key  (比如好房拓用户  estate:hft-service:userinfo) | 二手房、新房、租房 |
| 2库 | 人员、用户相关 | 好房拓 |
| 3库 | 内容相关 | 日志等公用服务 |
| 13库 | 数据中心 | dc:代码仓库名称:key  (比如dc:app-java-crm:housing) | 数据中心 |

**四、redis使用方式**

新框架：

**config/database.php**

|  |
| --- |
| 'redis' => [      'cluster' => false,      'default' => [          'host'     => env('DB\_REDIS\_DEFAULT\_HOST'),          'port'     => env('DB\_REDIS\_DEFAULT\_PORT'),          'database' => env('DB\_REDIS\_DEFAULT\_DATABASE'),      ],      'store' => [          'host'     => env('DB\_REDIS\_STORE\_HOST'),          'port'     => env('DB\_REDIS\_STORE\_PORT'),          'database' => env('DB\_REDIS\_STORE\_DATABASE'),      ],  ], |
| \Cache::setConnection('store');            // 选择连接  \Cache::put('key', 'value', $mMinutes);  // 使用缓存  \Cache::setConnection('default');        // 还原默认连接 |

老框架：

**config/redis.php**

|  |
| --- |
| $G\_CONFIG['redis']['xf\_store'] = array(          array(                  'host' => '10.59.72.31',                  'port' => 6379,          )  ); |
| load\_lib('/util/cache\_redis');  $oRedis = util\_cache\_redis::instance('xf\_store');  $oRedis->select(2);  $oRedis->setex('key', $iSeconds, 'value'); |

**五、注意事项**

* 由于其特殊性并不能保证数据的安全（比如：在达到maxmemory时，可能会出各种问题），所以把redis作为持久化储存时，建议同时把数据放在mysql作为备份（热备或者冷备）
* 所有key都必须设置有效时间