

# HireGL API description

## General notes:

- Each request (except login) has to have the following header: Authorization: Bearer <token>
- If header is not present or not valid, endpoints return 401
- If input JSON is not valid, endpoints return 400

## POST /api/auth/login

Login endpoint

Status code:

- 401 - unauthorized
- 200 - OK

Sample request:

```
{"userName": "user1", "password": "secret"}
```

Sample response:

```
{
  "user": {
    "firstName": "First",
    "lastName": "User",
    "loginName": "user1",
    "email": "first.user@user.com",
    "photoUrl": "http://localhost:8081/images/user1.png"
  },
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpYXQiOiJlODE2MzU2ODksInN1Yil6InVzZXIhIn0.urh7CmQmz4c03dVq5qoFyFwXypTqW_95x9uBtvoKrVY9IVDeMXBjK2nCp_R9hXgPKuR_ggdqUhZPMu7uSZIUeA"
}
```

## POST http://localhost:8081/api/auth/logout

Logout endpoint

Status code:

- 204 - OK

## GET /api/locations

List of all locations

Status code:

- 200 - OK

Sample response:

```
["BANSKA BYSTRICA","KOSICE","ZILINA"]
```

## GET /api/locations/{location}/rooms

List of all rooms for given location

Status code:

- 200 - OK

Sample response:

```
["Matrix","Barcode"]
```

## GET /api/positions

List of all positions

Status code:

- 200 - OK

Sample response:

```
["ARCHITECT","DEVELOPMENT MANAGER","LEAD DEVELOPER","PROJECT  
MANAGER","SCRUMMASTER","TEST MANAGER"]
```

## GET /api/users

List of all users

Status code:

- 200 - OK

Sample response:

```
{
  "firstName": "First",
  "lastName": "User",
  "loginName": "user1",
  "email": "first.user@user.com",
  "photoUrl": "http://localhost:8081/images/user1.png"
}, {
  "firstName": "Second",
  "lastName": "User",
  "loginName": "user2",
  "email": "second.user@user.com",
  "photoUrl": "http://localhost:8081/images/user2.png"
}, {
  "firstName": "Third",
  "lastName": "User",
  "loginName": "user3",
  "email": "third.user@user.com",
  "photoUrl": "http://localhost:8081/images/user3.png"
}, {
  "firstName": "Fourth",
  "lastName": "User",
  "loginName": "user4",
  "email": "fourth.user@user.com",
  "photoUrl": "http://localhost:8081/images/user4.png"
}]
```

## POST /api/interviews

Creates interview

Status code:

- 201 - OK

Sample request:

```
{
  "candidate": {
    "firstName": "first name",
    "lastName": "last name",
    "phone": "111",
    "skype": "skype",
    "email": "email@email.sk",
    "position": "PROJECT MANAGER"
  }
}
```

```

    },
    "interview": {
      "location": "KOSICE",
      "room": "MATRIX",
      "dateTime": "2016-12-13T09:34Z",
      "userId": 1
    }
  }
}

```

Sample Response: Location header contains URL to created resource, example: Location: <http://localhost:8081/interviews/7>

Notes:

- userId is optional parameter
- skype is optional parameter

## GET /api/interviews?limit=x?start=y

Returns all interviews (starts at yth record and returns x records)

Sample response:

```

[{
  "id": 2,
  "candidate": {
    "firstName": "first name",
    "lastName": "last name",
    "phone": "111",
    "email": "email@email.sk",
    "skype": "skype",
    "position": "PROJECT MANAGER"
  },
  "interview": {
    "dateTime": "2016-12-13T09:34",
    "location": "KOSICE",
    "room": "MATRIX",
    "note": null,
    "status": "CREATED"
  }
}, {
  "id": 1,
  "candidate": {
    "firstName": "new first name",

```

```

        "lastName": "new last name",
        "phone": "new 111",
        "email": "new email@email.sk",
        "skype": "new skype",
        "position": "ARCHITECT"
    },
    "interview": {
        "dateTime": "2017-12-13T09:34",
        "location": "KOSICE",
        "room": "MATRIX",
        "note": null,
        "status": "CREATED"
    }
}, {
    "id": 3,
    "candidate": {
        "firstName": "first name",
        "lastName": "last name",
        "phone": "111",
        "email": "email@email.sk",
        "skype": "skype",
        "position": "PROJECT MANAGER"
    },
    "interview": {
        "dateTime": "2016-12-13T09:34",
        "location": "KOSICE",
        "room": "MATRIX",
        "note": null,
        "status": "CREATED",
        "user": {
            "id": 1,
            "firstName": "First",
            "lastName": "User",
            "email": "first.user@user.com",
            "loginName": "user1",
            "photoUrl": "http://localhost:8081/images/user1.png"
        }
    }
}
}]

```

#### Notes:

- If limit and start parameters are not present, it returns all records.
- If both limit and start parameters are present, it returns values from start to start + limit

## GET /api/interviews/{interviewId}

Gets one interview

Sample response:

```
{
  "id": 1,
  "candidate": {
    "firstName": "new first name",
    "lastName": "new last name",
    "phone": "new 111",
    "email": "new email@email.sk",
    "skype": "new skype",
    "position": "ARCHITECT"
  },
  "interview": {
    "dateTime": "2017-12-13T09:34",
    "location": "KOSICE",
    "room": "MATRIX",
    "note": null,
    "status": "CREATED"
  }
}
```

Status code:

- 200 - OK
- 404 - Resource doesn't exist

## PUT /api/interviews/{interviewId}

Updates one interview

Status code:

- 200 - OK
- 404 - Resource doesn't exist
- 409 - Interview doesn't have valid state for update

Sample request:

```
{
  "candidate": {
    "firstName": "new first name",
```

```
    "lastName": "new last name",
    "phone": "new 111",
    "email": "new email@email.sk",
    "skype": "new skype",
    "position": "ARCHITECT"
  },
  "interview": {
    "dateTime": "2017-12-13T09:34",
    "location": "KOSICE",
    "room": "MATRIX",
    "userId": 3
  }
}
```

Notes:

- userId is optional parameter

## DELETE /api/interviews/{interviewId}

Deletes one interview

Status code:

- 204 - OK
- 404 - Resource doesn't exist

## PUT /api/interviews/{interviewId}/closed

Closes one interview

Status code:

- 204 - OK
- 404 - Resource doesn't exist

## GET /api/interviews/count

Count of interviews

Status code:

- 200 - OK

Sample response: {"count":1}