

ASDAN 商业竞赛项目 API 设计

Tomato 小组

2017 年 11 月 17 日

目录

1	一些修改及说明	3
2	Client API Endpoints	3
2.1	Get Information	3
2.2	Get Avatar	4
2.3	Get Property	5
2.4	Update Avatar	6
2.5	Get All User	7
2.6	交易：发出出售请求，buyer 监听	8
2.7	交易结束给卖家转发账单和现有资产	8
2.8	交易结束给队友转发账单和现有资产	10
2.9	监听比赛状态改变	11
2.10	监听 produce 后资产的变化	12
2.11	Get Trade History	13
2.12	Get Produce History	14
2.13	撤销，卖方监听	15
2.14	撤销，买方监听	16
3	Admin API Endpoints	17
3.1	Get All Competitions	17
3.2	Create Competition	18
3.3	Delete Competition By ID	20
3.4	Update Competition Status	21

3.5	Get Auction Machine	21
3.6	Record Auction Result	22
3.7	Get Competition Property	23
3.8	Update Competition Property	24
3.9	Get Competition Information	26
3.10	Record machine owner	28
4	Administrative Endpoints	30
4.1	Login Admin	31
4.2	Change Admin Password	31
4.3	Login Client	32
5	Utility Endpoints	33

1. Client API Endpoints: 为特定用户提供相关信息。
2. Admin API Endpoints: 为管理员提供相关信息。
3. Administrative Endpoints: 登录和交易时使用。
4. Utility Endpoints: 询问系统相关信息。

1 一些修改及说明

10.29: 修改了时间的表示方法, 目前按照 ISO-8601 格式, 具体请参照这篇文章: <https://stackoverflow.com/questions/19013562/java-dates-and-standard-formats>。

```
"yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
```

2 Client API Endpoints

为特定用户提供相关信息。

2.1 Get Information

输入 ID, 获得与这一 ID 相关的用户的配置信息。用户有默认头像。如果比赛还没有开始, 则 rank 为 0。

Request

```
GET /api/client/info/id={id}

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json
```

Returns

```
HTTP 200 OK

{
  "memberList":
  [
```

```

        "member1": "wangmz",
        "member2": "songsh",
        "member3": "wtf"
    ],
    "username": "team1",
    "id": 3,
    "rank": 1,

    "gameStatus": "auction" ("not_start", "auction_not_record", "auction_recorded", "trade", "rest", "end")
    "round": 0/1/2/3 (第 (round+1) 轮)
    "timeLeft": 300(s)
}

```

Error

HTTP 404 NOT FOUND

2.2 Get Avatar

获得用户头像

Request

GET /api/client/info/getAvatar/id={id}

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

Returns

HTTP 200 OK

HashMap

```

{
    "fileoriginalsize", size;
    "contenttype", contentType;
    "base64", new String(Base64Utils.encode(IOUtils.toByteArray(stream)));
}

```

Error

```
HTTP 404 NOT FOUND
```

2.3 Get Property

输入 ID，获得与这一 ID 相关的用户的财产信息。包括机器的使用情况和材料的价格。

Request

```
GET /api/client/property/id={id}

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json
```

Returns

```
HTTP 200 OK

{
  "wealth": 3000,
  "machine":
  [
    {
      "id": 0073,
      "type": "Cement",
      "left": 3
      "lock":false
    },
    {
      "id": 0793,
      "type": "Brick",
      "left": 0
      "lock":true(正处于出售中的机器和材料 lock == true)
    },
    {
      "id": 8765,
      "type": "Wood",
      "left": 2
      "lock":false
    }
  ],
}
```

```
"material":  
[  
  {  
    "type": "Wood",  
    "price": 10,  
    "number": 20,  
    "lock":true  
  },  
  {  
    "type": "Brick",  
    "price": 20,  
    "number": 0,  
    "lock":false  
  },  
  {  
    "type": "Cement",  
    "price": 80,  
    "number": 150,  
    "lock":false  
  }  
]  
}
```

Error

HTTP 404 NOT FOUND

2.4 Update Avatar

向服务器发送用户更改过的头像。(其它队友获取新头像直接调用 get Avatar)

Request

POST /api/client/info/updateAvatar/id={id}

Host: localhost:8080

MultipartHttpServletRequest request

Returns

HTTP 200 OK

HashMap

```
{
    "fileoriginalsize", size;
    "contenttype", contentType;
    "base64", new String(Base64Utils.encode(IOUTils.toByteArray(stream)));
}
```

Error

HTTP 404 NOT FOUND

2.5 Get All User

get 所有队伍，(除了发送消息的队伍)，用来发 sell Request 时进行选择

Request

GET /api/client/getAllUser/id={id}

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

Returns

HTTP 200 OK

```
{
  [
    {
      "teamId": 889,
      "username": "dddd",
    },
    {
      "teamId": 999,
      "username": "ddfadf",
    },
  ]
}
```

Error

HTTP 404 NOT FOUND

2.6 交易：发出出售请求，buyer 监听

```
MessageMapping: /api/client/property/sellerId={sellerId}/buyerId={buyerId}

SendTo: /api/client/property/buyerId={buyerId}

EndPoint: http://127.0.0.1:8090/trade

Get JSon Pattern: (卖方发送)
{
  "tradeId": '',
  "sellerId": sellerId,
  "buyerId": buyerId,
  "buyer": "TOMATO"
  "typeOrMachineID": "9987" ("Wood" "Cement" "Brick" OR machineID)
  "price": 300,
  "number": 7
  "seller": "Rua"
}
Send JSon Pattern: (发给买方)
{
  "tradeId": tradeId,
  "sellerId": sellerId,
  "buyerId": buyerId,
  "buyer": "TOMATO"
  "typeOrMachineID": "9987" ("Wood" "Cement" "Brick" OR machineID)
  "price": 300,
  "number": 7
  "seller": "Rua"
}
```

2.7 交易结束给卖家转发账单和现有资产

```
MessageMapping: /api/client/tradeFinish/sellerId={sellerId}

SendTo: /api/client/tradeFinish/id={sellerId}

EndPoint: http://127.0.0.1:8090/tradeFinish

Get JSon Pattern:
```



```

{
  "tradeId":tradeId,
  "sellerId":sellerId,
  "buyerId":buyerId,
  "buyer":"TOMATO"
  "typeOrMachineID":"9987" ("Wood" "Cement" "Brick" OR machineID)
  "price":300,
  "number":7
  "seller":"Rua"

  "isAccept":true
}

```

Send JSon Pattern:

```

{
  "reply":
  {
    "tradeId":tradeId,
    "sellerId":sellerId,
    "buyerId":buyerId,
    "buyer":"TOMATO"
    "typeOrMachineID":"9987" ("Wood" "Cement" "Brick" OR machineID)
    "price":300,
    "number":7
    "seller":"Rua"

    "isAccept":true
  }

  "propertyList":
  {
    "wealth":1000,
    "machineList":
    [
      {
        "id": 0073,
        "type": "Wood",
        "left": 3
        "lock":false
      },
    ],
    "materialList":
    [
      {
        "type": "Brick",
        "price": 20,

```

```

        "number": 0,
        "lock":false
    },
    ]
}
}

```

2.8 交易结束给队友转发账单和现有资产

```

MessageMapping: /api/client/tradeFinish/buyerId={buyerId}

SendTo: /api/client/tradeFinish/id={buyerId}

EndPoint: http://127.0.0.1:8090/tradeFinish

Get JSon Pattern:
{
    "tradeId":tradeId,
    "sellerId":sellerId,
    "buyerId":buyerId,
    "buyer":"TOMATO"
    "typeOrMachineID":"9987" ("Wood" "Cement" "Brick" OR machineID)
    "price":300,
    "number":7
    "seller":"Rua"

    "isAccept":true
}

Send JSon Pattern:
{
    "reply":
    {
        "tradeId":tradeId,
        "sellerId":sellerId,
        "buyerId":buyerId,
        "buyer":"TOMATO"
        "typeOrMachineID":"9987" ("Wood" "Cement" "Brick" OR machineID)
        "price":300,
        "number":7
        "seller":"Rua"

        "isAccept":true
    }
}

```

```

"propertyList":
{
  "wealth":1000,
  "machineList":
  [
    {
      "id": 0073,
      "type": "Wood",
      "left": 3
      "lock":false
    },
  ],
  "materialList":
  [
    {
      "type": "Brick",
      "price": 20,
      "number": 0,
      "lock":false
    },
  ]
}
}

```

2.9 监听比赛状态改变

```

MessageMapping: /api/admin/competition/status/id={id}

SendTo: /api/client/CompetitionStatus/id={id}

EndPoint: http://127.0.0.1:8090/competitionStatus

Send JSon Pattern:
{
  "gameStatus": "auction" ("not_start", "auction_not_record", "auction_recorded", "trade", "rest", "
    end")
  "round": 0/1/2/3 (第 (round+1) 轮)
  "timeLeft":227(s)
}

Get JSON Pattern:
{
  "gameStatus": "auction" ("not_start", "auction_not_record", "auction_recorded", "trade", "rest",
  "round": 0/1/2/3 (从0开始)
}

```

2.10 监听 produce 后资产的改变

MessageMapping: /api/client/ListenProperty/id=3

SendTo: /api/client/ListenProperty/receive/id=3

EndPoint: <http://127.0.0.1:8090/listenProperty>

Send JSon Pattern:

```
{
  "wealth":1000,
  "machine":
  [
    {
      "id": 0073,
      "type": "Wood",
      "left": 3
      "lock":false
    },
    {
      "id": 0793,
      "type": "Brick",
      "left": 0
      "lock":true
    },
    {
      "id": 8765,
      "type": "Cement",
      "left": 2
      "lock":false
    }
  ],
  "material":
  [
    {
      "type": "Wood",
      "price": 10,
      "number": 20,
      "lock":false
    },
    {
      "type": "Brick",
      "price": 20,
      "number": 0,
      "lock":false
    },
    {
```

```
        "type": "Cement",
        "price": 80,
        "number": 150,
        "lock": false
    }
]
}
```

Get JSON Pattern:

```
{
  "id": 2,
  "times": 1,
}
```

2.11 Get Trade History

交易历史信息。在发订单的时候客户端手动更新 History。

Request

```
GET /api/client/tradeHistory/id={id}
```

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

Returns

HTTP 200 OK

```
[
  {
    "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
    "target": "team1",
    "action": "sell",
    "content": "wood",
    "price": 10,
    "number": 2
    "status": 1, (完成)
    "tradeId": 44,
    "buyerId": 9
  },
]
```

```
{
  "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
  "target": "team1",
  "action": "buy",
  "content": "1234" (machine.id ==1234)
  "price": 10,
  "number": 1 (只能是1)
  "status": 0, (正在进行)
  "tradeId":44,
  "buyerId":9
},

{
  "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
  "target": "team1",
  "action": "buy",
  "content": "6666" (machine.id ==1234)
  "price": 10,
  "number": 1 (只能是1)
  "status": -1, (失败)
  "tradeId":44,
  "buyerId":9,
}
]
```

Error

HTTP 404 NOT FOUND

2.12 Get Produce History

生产历史信息。

Request

GET /api/client/produceHistory/id={id}

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

Returns

HTTP 200 OK

```
[
  {
    "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
    "machineId": 9987,
    "content": "Brick",
    "price": 10,
    "number": 2
  },
  {
    "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
    "machineId": 3457,
    "content": "Wood",
    "price": 10,
    "number": 2
  },
  {
    "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
    "machineId": 5777,
    "content": "Cement",
    "price": 10,
    "number": 2
  },
]
```

Error

HTTP 404 NOT FOUND

2.13 撤销，卖方监听

MessageMapping: /api/client/undo/sendToSeller/sellerId={sellerId}/buyerId={buyerId}

SendTo: /api/client/receiveUndo/id={sellerId}

EndPoint: <http://127.0.0.1:8090/undo>

Get JSon Pattern: (卖方发送)

```
{
  "tradeId": 77,
}
```

Send JSon Pattern: (发给卖方)

```
{
```

```

"request":
{
  "tradeId":77,
  "sellerId":sellerId,
  "buyerId":buyerId,
  "buyer":"TOMATO"
  "typeOrMachineID":"9987" ("Wood" "Cement" "Brick" OR machineID)
  "price":300,
  "number":7
  "seller":"Rua"
}

"propertyList":
{
  "wealth":1000,
  "machineList":
  [
    {
      "id": 0073,
      "type": "Wood",
      "left": 3
      "lock":false
    },
  ],
  "materialList":
  [
    {
      "type": "Brick",
      "price": 20,
      "number": 0,
      "lock":false
    },
  ]
}
}

```

2.14 撤销，买方监听

```

MessageMapping: /api/client/undo/sendToBuyer/sellerId={sellerId}/buyerId={buyerId}

SendTo: /api/client/receiveUndo/id={buyerId}

EndPoint: http://127.0.0.1:8090/undo

Get JSon Pattern: (卖方发送)
{

```



```

    "tradeId": 77,
  }
}
Send JSon Pattern: (发给买方)
{
  "request":
  {
    "tradeId":77,
    "sellerId":sellerId,
    "buyerId":buyerId,
    "buyer":"TOMATO"
    "typeOrMachineID":"9987" ("Wood" "Cement" "Brick" OR machineID)
    "price":300,
    "number":7
    "seller":"Rua"
  }

  "propertyList":
  {
    "wealth":1000,
    "machineList":
    [
      {
        "id": 0073,
        "type": "Wood",
        "left": 3
        "lock":false
      },
    ],
    "materialList":
    [
      {
        "type": "Brick",
        "price": 20,
        "number": 0,
        "lock":false
      },
    ]
  }
}

```

3 Admin API Endpoints

3.1 Get All Competitions

列出全部比赛。

Status 是"not_start", "auction_not_record", "auction_recorded", "trade", "rest", "end" 之一。

Request

```
GET /api/admin/competition/getall

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json
```

Returns

```
HTTP 200 OK

[
  {
    "id": "competiton1_id",
    "username": "competition1",
    "status": "auction"
  },
  {
    "id": "competiton2_id",
    "username": "competition2",
    "status": "end"
  }
]
```

Error

```
HTTP 204 NO CONTENT
```

3.2 Create Competition

新建一场比赛。注意，底层也要生成机器的 id。注意每场比赛的基本配置（比赛名称，参赛人数）只能创建一次，不能修改。

Request

POST /api/admin/competition/new

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

```
{
  "username": "competition_username",
  "round": 2,
  "startWealth": 1000,

  "teamNum": 2,
  "participantNum": 3,
  "team":
  [
    {
      "username": "team1",
      "participant": ["mem11", "mem12", "mem13"],
      "password": "111111",
    },
    {
      "username": "team2",
      "participant": ["mem21", "mem22", "mem23"],
      "password": "222222",
    }
  ]

  "roundParameter":
  [
    {
      "machineStartPrice": [300, 350, 400],
      "machineNum": [1, 1, 1],
      "materialProduceCost": [10, 20, 30],
      "time": 900,
    },
    {
      "machineStartPrice": [300, 350, 400],
      "machineNum": [1, 1, 1],
      "materialProduceCost": [10, 20, 30],
      "time": 900,
    }
  ]
}
```

Returns

```
HTTP 201 CREATED
```

Error

```
HTTP 404 NOT FOUND
```

```
{
  "error": "Unable to delete. Competition with id xxx not found."
}
```

3.3 Delete Competition By ID

通过 ID 删除比赛。

Request

```
DELETE /api/admin/competiton/id={competition_id}
```

```
Host: localhost:8080
```

```
Auth:
```

```
Content-type: application/json
```

```
Accept: application/json
```

Returns

```
HTTP 200 OK
```

```
[
  {
    "id": "competiton2_id",
    "username": "competition2",
    "status": "end"
  }
]
```

Error

```
{
  "error": "Unable to delete. Competition with id xxx not found."
}
```

3.4 Update Competition Status

需要进入下一环节时，管理员端会向服务器发送更新比赛状态请求，服务器返回当前比赛信息以便管理员端更新到最新的比赛状态。

Web Socket

```
MessageMapping: /api/admin/status/update/id=3

SendTo: /api/admin/status/id=3

EndPoint: http://127.0.0.1:8090/competitionStatus

Send JSON Pattern:
{
  "status": "auction" ("not_start", "auction_not_record", "auction_recorded", "trade", "rest", "end")
  "round": 0/1/2/3
  "timeLeft":227(s)
}

Get JSON Pattern:
{
  "status": "auction" ("not_start", "auction_not_record", "auction_recorded", "trade", "rest", "end")
  "round": 0/1/2/3
}
```

Error

```
HTTP 404 NOT FOUND
{
  "error":"Unable to update. Competition with id xxx not found"
}
```

3.5 Get Auction Machine

获得某场比赛某一轮拍卖机器的初始信息。

Request

```
GET /api/admin/competition/auction/id={id}/round={round}

Host: localhost:8080
Auth:
```

```
Content-type: application/json
Accept: application/json
```

Returns

HTTP 200 OK

```
[
  {
    "machineId": "machine1",
    "type": "Wood",
    "startPrice": 200,
  },
  {
    "machineId": "machine2",
    "type": "Brick",
    "startPrice": 300,
  },
  {
    "machineId": "machine3",
    "type": "Cement",
    "startPrice": 400,
  }
]
```

Error

HTTP 404 NOT FOUND

```
{
  "error": "Competition with id xxx not found." (or Competition with id xxx does not have round xxx)
}
```

3.6 Record Auction Result

登记某场比赛某一轮的拍卖结果。

Request

POST /api/admin/competition/record/id={id}/round={round}

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

Returns

```
HTTP 200 OK
[
  {
    "machineId": "machine1",
    "teamId": "team1",
    "price": 2000,
  },
  {
    "machineId": "machine2",
    "teamId": "brick",
    "price": 3000,
  },
  {
    "machineId": "machine3",
    "teamId": "cement",
    "price": 4000,
  }
]
```

Error

```
HTTP 404 NOT FOUND
{
  "error": "Competition with id xxx not found." (or Competition with id xxx does not have round xxx)
}
```

3.7 Get Competition Property

从服务器按 id 获取某一比赛的各种属性。如果该比赛的属性尚未被设置，则该项为空。属性包括名称、比赛轮数（如果比赛已开始，则不能删除已开始或结束的轮）、比赛各项参数（不能修改已开始或结束的轮的参数）、机器的 id 等等。

Request

```
GET /api/admin/competition/property/id={id}

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json
```

Returns

```
HTTP 200 OK
{
  "id": "competition_id",
  "username": "competition_username",
  "status": "not started",
  "teamNum": 1,
  "participantNum": 2,
  "team":
  [
    {
      "username": "team1",
      "participant": ["member1", "member2", "member2"],
      "password": "password",
    }
  ]
  "round": 1,
  "startWealth": 1000,
  "roundParameter":
  [
    {
      "machineStartPrice": [300, 350, 400],
      "machineNum": [1, 1, 1],
      "materialProduceCost": [10, 20, 30],
      "time": 900,
    }
  ]
}
```

Error

```
HTTP 404 NOT FOUND
{
  "error": "Competition with id xxx not found."
}
```

3.8 Update Competition Property

更新比赛的各种属性。属性包括名称、比赛轮数（如果比赛已开始，则不能更改）、比赛各项参数（不能修改已开始或结束的轮的参数）。

Request


```
PUT /api/admin/competition/property/id={id}
```

```
Host: localhost:8080
```

```
Auth:
```

```
Content-type: application/json
```

```
Accept: application/json
```

```
{
  "round": 2,
  "startWealth": 1000,
  "round_parameter":
  [
    {
      "machineStartPrice": [300, 350, 400],
      "machineNum": [1, 1, 1],
      "materialProduceCost": [10, 20, 30],
      "time": 900,
    },
    {
      "machineStartPrice": [300, 350, 400],
      "machineNum": [1, 1, 1],
      "materialProduceCost": [10, 20, 30],
      "time": 900,
    }
  ]
}
```

Returns

```
HTTP 201 CREATED
```

Error

```
HTTP 404 NOT FOUND
```

```
{
  "error": "Competition with id xxx not found."
}
```

```
HTTP 400 INVALID REQUEST
```

```
{
  "error": "Cannot update competition id xxx with given changes."
}
```

3.9 Get Competition Information

获取当前比赛信息，包括队伍的数量、资产、交易记录、机器的使用情况等。

Request

```
GET /api/admin/competition/info/id={id}

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json
```

Returns

```
HTTP 200 OK
{
  "id": "competition_id",
  "username": "competition_username",
  "status": "not_start",
  "round": 2,
  "presentRound": 0,
  "teamInfo":
  [
    {
      "id": "id1",
      "wealth": 100,
      "material": [30, 40, 50],
      "machine":
      [
        {
          "id": "machine1_id",
          "type": "type1",
          "left": 3
        },
        {
          "id": "machine2_id",
          "type": "type2",
          "left": 2
        }
      ]
    },
    {
      "id": "id2",
      "wealth": 100,
```

```

    "material": [30, 40, 50],
    "machine":
    [
        {
            "id": "machine1_id",
            "type": "type1",
            "left": 3
        },
        {
            "id": "machine2_id",
            "type": "type2",
            "left": 2
        }
    ]
},
],
"trade_history":
[
    {
        "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
        "sell": "team_id1",
        "buy": "team_id2",
        "content": {"wood": 1},
        "price": 10
    },
    {
        "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
        "sell": "team_id1",
        "buy": "team_id2",
        "content": {"machine_wood": 1},
        "price": 20
    }
]
}

```

content 中是 wood, brick, cement, $machine_{wood}$ $machine_{brick}$ $machine_{cement}$

Error

```

HTTP 404 NOT FOUND
{
    "error": "Competition with id 1 not found."
}

```

3.10 Record machine owner

向服务器发送对比赛的更新信息。增加机器、分配财产之类的。

Request

```
PUT /api/admin/competition/info/id={id}
```

```
Host: localhost:8080
```

```
Auth:
```

```
Content-type: application/json
```

```
Accept: application/json
```

```
{
  "round": 2,
  "present_round": 0,
  "team_info":
  [
    {
      "id": "id1",
      "wealth": "100",
      "machine":
      [
        {
          "id": "machine1_id",
          "left": 3
        },
        {
          "id": "machine2_id",
          "left": 2
        }
      ]
    },
    {
      "id": "id2",
      "wealth": "100",
      "machine":
      [
        {
          "id": "machine1_id",
          "left": 3
        },
        {
          "id": "machine2_id",
          "left": 2
        }
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

Returns

HTTP 200 OK

```
{  
  "id": "competition_id",  
  "username": "competition_username",  
  "status": "not started",  
  "round": "2",  
  "present_round": "0",  
  "team_info":  
  [  
    {  
      "id": "id1",  
      "wealth": 100,  
      "material": [30, 40, 50],  
      "machine":  
      [  
        {  
          "id": "machine1_id",  
          "type": "type1",  
          "left": 3  
        },  
        {  
          "id": "machine2_id",  
          "type": "type2",  
          "left": 2  
        }  
      ]  
    },  
    {  
      "id": "id2",  
      "wealth": 100,  
      "material": [30, 40, 50],  
      "machine":  
      [  
        {  
          "id": "machine1_id",  
          "type": "type1",  
          "left": 3  
        },  
        {  
          "id": "machine2_id",  
          "type": "type2",  
          "left": 2  
        }  
      ]  
    }  
  ]  
}
```

```

        "type": "type2",
        "left": 2
    }
]
},
],
"trade_history":
[
    {
        "time": "hh:MM:ss",
        "sell": "team_id1",
        "buy": "team_id2",
        "content": {"wood": 1},
        "price": 10
    },
    {
        "time": "hh:MM:ss",
        "sell": "team_id1",
        "buy": "team_id2",
        "content": {"machine_wood": 1},
        "price": 20
    }
]
}

```

content 中是 wood, brick, cement, machine_wood machine_brick machine_cement

Error

HTTP 404 NOT FOUND

```

{
    "error": "Competition with id xxx not found."
}

```

HTTP 400 INVALID REQUEST

```

{
    "error": "Cannot update competition id xxx with given information."
}

```

4 Administrative Endpoints

登录和交易时使用。

4.1 Login Admin

管理员登录。

Request

```
POST /api/admin/login

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json

{
  "username": "admin",
  "password": "admin",
}
```

Returns

```
HTTP 200 OK

{
  "username": "admin",
  "token": "1283091828021803120",
}
```

Error

```
HTTP 401 NOTAUTHORIZED

{
  "error": "Admin with username admin doesn't exist or password is wrong."
}
```

4.2 Change Admin Password

更改管理员密码。

Request

```
POST /api/admin/update

Host: localhost:8080
```

```
Auth:
Content-type: application/json
Accept: application/json

{
  "username": "admin",
  "prePassword": "admin",
  "newPassword": "newpwd",
}
```

Returns

HTTP 200 OK

Error

```
HTTP 401 NOTAUTHORIZED

{
  "error": "Wrong password of admin."
}
```

4.3 Login Client

用户登录。

Request

```
POST /api/client/login

Host: localhost:8080
Auth:
Content-type: application/json
Accept: application/json

{
  "username": "client",
  "password": "client",
}
```


Returns

HTTP 200 OK

```
{
  "username": "client",
  "id": "3",
  "token": "1283091828021803120",
}
```

Error

HTTP 401 NOTAUTHORIZED

```
{
  "error": "Client with userusername admin doesn't exist or password is wrong."
}
```

5 Utility Endpoints

询问系统相关信息。