

# Тестовое задание

Решение просьба выложить на GitHub (или на аналогичный сервис). В readme.md должна быть краткая инструкция по запуску.

Необходимо реализовать при помощи JMeter следующий тест-план для нагрузочного тестирования. Ответы должны валидироваться по коду ответа (200 или 201). По итогам тестирования должен быть сформирован csv-файл.

Запрос	Распределение нагрузки, %
Получение гостевого токена (grant_type=client_credentials, scope=guest:default)	однократно для каждого потока
Регистрация игрока	однократно для каждого потока
Авторизация под игроком	однократно для каждого потока
Запрос профиля игрока	10
Запрос списка игр игроком	40
Запрос списка игр игроком с сортировкой	40
Запрос одной игры игроком по id	10

## API

URL: <http://test-api.d6.dev.devcaz.com>

HTTP basic authentication username: *front\_2d6b0a8391742f5d789d7d915755e09e*

## Authorization

### Client Credentials Grant

Access Token Request

#### Example

Request

```
POST /v2/oauth2/token HTTP/1.1
Host: server.example.com
Authorization: Basic c29tZWNSaWVudDphbmRpdHNzZWNyZXQ=
Content-Type: application/json
```

```
{
  "grant_type": "client_credentials",
  "scope": "scope1:read scope1:write scope2:read"
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "refresh_token": "def50200fcc006121b6a068eced57",
}
```

## Resource Owner Password Credentials Grant

### Access Token Request

#### Example

##### Request

```
POST /v2/oauth2/token HTTP/1.1
Host: server.example.com
Authorization: Basic c29tZWNSaWVudDphbmRpdHNzZWNyZXQ=
Content-Type: application/json

{
  "grant_type": "password",
  "username": "johndoe",
  "password": "am9obmRvZTEyMw=="
}
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "refresh_token": "def50200fcc006121b6a068eced57",
}
```

## Protected Resource Request

#### Example

##### Request

```
GET /v2/players HTTP/1.1
Host: server.example.com
Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
```

```
[]
```

# Players

## Register a new player

POST /v2/players HTTP/1.1

### Parameters

- **username**: string, required, unique user's login
- **password\_change**: string, required, base64 encoded password
- **password\_repeat**: string, required, base64 encoded password repeat
- **email**: string, required, user's email
- **name**: string, optional, user's first name
- **surname**: string, optional, user's last name
- **currency\_code**: string, required, default currency code according to ISO4217

### Example

#### Request

POST /v2/players HTTP/1.1

```
{
  "username": "janedoe",
  "password_change": "amFuZWVZTEYmW==",
  "password_repeat": "amFuZWVZTEYmW==",
  "email": "janedoe@example.com",
  "name": "Jane",
  "surname": "Doe",
  "currency_code": "EUR"
}
```

## Response

```
HTTP/1.1 201 Created
```

```
{
  "id": 1,
  "country_id": null,
  "timezone_id": null,
  "username": "johndoe",
  "email": "janedoe@example.com",
  "name": "John",
  "surname": "Doe",
  "gender": null,
  "phone_number": null,
  "birthdate": null,
  "bonuses_allowed": true,
  "is_verified": false
}
```

## Get a single player profile

GET /v2/players/:id HTTP/1.1

### Example

Request

```
GET /v2/players/1 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "id": 1,
  "country_id": 1,
  "timezone_id": null,
  "username": "johndoe",
  "email": null,
  "name": "John",
  "surname": "Doe",
  "gender": "male",
  "phone_number": "+1234567891",
  "birthdate": "1950-01-01",
  "bonuses_allowed": true,
  "is_verified": true
}
```

## Games

Retrieving collection or a single object

GET /v2/games

### Example (collection)

Request

```
GET /v2/games HTTP/1.1
```

## Response

HTTP/1.1 200 OK

```
[
  {
    "id": 622,
    "type_id": 1,
    "provider_id": 7,
    "name": "100 Super Hot",
    "description": "100 Super Hot",
    "has_tables": false,
    "is_favorite": false,
    "is_active": true
  },
  {
    "id": 626,
    "type_id": 1,
    "provider_id": 7,
    "name": "40 Super Hot",
    "description": "40 Super Hot",
    "has_tables": false,
    "is_favorite": false,
    "is_active": true
  }
]
```

## Example (single object)

### Request

GET /v2/games/622 HTTP/1.1

### Response

```
{
  "id": 622,
  "type_id": 1,
  "provider_id": 7,
  "name": "20 Super Hot",
  "description": "20 Super Hot",
  "has_tables": false,
  "is_favorite": false,
  "is_active": true
}
```

## Example (collection with sorting)

### Request

GET /v2/games?sort=name HTTP/1.1

## Response

HTTP/1.1 200 OK

```
[
  {
    "id": 626,
    "type_id": 1,
    "provider_id": 7,
    "name": "40 Super Hot",
    "description": "40 Super Hot",
    "has_tables": false,
    "is_favorite": false,
    "is_active": true
  },
  {
    "id": 622,
    "type_id": 1,
    "provider_id": 7,
    "name": "100 Super Hot",
    "description": "100 Super Hot",
    "has_tables": false,
    "is_favorite": false,
    "is_active": true
  }
]
```