

# Socket API - Game Part

## URL

<http://baseURL/game>

## 发送 API (emit)

玩家加入游戏（连接上 **socket** 后立即发送该事件）

事件名: join

参数: userID:int, gameId:int

客户端资源加载完毕，已准备好开始游戏

事件名: ready

参数: models:[string]

玩家移动

事件名: move

参数: position:[x:int, y:int, z:int]

玩家更换投票物品（或初次展示投票物品）

事件名: change vote model

参数: newModelName:string, oldModelName:string?

玩家投票

事件名: vote

参数: modelName:string

玩家取消投票

事件名: unvote

参数: modelName:string

玩家解锁（打开）了一道门

事件名: unlock

参数: door:int

一回合计时结束

事件名: round end

参数: models:[string], doors:[int]

游戏结束

事件名: game over  
参数: success:bool, score:int

## 接收 API (on)

所有玩家都已加入游戏，向各玩家返回该局游戏的基本信息

事件名: game info  
参数: users:[userID:int, initialRoom:[x:int, y:int]], mapLength:int, mapWidth:int, matrix:[[int]]

所有玩家都已准备好开始游戏

事件名: all are ready  
参数: hint:string

有玩家移动

事件名: moved  
参数: userID:int, position:[x:int, y:int, z:int]

有玩家更换投票物品（或初次展示投票物品）

事件名: vote model changed  
参数: userID:int, modelName:string

有玩家投票

事件名: voted  
参数: userID:int, modelName:string

有玩家取消投票

事件名: unvoted  
参数: userID:int, modelName:string

有一道门被解锁（打开）

事件名: unlocked  
参数: door:int

一回合正式结束，准备进入下一回合

事件名: next round  
参数: hint:string, success:bool, lockedDoor:int?, outUser:int?

游戏已结束，向各玩家返回排行榜信息

事件名: game is over  
参数: scores:[userID:int => score:int]

## 其他说明

- 类型后跟问号代表该字段为可选值