

# ASDAN 商业竞赛项目 API 设计

Tomato 小组

2017 年 11 月 2 日

## Contents

<b>1</b>	<b>一些修改及说明</b>	<b>2</b>
<b>2</b>	<b>Client API Endpoints</b>	<b>2</b>
2.1	Get Information . . . . .	2
2.2	Get Property . . . . .	3
2.3	Update Information . . . . .	4
2.4	Produce . . . . .	5
2.5	Sell Material . . . . .	6
2.6	Sell Machine . . . . .	7
2.7	Get History . . . . .	8
<b>3</b>	<b>Admin API Endpoints</b>	<b>9</b>
3.1	Get All Competitions . . . . .	9
3.2	Create Competition . . . . .	10
3.3	Delete Competition By ID . . . . .	11
3.4	Update Competition Status . . . . .	11
3.5	Get Auction Machine . . . . .	13
3.6	Record Auction Result . . . . .	14
3.7	Get Competition Property . . . . .	15
3.8	Update Competition Property . . . . .	16
3.9	Get Competition Information . . . . .	17
3.10	Record machine owner . . . . .	19

<b>4</b>	<b>Administrative Endpoints</b>	<b>22</b>
4.1	Login Admin . . . . .	22
4.2	Login Client . . . . .	23
<b>5</b>	<b>Utility Endpoints</b>	<b>24</b>

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。

TO-DO: 用户自己设置头像。TO-DO: 三个用户同时登录。

#### Request

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

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

#### Returns

```
HTTP 200 OK

{
  "username": "team1",
  "id": "3",
  "avatar": "/9j/4QAYRXhpZg ...",
```

```
"gameStatus": "第轮2"  
"rank": 1  
}
```

## Error

HTTP 404 NOT FOUND

## 2.2 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": "type1",  
        "left": 3  
      },  
      {  
        "id": "0793",  
        "type": "type1",  
        "left": 0  
      },  
      {  
        "id": "8765",  
        "type": "type2",  
        "left": 2  
      }  
    ],  
}
```

```
"material":  
[  
  {  
    "type": "wood",  
    "price": 10,  
    "number": 20,  
  },  
  {  
    "type": "brick",  
    "price": 20,  
    "number": 0,  
  },  
  {  
    "type": "cement",  
    "price": 80,  
    "number": 150,  
  }  
]  
}
```

## Error

HTTP 404 NOT FOUND

## 2.3 Update Information

向服务器发送更改过的用户配置信息。

### Request

PUT /api/client/info/id={id}

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

```
{  
  "username": "team1",  
  "id": 3,  
  "avatar": ".....",  
  "rank": 1,  
}
```

subsubsection\*Returns

HTTP 200 OK

```
{
  "username": "team1",
  "id": 3,
  "avatar": "/9j/4QAYRXhpZg ...",
  "rank": 1
}
```

## Error

HTTP 404 NOT FOUND

## 2.4 Produce

向服务器发送需要生产啊的机器的 ID 和生产次数

### Request

POST /api/client/property/produce/id={id}

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

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

### Returns

HTTP 200 OK

```
{
  "wealth": 1000,
  "machine":
  [
    {
      "id": "0073",
      "type": "type1",
      "left": 3
    },
    {

```

```
    "id": "0793",
    "type": "type1",
    "left": 0
  },
  {
    "id": "8765",
    "type": "type2",
    "left": 2
  }
],
"material":
[
  {
    "type": "wood",
    "price": 10,
    "number": 20,
  },
  {
    "type": "brick",
    "price": 20,
    "number": 0,
  },
  {
    "type": "cement",
    "price": 80,
    "number": 150,
  }
]
}
```

## Error

HTTP 404 NOT FOUND

## 2.5 Sell Material

向服务器发送出售的 Material 的种类，数量，和单位数量的价格，以及接受的人的 ID

### Request

POST /api/client/property/sellMaterial/id={id}

Host: localhost:8080

Auth:

Content-type: application/json

```
Accept: application/json
```

```
{
  "type": "wood"
  "number": 2
  "price": 300,
  "recieverID": 7
}
```

### Returns

```
HTTP 200 OK
```

### Error

```
HTTP 404 NOT FOUND
```

## 2.6 Sell Machine

向服务器发送: 出售的机器的 id, 售出的价格, 和售出的一台机器的价格, 以及接受机器的人的 ID

### Request

```
POST /api/client/property/sellMachine/id={id}
```

```
Host: localhost:8080
```

```
Auth:
```

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

```
Accept: application/json
```

```
{
  "id": 9870
  "price": 1000,
  "recieverID": 7
}
```

### Returns

```
HTTP 200 OK
```

### Error

```
HTTP 404 NOT FOUND
```



## 2.7 Get History

获取与某一用户相关的历史信息。在发订单的时候客户端手动更新 History。

### Request

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

```
Host: localhost:8080
```

```
Auth:
```

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

```
Accept: application/json
```

### Returns

```
HTTP 200 OK
```

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

  {
    "time": "yyy-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）
  },

  {
    "time": "yyy-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）
  }
]
```

```
}  
]
```

## Error

HTTP 404 NOT FOUND

## 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,
  "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的请求，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。

#### Request

```
POST /api/admin/competition/status/id={id}
```

Host: localhost:8080

Auth:

Content-type: application/json

Accept: application/json

```
{
```

```
"status": "auction"
"round": 1 (-1 if round field is not necessary)
}
```

## Returns

HTTP 200 OK

```
{
  "id": "competition_id",
  "username": "competition_username",
  "status": "auction",
  "round": 2,
  "presentRound": 1,
  "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
        }
      ]
    }
  ]
}
```

```

        {
            "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": {"material1": 1},
        "price": 10
    },
    {
        "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
        "sell": "team_id1",
        "buy": "team_id2",
        "content": {"material2": 1},
        "price": 20
    }
]
}

```

如果是生产交易环节还需要添加

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

## 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": {"material1": 1},
    "price": 10
  },
  {
    "time": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'",
    "sell": "team_id1",
    "buy": "team_id2",
    "content": {"material2": 1},
    "price": 20
  }
]
}

```

## 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"
        }
      ]
    }
  ]
},
],

```

```
"trade_history":
[
  {
    "time": "hh:MM:ss",
    "sell": "team_id1",
    "buy": "team_id2",
    "content": {"material1": 1},
    "price": "10"
  },
  {
    "time": "hh:MM:ss",
    "sell": "team_id1",
    "buy": "team_id2",
    "content": {"material2": 1},
    "price": "20"
  }
]
```

## 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 Login Client

用户登录。

### Request

```
POST /api/client/login

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

{
  "username": "client",
  "id": "3",
  "password": "client",
}
```

### Returns



HTTP 200 OK

```
{  
  "username": "client",  
  "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。