[Redis类型 1](#_Toc2774540)

[String类型 1](#_Toc2774541)

[Hash哈希类型 2](#_Toc2774542)

[List列表类型 2](#_Toc2774543)

[Set集合类型 2](#_Toc2774544)

[Zset有序集合类型 2](#_Toc2774545)

[Redis发布和订阅 2](#_Toc2774546)

[phpRedisAdmin可视化管理工具 3](#_Toc2774547)

[Php中使用redis 3](#_Toc2774548)

[Php安装redis扩展 3](#_Toc2774549)

[查看php中的redis类 3](#_Toc2774550)

[php使用redis 3](#_Toc2774551)

[把session数据写到php中 4](#_Toc2774552)

[Laravel中使用redis 4](#_Toc2774553)

[安装redis 4](#_Toc2774554)

[Laravel使用redis 4](#_Toc2774555)

[Redis当缓存机制 5](#_Toc2774556)

# Redis类型

## String类型

Clear 清空面板

Keys \* 列出所有键

Set [key] [value] 设置键值对

Get [key] 获取值

Getrange [key] [start] [end] 获取值start到end的字符

Mget [key] [key] ….. 获取多个值

Incr [key] 值加1

## Hash哈希类型

（命令以h为前缀，类似于php关联数组）

Hset [key] [field] [value] 设置哈希，类似于二维关联数组

Hkeys [key] 获取所有键

Hget [key] [field] 获取值

## List列表类型

（命令以l（left）和r（right）为前缀，类似于php索引数组）

Help lset 查看lset（或其他命令）命令帮助

Lpush [key] [value1] [value2] …… 初始化元素或从左插入一个或多个原始

Lrange [key] [start] [end] 列出从start到end的值

Lindex [key] [index] 列出所以为index的值

## Set集合类型

（集合是无序的，集合成员是唯一的，命令以s为前缀）

Sadd [key] [member1] [member2] …… 集合存储值，值重复不计

Smembers 列出集合值

Scard 列出集合值得数量

Sinter [key1] [key2] …… 求集合的交集

Sunion [key1] [key2] …… 求集合的并集

## Zset有序集合类型

（命令以z为前缀）

# Redis发布和订阅

（一个订阅号可以给多个用户发布消息）

Subscribe [channel] 订阅频道

Publish [channel] “message” 发布订阅消息

# phpRedisAdmin可视化管理工具

下载地址：<https://github.com/ErikDubbelboer/phpRedisAdmin>，注意vendor文件夹重新下载。

在web目录下直接访问下载文档即可

# Php中使用redis

## Php安装redis扩展

参考

<https://blog.csdn.net/q810391679/article/details/80812578>

<https://blog.csdn.net/TCF_JingFeng/article/details/80756617>

1. 本地127.0.0.1/phpinfo中查看PHP Extension Build一项，找到对应的值。在https://windows.php.net/downloads/pecl/snaps/redis/ 中下载redis插件。不成功换版本。
2. 将php\_redis.dll和php\_redis\_pdb文件并复制到php的ext文件夹中。
3. 将D:\Wamp\bin\apache\apache2.4.33\bin\php.ini和D:\Wamp\bin\php\php5.6.35\php.ini都添加如下内容：extension=php\_redis.dll。
4. 重启wamp，phpinfo查看有redis扩展即可。

## 查看php中的redis类

**<?php**$reflection = **new** ReflectionClass(**'Redis'**);  
$methods = $reflection->getMethods();  
*var\_dump*($methods);

## php使用redis

*//1、创建redis对象*$redis = **new** Redis();  
  
*//2、建立redis连接*$redis->connect(**'127.0.0.1'**,6379);  
  
*//3、操作redis*$redis->set(**'name'**,**'aaa'**);*//set命令即redis中的命令***echo** $redis->get(**'name'**);  
  
*//4、关闭redis*$redis->close();

## 把session数据写到php中

*//修改session驱动  
ini\_set*(**'session.save\_handler'**,**'redis'**);*//修改配置项，只对当前脚本有效。如果改php.ini文件，重启服务器即永久有效。  
ini\_set*(**'session.save\_path'**,**'tcp://127.0.0.1:6379'**);*//修改session驱动,session储存在当前文件中，在别的文件访问session，要做同样的驱动修改  
//开启session  
session\_start*();  
*//使用session*$\_SESSION[**'sess\_name'**] = **'bbb'**;  
$\_SESSION[**'sess\_age'**] = 19;  
*var\_dump*($\_SESSION);*//执行，phpRedisAdmin查看，多一个文件夹PJPREDIS\_SESSION*

# Laravel中使用redis

## 安装redis

使用composer安装redis: composer require predis/predis

## Laravel使用redis

使用redis门面操作redis数据库 use Illuminate\Support\Facades\Redis;

**<?php  
  
namespace** App\Http\Controllers;  
  
*//引用门面***use** Illuminate\Support\Facades\Cache;  
**use** Illuminate\Support\Facades\Redis;  
  
**class** IndexController **extends** Controller{  
 **public function** redis(){  
 *//redis作为缓存* Cache::**forever**(**'r\_name'**,**'ww'**);  
 *var\_dump*(Cache::*get*(**'r\_name'**));  
*// var\_dump(Redis::get('name'));  
// Redis::set('lname','mm');  
// Redis::set('lage',20);  
// var\_dump(Redis::get('name'));* }  
}

如果报错Class 'Predis\Client' not found，在项目中用composer加入predis即可，composer require predis/predis

## Redis当缓存机制

修改缓存配置，.evn文件中作以下修改：

CACHE\_DRIVER=file 改成 CACHE\_DRIVER=redis

缓存位置不一样，可能缓存在别的database，用phpRedisAdmin查看，文件名前缀为laravel\_cache