# 机器人后台操作文档

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 | 版本号 | 时间 | 类型 |
| 小六 | V1.0 | 2019-06-27 |  |

目录

[机器人后台操作文档 1](#_Toc12557545)

[一、 机器人管理 3](#_Toc12557546)

[**1.** **游戏记录** 3](#_Toc12557547)

[**2.** **机器人** 3](#_Toc12557548)

[**3.** **游戏管理** 4](#_Toc12557549)

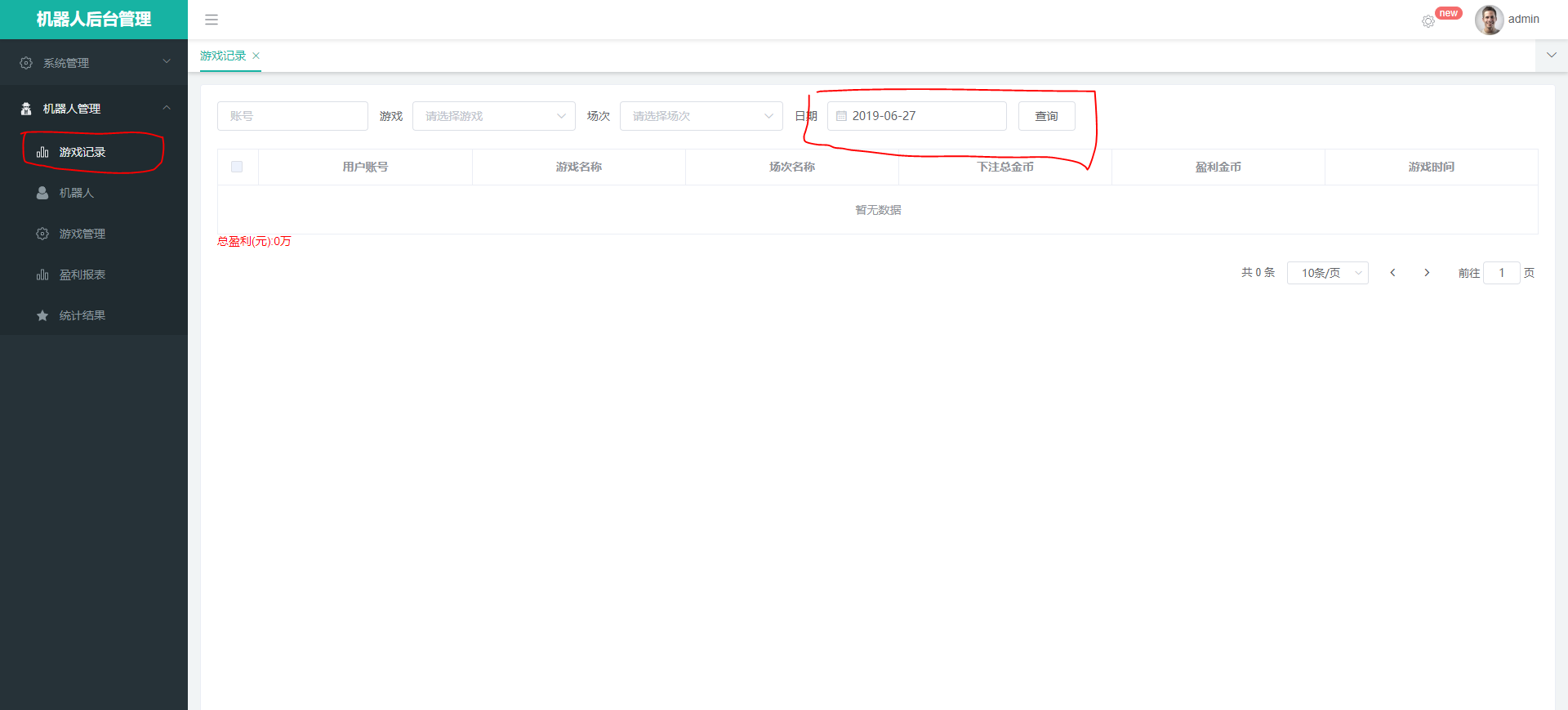
[**4.** **盈利报表** 5](#_Toc12557550)

[**5.** **统计结果** 5](#_Toc12557551)

# 机器人管理

1. **游戏记录**

**介绍:**游戏记录默认查看的是当天的，左下角的统计是当前查询条件下所有机器人游戏记录的总和,游戏记录只会保存今日和昨日的，在凌晨4点会把昨日以前的游戏记录全部删除。

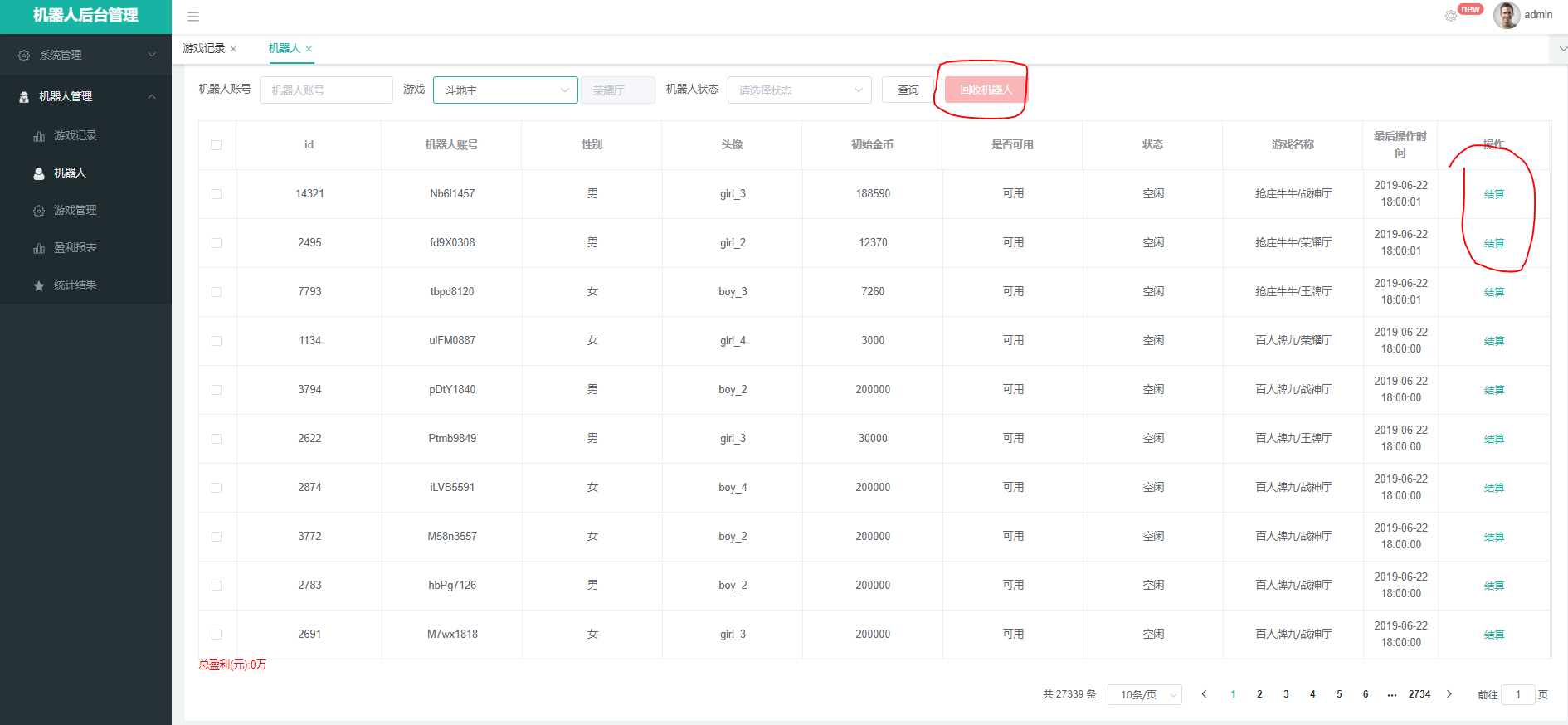


1. **机器人**

**介绍：**左下角总盈利是通过上面查询框的条件来变动的，实时的机器人盈利情况，统计的是robot中的profit\_coin字段

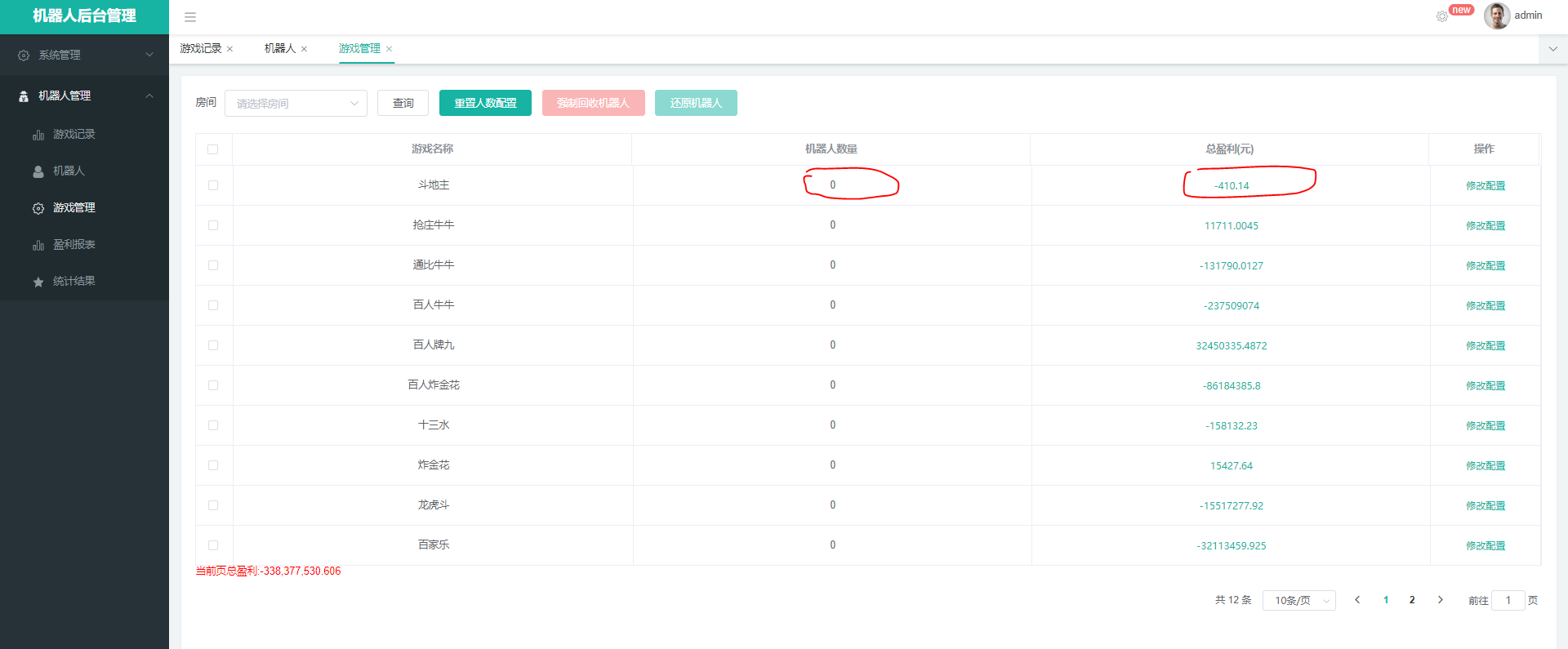
**其中的回收机器人和结算机器人主要是通过gameList接口中的offlineRobotUrl路径来实现的。**

1. **回收机器人：**回收机器人其实也是批量结算机器人，（在游戏中的机器人玩完当前的游戏就会通过消息队列返还，否则就是丢失了不在游戏中了，那么就通过后台逻辑回收）主要步骤是将机器人token传给机器人服务，机器人服务将找不到的机器人token返回，返回的token需要自己手动结算，如果没有返回token，那么这些回收的机器人就会在玩完这一把游戏通过MQ消息队列返还机器人。
2. **结算：**结算就是单个回收机器人接口



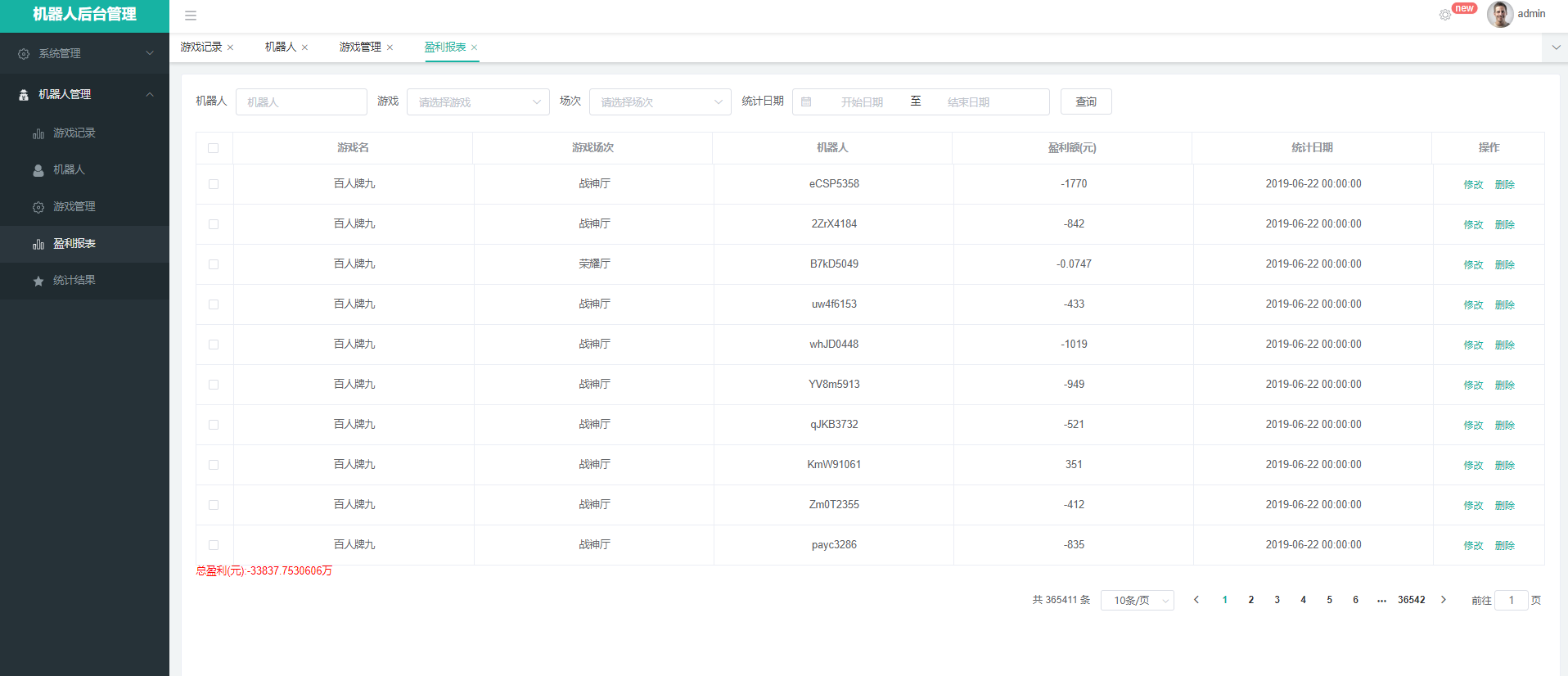
1. **游戏管理**

游戏中的机器人数量是通过gameList接口后，通过游戏ID分类，将每个场次的人数累加后得到的数据。总盈利是通过后台统计每个机器人的盈利报表来的数据，数据不包含今天的盈利（机器人盈利报表都是今天凌晨统计昨天）



1. **总盈利跳转：**点击总盈利跳转是传入游戏ID请求gameList，以此来获取每个场次的人数。
2. **重置人数配置：**调用gameList接口，获取到机器人服务中每款游戏的getrobotconfurl（获取配置）和putrobotconfurl（提交配置），通过getrobotconfurl路径获取到配置，将其中的机器人数量修改为0，并一个个通过putrobotconfurl提交。
3. **强制回收机器人：**根据游戏ID调用gameList获取所有场次的offlineroboturl路径，强行让选中的游戏所有场次所有机器人立刻离线
4. **还原机器人：**根据游戏ID调用gameList获取所有场次的forceonroboturl路径，强行让选中的游戏所有场次所有机器人立刻离线
5. **修改配置：**根据游戏ID调用gameList方法，再获取其中每个场次的getrobotconfurl，获取到配置，回显到页面中，当用户修改后，将参数再通过putrobotconfurl一个个提交。
6. **盈利报表**

盈利报表中的数据，是通过定时任务每天凌晨4点执行一次，统计昨日每个机器人盈利情况（游戏记录）



1. **统计结果**

统计结果，是每天凌晨4天统计无论成功或者失败的结果，防止统计不成功数据丢失（游戏记录每天都会删除两天前的），当有统计失败的情况，可以点击重新统计，重新统计就是触发凌晨的定时任务。

