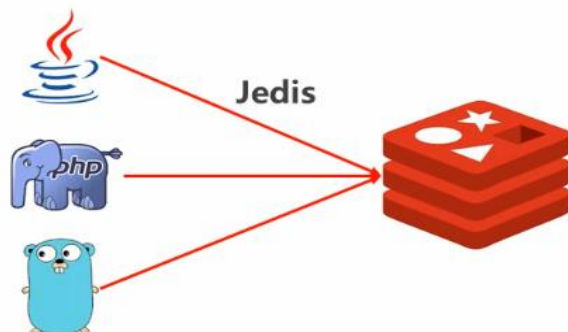


编程语言与redis

- Java语言连接redis服务
Jedis



编程语言与redis

- Java语言连接redis服务
Jedis
SpringData Redis
Lettuce
- C、C++、C#、Erlang、Lua、Objective-C、Perl、PHP、Python、Ruby、Scala

准备工作

- jar包导入
下载地址: <https://mvnrepository.com/artifact/redis.clients/jedis>
- 基于maven

```
<dependency>  
  <groupId>redis.clients</groupId>  
  <artifactId>jedis</artifactId>  
  <version>2.9.0</version>  
</dependency>
```

客户端连接redis

- 连接redis

```
Jedis jedis = new Jedis("localhost", 6379);
```

- 操作redis

```
jedis.set("name", "itheima");  
jedis.get("name");
```

- 关闭redis连接

```
jedis.close();
```

Jedis读写redis数据



案例：服务调用次数控制

人工智能领域的语义识别与自动对话将是未来服务业机器人应答呼叫体系中的重要技术，百度自研用户评价语义识别服务，免费开放给企业试用，同时训练百度自己的模型。现对试用用户的使用行为进行限速，限制每个用户每分钟最多发起10次调用

- 案例要求

- ① 设定A、B、C三个用户
- ② A用户限制10次/分调用，B用户限制30次/分调用，C用户不限制



案例：需求分析

- ① 设定一个服务方法，用于模拟实际业务调用的服务，内部采用打印模拟调用
- ② 在业务调用前服务调用控制单元，内部使用redis进行控制，参照之前的方案
- ③ 对调用超限使用异常进行控制，异常处理设定为打印提示信息
- ④ 主程序启动3个线程，分别表示3种不同用户的调用

案例：实现步骤

4. 设计启动主程序

```
public static void main(String[] args) {  
    MyThread mt1 = new MyThread("初级用户", 10);  
    MyThread mt2 = new MyThread("高级用户", 30);  
    mt1.start();  
    mt2.start();  
}
```

后续1：对业务控制方案进行改造，设定不同用户等级的判定

后续2：将不同用户等级对应的信息、限制次数等设定到redis中，使用hash保存

Jedis简易工具类开发

基于连接池获取连接

- JedisPool: Jedis提供的连接池技术
 - poolConfig: 连接池配置对象
 - host: redis服务地址
 - port: redis服务端口号

```
public JedisPool(GenericObjectPoolConfig poolConfig, String host, int port) {  
    this(poolConfig, host, port, 2000, (String)null, 0, (String)null);  
}
```