|  |
| --- |
| 请求方法都是post，数据格式都是json |
| 注册、登录 |
| 发送验证码：  url: /send\_check\_code  请求数据：  {  phone\_number: 电话号码  }  响应数据：  成功时：  {  status: “ok”  } |
| 注册：  url: /sign\_up  请求数据：  {  code: 验证码  name: 用户名  }  响应数据：  成功时：  {  status: “ok”,  public\_key: 生成的公钥  address: 生成的地址  private\_key: 生成的私钥  } |
| 登录：  url: /sign\_in  请求数据：  {  private: 用户的私钥  }  响应数据：  成功时：  {  status: “ok”  type: “user”或者“manager”  }  user表示普通用户登录，要跳转到普通用户界面；manager表示管理员登录，要跳转到管理员界面 |
| 用户 |
| 获取当前登录用户的信息：  url: /get\_account\_info  请求数据：无  响应数据：  成功时：  {  status: “ok”,  info: {  name: 用户名  balance: 余额  drawCount: 剩余抽奖次数  cardsId: 用户持有的所有的卡片的id  transactionsId: 用户参与的所有的交易的id  }  } |
| 卡片 |
| 根据卡片id获取卡片信息  url: /get\_card\_info  请求数据：  {  card\_id: 卡片的id  }  响应数据：  成功时：  {  status: “ok”,  info{  name: 卡片名字  level: 卡片等级  cardId: 卡片id  url: 卡片url  isOnSale: 是否正在出售（上架）  owner: 卡片当前的持有者  }  } |
| 根据卡片的id设置卡片价格  url: /set\_card\_price  请求数据：  {  card\_id: 卡片的id,  price: 价格  }  响应数据：  成功时：  {  status: “ok”  }  对于正在出售的卡片，可以改变出售的价格 |
| 交易 |
| 根据交易的id获取交易信息  url: /get\_transaction\_info  请求数据：  {  transaction\_id: 交易的id  }  响应数据：  成功时：  {  status: “ok”,  info: {  timestamp: 时间戳  cardId: 交易涉及的卡片的id  cardName: 交易涉及的卡片的名字  price: 交易涉及的卡片的价格  sellerAddress: 交易的卖方  buyerAddress: 交易的买方  transactionId: 交易的id  isReversing: 交易是否正在冲正中  isReversed: 交易是否已经冲正  reverseResult: 交易冲正的结果  }  }  在显示交易信息的时候，如果isReversing为true，显示交易正在冲正中；如果isReversed为true，此时，如果reverseResult为true，则显示冲正成功，如果为false则显示冲正失败；如果isReversing为false并且isReversed为false，则提供一个申请冲正的按钮，冲正时需要提供原因。 |
| 市场 |
| 获取市场中所有正在出售的卡片（暂时不考虑分页）：  url: /get\_cards\_on\_sale  请求数据：无  响应数据：  成功时：  {  status: “ok”,  cards\_on\_sale:[  {见获取卡片信息的info},  {见获取卡片信息的info},  ……  ]  } |
| 购买卡片：  url: /buy\_card  请求数据：  {  card\_id: 要购买的卡片的id  }  响应数据：  成功时：  {  status: “ok”  } |
| 下架（停止出售）卡片：  url: /pull\_card  请求数据：  {  card\_id: 要下架的卡片的id  }  响应数据：  成功时：  {  status: “ok”  } |
| 上架（出售）卡片  url: /push\_card  请求数据：  {  card\_id: 要上架的卡片  price: 价格  }  响应数据：  成功时：  {  status: “ok”  } |
| 抽卡（输入一段话抽卡（这段话会影响抽到的卡的等级））  url: /draw\_card  请求数据：  {  wish: “……”  }  响应数据：  成功时  {  status: “ok”,  card\_info: 抽到的卡片的信息（格式见上面的“获取卡片信息“）  } |
| 充值：  url: /recharge  请求数据：  {  amount: 充值的金额  }  响应数据：  成功时：  {  status: “ok”  } |
| 冲正（仲裁） |
| 用户申请冲正：  url: /apply\_for\_reverse  请求数据：  {  transaction\_id: 交易的id  reason: 理由  }  响应数据：  成功时：  {  status: “ok”  } |
| （管理员）获取所有冲正申请：  url: /get\_reverse\_applies  请求数据：无  响应数据：  成功时：  {  status: “ok”,  reverses: [  {reverse信息},  {reverse信息},  {reverse信息}  ……  ]  }  一个reverse信息的结构如下：  {  reverseId: 冲正的id  transactionId: 交易的id  role: 发起冲正申请的角色（买家或者卖家）  reason: 冲正的理由  dealed: 是否已经处理完毕（冲正结束）  }  在冲正申请信息附近提供一个按钮，使得管理员可以向被冲正的一方发送冲正消息  失败时：用户不是管理员  {  status: “error”,  error\_type: “not\_manager”  } |
| （被冲正的一方）获取（管理员发来的）冲正信息：  url: /get\_reverse\_info  请求数据：  {  reverse\_id: 冲正的id  }  响应数据：  成功时：  {  status: “ok”  info: {  reverseId: ……  transactionId: ……  role: ……  reason: ……  dealed: ……  }  } |
| （管理员向被冲正的一方）发送冲正消息：  url: /send\_reverse  请求数据：  {  reverse\_id: 冲正的id  }  响应数据：  成功时：  {  status: ok  }  失败时：用户不是管理员  {  status: “error”,  error\_type: “not\_manager”  } |
| （被冲正的一方在查看自己的冲正信息时）同意或拒绝冲正：  url: /set\_reverse\_result  请求数据：  {  reverse\_id: 冲正的id  result: 同意或者拒绝  }  响应数据：  成功时：  {  status: “ok”  } |