

Freelance Job Manager Backend

Bogdan A. Constantin

May 12, 2016

Contents

1	Installing and Running the Server	3
2	Updating the Server	3
3	Method Reference	3
3.1	Session Methods	3
3.1.1	Login	3
3.2	User Methods	4
3.2.1	User Registration	4
3.2.2	Read User	4
3.2.3	Update user	4
3.2.4	List Users	4
3.2.5	User Update	5
3.2.6	Delete User	5
3.3	Freelancer Methods	5
3.3.1	Freelancer Registration	5
3.3.2	Update Freelancer	6
3.3.3	Read Freelancer	6
3.3.4	List Freelancers	6
3.3.5	Delete Freelancer	6
3.4	JobType Methods	7
3.4.1	Create Job Type	7
3.4.2	Update Job Type	7
3.4.3	Read Job Type	7
3.4.4	List Job Types	7
3.4.5	Delete Job Type	7
3.5	Location Methods	8
3.5.1	Create Location	8

3.5.2	Update Location	8
3.5.3	Read Location	8
3.5.4	List Locations	8
3.5.5	Delete Location	9
3.6	Job Methods	9
3.6.1	Job Statuses	9
3.6.2	Create Job Request	9
3.6.3	Update Job Request	9
3.6.4	Delete Job Request	9
3.6.5	Read Job	9
3.6.6	List User Jobs by Status	9
3.6.7	List Freelancer Jobs by Status	10
3.6.8	List Open Jobs by JobType	10
3.6.9	List Open Jobs by JobType and Location Area	10
3.6.10	Request payment	11
3.7	Subscription Methods	11
3.7.1	Subscribe / Update subscription	11
3.7.2	Accept subscription	11
3.7.3	Delete subscription	11
3.7.4	List Job subscriptions	11
3.7.5	List Freelancer Subscriptions	12

1 Installing and Running the Server

- Unzip the archive (don't create a new folder)
- cd into the FreeJobServer directory
- Run start.bat. java.exe must be installed and on the path. If not, please edit start.bat and use the full java.exe path.

2 Updating the Server

- To stop the server enter **stop 0** on the console and then press ENTER.
- Unpack the update archive into the /bundle directory.
- Delete **felix-cache** directory
- Restart the server

3 Method Reference

The REST servlet is accessible at the following URL: localhost:48080/freejob
Each method URL must be prefixed with this string.

3.1 Session Methods

3.1.1 Login

POST /sessions/login Login as a customer or freelancer. Sample payload for customer login:

```
{
  "role" : "CUSTOMER",
  "email" : "johndoe.constantin@itsix.com",
  "password" : "mypassword"
}
```

RESPONSE Sample for a customer login.

```
{
  "data": {
    "sessionId": "e8599496-7099-41b7-a554-65dfd38e02f2",

```

```

    "user": {
      "id": "5a6a7a5b-a66c-4cb8-bc61-14dea633a8ca",
      "firstName": "John",
      "lastName": "Doe ITSIX",
      "email": "johndoe.constantin@itsix.com",
      "role": "CUSTOMER",
      "password": null
    },
    "created": 1461337503669
  },
  "success": true,
  "message": null
}

```

3.2 User Methods

3.2.1 User Registration

POST /users/register Register a new user. Sample payload:

```

{
  "firstName" : "John",
  "lastName" : "Doe",
  "email" : "johndoe@example.org",
  "password" : "mypassword",
}

```

3.2.2 Read User

GET /users/{userId} Read user.

3.2.3 Update user

POST /users Update user data

3.2.4 List Users

GET /users List all users. Sample response:

```

{
  "data": [
    {
      "id": "5fd2124b-3cf9-434a-97df-5b0af469a9f2",

```

```

        "firstName": "John",
        "lastName": "Doe",
        "email": "johndoe@example.org",
        "role": "USER",
        "password": null
    }
],
"success": true,
"message": null
}

```

3.2.5 User Update

POST /users Update user data. If id is null, a new user is created.
Payload:

```

{
  "id": "5fd2124b-3cf9-434a-97df-5b0af469a9f2",
  "firstName": "John",
  "lastName": "Doe",
  "email": "johndoe@example.org",
  "role": "USER",
  "password": null
}

```

3.2.6 Delete User

DELETE /users/{userId} Delete a user.

3.3 Freelancer Methods

3.3.1 Freelancer Registration

POST /freelancers/register Register a new freelancer. Sample payload:

```

{
  "jobTypeId": "60243132-4316-4400-86aa-7b17c866584b",
  "firstName": "John",
  "lastName": "Doe",
  "email": "johndoe@example.org",
  "password": "some password",
}

```

3.3.2 Update Freelancer

POST /freelancers Update freelancer

3.3.3 Read Freelancer

GET /freelancers/{freelancerId} Read freelancer.

3.3.4 List Freelancers

GET /freelancers List all freelancers. Sample response:

```
{
  "data": [
    {
      "id": "baf538a8-fbbe-4bdf-bc83-7819f60b0d62",
      "jobTypeId": "60243132-4316-4400-86aa-7b17c866584b",
      "firstName": "John",
      "lastName": "Doe",
      "email": "johndoe@example.org",
      "password": null,
      "address": null,
      "latitude": null,
      "longitude": null,
      "city": null,
      "county": null,
      "avgRating": 0,
      "bankName": null,
      "accountNumber": null
    }
  ],
  "success": true,
  "message": null
}
```

3.3.5 Delete Freelancer

DELETE /freelancers/{freelancerId} Delete a freelancer.

3.4 JobType Methods

3.4.1 Create Job Type

POST /jobtypes Create a new job type. Sample payload:

```
{
  "name" : "Baby-Sitter",
  "description" : "Takes care of babies",
  "commission": 2.5
}
```

3.4.2 Update Job Type

POST /jobtypes

3.4.3 Read Job Type

GET /jobtypes/{jobTypeId} Read job type.

3.4.4 List Job Types

GET /jobtypes List all job types. Sample response:

```
{
  "data": [
    {
      "id": "60243132-4316-4400-86aa-7b17c866584b",
      "name": "Baby-Sitter",
      "description": "Takes care of babies",
      "commission" : 2.5
    }
  ],
  "success": true,
  "message": null
}
```

3.4.5 Delete Job Type

DELETE /jobtypes/{jobTypeId} Delete a job type.

3.5 Location Methods

3.5.1 Create Location

POST /users/{userId}/locations Create a new location for a user.

Sample payload:

```
{
  "name" : "Home",
  "address" : "Sesame Street 15",
  "city" : "Pielesti",
  "county" : "Dolj",
  "latitude" : 45.318266,
  "longitude" : 24.825482
}
```

3.5.2 Update Location

POST /users/{userId}/locations

3.5.3 Read Location

GET /users/{userId}/locations/{locationId} Read a location.

3.5.4 List Locations

GET /users/{userId}/locations List all locations of a user. Sample response:

```
{
  "data": [
    {
      "id": "364b5a88-6ebe-43da-8ebb-0035e1305d5a",
      "name" : "Home",
      "userId": "5a6a7a5b-a66c-4cb8-bc61-14dea633a8ca",
      "address": "Sesame Street 15",
      "city": "Pielesti",
      "county": "Dolj",
      "latitude": 45.318266,
      "longitude": 24.825482
    }
  ],
  "success": true,
}
```



```
  "message": null
}
```

3.5.5 Delete Location

DELETE /users/{userId}/locations/{locationId} Delete a location for a user.

3.6 Job Methods

3.6.1 Job Statuses

Valid Job statuses are OPEN, ACCEPTED, ONGOING, FINISHED, PAYMENT_REQUESTED, BILLED, CANCELLED

3.6.2 Create Job Request

POST /users/{userId}/jobs Create a new job request for a user. Sample payload:

```
{
  "title" : "Baby sitter request",
  "description" : "Need baby sitter for a 5 yr old.",
  "jobTypeId" : "e0d136eb-8e33-426d-aef3-a66ce5c668de",
  "locationId" : "364b5a88-6ebe-43da-8ebb-0035e1305d5a"
}
```

3.6.3 Update Job Request

POST /users/{userId}/jobs

3.6.4 Delete Job Request

DELETE /users/{userId}/jobs/{jobId} Delete a job request.

3.6.5 Read Job

GET /users/{userId}/jobs/{jobId} Read job

3.6.6 List User Jobs by Status

GET /users/{userId}/jobs?status={status} List all jobs of a user. Optionally, a status may be specified.

3.6.7 List Freelancer Jobs by Status

GET /freelancers/{freelancerId}/jobs?status={status} List all jobs of a freelancer. Optionally, a status may be specified.

3.6.8 List Open Jobs by JobType

GET /jobtypes/{jobTypeId}/jobs List all open jobs of a specific type.

3.6.9 List Open Jobs by JobType and Location Area

GET /jobtypes/{jobTypeId}/jobs/latitude/{latitude}/

longitude/{longitude}/range/{range} List all open jobs of a specific type by location coordinates and a range in km.

Sample response:

```
{
  "data": [
    {
      "id": "6ec322a1-c190-4386-a17f-5ff9cb87b01c",
      "status": "OPEN",
      "title": "Baby sitter request",
      "description": "",
      "created": 1462883463063,
      "rating": 0,
      "jobTypeId": "e0d136eb-8e33-426d-aef3-a66ce5c668de",
      "freelancerId": null,
      "userId": "5a6a7a5b-a66c-4cb8-bc61-14dea633a8ca",
      "locationId": "364b5a88-6ebe-43da-8ebb-0035e1305d5a",
      "netAmount": 0.0,
      "total": 0.0,
      "latitude": 45.318266,
      "longitude": 24.825482
    }
  ],
  "success": true,
  "message": null
}
```

3.6.10 Request payment

POST /freelancers/{freelancerId}/jobs/{jobId}/netAmount/{netAmount}
Request payment for the job. Sets the netAmount, computes total and sets status to PAYMENT_REQUESTED.

3.7 Subscription Methods

3.7.1 Subscribe / Update subscription

POST /freelancers/{freelancerId}/jobs/{jobId}/subscriptions Subscribe to a job. This method can also be used to modify the subscription message.

Sample payload:

```
{
    "message" : "Some message from the freelancer"
}
```

3.7.2 Accept subscription

POST /users/{userId}/jobs/{jobId}/subscriptions/{freelancerId}
Accept freelancer subscription. Sets ACCEPTED status and copies freelancerId and message to the job.

3.7.3 Delete subscription

DELETE /freelancers/{freelancerId}/jobs/{jobId}/subscriptions Delete subscription

3.7.4 List Job subscriptions

GET /users/{userId}/jobs/{jobId}/subscriptions List all subscriptions to a job

Sample response:

```
{
  "data": [
    {
      "jobId": "6ec322a1-c190-4386-a17f-5ff9cb87b01c",
      "freelancerId": "e0d136eb-8e33-426d-aef3-a66ce5c668de",
      "message": "Some message from the freelancer"
    }
  ]
}
```

```
],  
  "success": true,  
  "message": null  
}
```

3.7.5 List Freelancer Subscriptions

GET /freelancers/{freelancerId}/subscriptions List all freelancer subscriptions

```
{  
  "data": [  
    {  
      "jobId": "6ec322a1-c190-4386-a17f-5ff9cb87b01c",  
      "freelancerId": "e0d136eb-8e33-426d-aef3-a66ce5c668de",  
      "message": "Some message from the freelancer"  
    }  
  ],  
  "success": true,  
  "message": null  
}
```