

Rest API Documentation of WebAlk_assignment

Bodnár Renátó
A4R0OD

URL Pattern:

/jobs/list

Title:

Giving back all the jobs from the server. Its is for browsing without any sort, and without the HRID.

Method:

HTTP method: GET

URL parameters:

Body parameters:

Successful response:

Satus code: 200

Response body: JSON

Structure definition:

```
{
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data.

Example request:

jobs/list

URL Pattern:

/jobs/searchWithName

Title:

Giving back all the data from the server, that contains in the name field the text, what is it in the parameter. It is for search a job, when only remember one word in the name.

Method:

HTTP method: GET

URL parameters:

Required: name = <text>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server don't have any data with that parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

jobs/searchWithName?name=Kukutyin

URL Pattern:

/jobs/searchWithMinMoney

Title:

List the jobs, that reached a minimum amount of salary, and the acceptable jobs in a sort of increasing salary.

Method:

HTTP method: GET

URL parameters:

Required: money = <integer>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server don't have any data with that parameter

HTTP Status 400 – Bad request //if the required parameter is missing

HTTP Status 406 – Not Acceptable //if the salary in the parameter doesn't reach the international minimum salary

Example request:

jobs/searchWithMinMoney?money=125000

URL Pattern:

/jobs/searchWithPlace

Title:

List the jobs, that are on the same location what given in the parameter. It is for search jobs close to the peoples.

Method:

HTTP method: GET

URL parameters:

Required: place = <text>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data with that parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

jobs/searchWithPlace?place=Budapest

URL Pattern:

/jobs/searchWithLanguage

Title:

List the jobs, that need that programming what is it in the parameter. It is for search jobs, where need to use that language what the people know.

Method:

HTTP method: GET

URL parameters:

Required: language= <text>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data with that parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

jobs/searchWithLanguage?language=java

URL Pattern:

/jobs/searchWithEducation

Title:

List the jobs, that need that level what it is in the parameter. It is for search jobs, with that education level what people absolved.

Method:

HTTP method: GET

URL parameters:

Required: education= <text>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data with that parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

jobs/searchWithEducation?education=master

URL Pattern:

/HR/add

Title:

Add a new job as a HR.

Method:

HTTP method: POST

URL parameters:

Body parameters:

```
{
    „hrid”: Text,
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    „hrid”: Text,
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 409 – Conflict //if the job ID already exists, or the name is the same with an existing one

HTTP Status 406 – Not Acceptable //if the salary doesnt reach the national minimum salary, or the email format is bad

Example request:

```
{
  "hrid": "Alabama",
  "jobID": 8,
  "jobName": "Blabla",
  "jobDescription": "Blabla",
  "jobEduReq": "master",
  "jobLangReq": "java",
  "jobPlace": "Miskolc",
  "jobMoney": 200000,
  "contact": "probah@gmail.com"
}
```

URL Pattern:

/HR/listByABC

Title:

List the jobs, with a sort by HR_ID in increasing by ABC.

Method:

HTTP method: GET

URL parameters:

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    „hrid”: Text,
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data

Example request:

HR/listByABC

URL Pattern:

/HR/searchHR

Title:

Search in the list with HR_ID.

Method:

HTTP method: GET

URL parameters:

Required: HR = <Text>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    „hrid”: Text,
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data with the parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

HR/searchHR?HR=Susan

URL Pattern:

/HR/searchWithID

Title:

Search in the list with Job_ID.

Method:

HTTP method: GET

URL parameters:

Required: ID = <Integer>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    „hrid”: Text,
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server dont have any data with the parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

HR/searchWithID?ID=2

URL Pattern:

/HR/listByHRWithDegreese

Title:

Search in the list with HR_ID, and the results is sort by decreasing salary.

Method:

HTTP method: GET

URL parameters:

Required: ID = <Integer>

Body parameters:

Successful response:

Status code: 200

Response body: JSON

Structure definition:

```
{
    „hrid”: Text,
    "jobID": Integer,
    "jobName": Text,
    "jobDescription": Text,
    "jobEduReq": Enum in text,
    "jobLangReq": "Enum in text,
    "jobPlace": Text,
    "jobMoney": Integer,
    "contact": Email address in text
}
```

Error response:

HTTP Status 404 – Not Found //if the server don't have any data with the parameter

HTTP Status 400 – Bad request //if the required parameter is missing

Example request:

HR/listByHRWithDegreese?HR=Susan