

# Тестовое задание

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

## API

Автоматизировать следующие тесты используя Rest Assured. Необходимые выдержки из документации по API приведены ниже.

Тест	Ожидаемый результат
1 Получить токен гостя (Client Credentials Grant, scope — guest:default)	<ul style="list-style-type: none"><li>• HTTP response code — 200</li><li>• Ответ содержит токен</li></ul>
2 Зарегистрировать игрока	<ul style="list-style-type: none"><li>• HTTP response code — 201</li><li>• Ответ соответствует документации</li></ul>
3 Авторизоваться под созданным игроком (Resource Owner Password Credentials Grant)	<ul style="list-style-type: none"><li>• HTTP response code — 200</li><li>• Ответ содержит токен</li></ul>
4 Запросить данные профиля игрока	<ul style="list-style-type: none"><li>• HTTP response code — 200</li><li>• Ответ соответствует документации</li></ul>
5 Запросить данные другого игрока	<ul style="list-style-type: none"><li>• HTTP response code — 404</li></ul>

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

HTTP basic authentication username: *front\_2d6b0a8391742f5d789d7d915755e09e*

## UI

Автоматизировать следующие тесты используя связку Java + Selenium WebDriver + Cucumber

Тест	Ожидаемый результат
1 Авторизоваться в админке	<ul style="list-style-type: none"><li>• Пользователь успешно авторизован, админ-панель загрузилась</li></ul>
2 Открыть список игроков	<ul style="list-style-type: none"><li>• Таблица с игроками загрузилась</li></ul>
3 Отсортировать по любому столбцу и проверить сортировку	<ul style="list-style-type: none"><li>• Таблица верно отсортирована по выбранному столбцу</li></ul>

URL: <http://test-app.d6.dev.devcaz.com/admin/login>

Admin login/password: *admin1 / [9k<k8^z!+\$GkuP*

# 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

### 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, **optional**, default currency code according to ISO4217

#### Example

##### Request

```
POST /v2/players HTTP/1.1

{
  "username": "janedoe",
  "password_change": "amFuZWRvZTEyMw==",
  "password_repeat": "amFuZWRvZTEyMw==",
  "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
}
```