Job Advertisement Application API Reference

Balázs Bordás - HCFECD

Query all jobs stored by the application:

URL: `/job/all`

Request Method: GET

Request Params: *None*

Success Response:

- Code: **200**
- Response type: application/json.
- Example Response Content:

```
[
    {
        "name": "Cleaner",
        "description": "We need someone to clean the office.",
        "uploadDate": "2018-12-12",
        "companyName": "Cleaner Inc.",
        "contact": "someone@cleaner.io",
        "wage": 2000,
        "type": "CLEANER"
    }
]
```

Query jobs by provided filters:

URL: `/job/by`

Request Method: POST

Request Params: *None*

Example Request Body:

```
{
  "filters": [
    {
      "fieldName": "string",
      "fieldValue": {},
      "type": "string"
    }
  ]
}
```

The filter should be provided as the following:

- **fieldName**: field of the model which you'd like to filter by
- **fieldValue**: value of which the stored entities will be compared to. This object can be more than just a basic type in cases of Date and Number filters. See the following example to determine how you should use in cases of different filter types:
    - *NUMBER*: you should provide the minimum and/or the maximum value of the field like the following:

      ```
      "fieldValue":

      {
            "min": 1000,
            "max": 2000
      }
      ```
    - *DATE*: you should provide the minimum and/or the maximum value of the field like the following:

      ```
      "fieldValue":

      {
            "min": "2018-12-11",
            "max": "2018-12-13"
      }
      ```
    - *EXACTMATCH*: you should provide the string value which you're exactly looking for. No similar values will be queried, only the strictly matching ones. Example:

      ```
      "fieldValue": "valueInString"
      ```

    - *PARTIALMATCH*: you should provide the string value which could occur in any substring of the value. If the field value contains the provided string in any way, it will be fetched by the application. Example:

      ```
      "fieldValue": "valueInString"
      ```

- **type**: type of the filter to use. Filter type can only be one of the following ones, determined by the type of the field which you'd like to query:
  `["NUMBER", "DATE", "EXACTMATCH", "PARTIALMATCH"]`

Success Response:

- Code: **200**
- Response type: application/json.
- Example Response Content:
```
[
    {
        "name": "Cleaner",
        "description": "We need someone to clean the office.",
        "uploadDate": "2018-12-12",
        "companyName": "Cleaner Inc.",
        "contact": "someone@cleaner.io",
        "wage": 2000,
        "type": "CLEANER"
    }
]
```

Error Response:

- Code: **400**
- Response type: text/plain.
- Example Response Content:

```
Search request filters are not valid.
java.lang.IllegalArgumentException: Unable to locate Attribute  with
the the given name [invalidField] on this ManagedType
[hu.iit.me.model.Job]
```

Store job entity in application:

URL: `/job`

Request Method: POST

Request Params: *None*

Example Request Body:

```
{
  "companyName": "string",
  "contact": "string",
  "description": "string",
  "name": "string",
  "type": "string",
  "uploadDate": "string",
  "wage": 0
}
```
Additional fields in the request body will be ignored.

Success Response:

- Code: **200**
- Response type: text/plain.
- Example Response Content:

```
Job saved successfully
```

Error Response:

- Code: **400**
- Response type: text/plain.
- Example Response Content:

```
Job fields are not valid.
```

Query all applicants stored by the application:

URL: /applicant/all

Request Method: GET

Request Params: *None*

Success Response:

- Code: **200**
- Response type: application/json.
- Example Response Content:

```
[
    {
        "foreName": "Ilona",
        "surName": "Kiss",
        "birthDate": "2018-11-23",
        "phoneNumber": "36205556478",
        "taxId": "123456791",
        "gender": "woman",
        "address": "23. Random Str. Hungary"
    }
]
```

Query applicants by provided filters:

URL: `/applicant/by`

Request Method: POST

Request Params: *None*

Example Request Body:

```
{
  "filters": [
    {
      "fieldName": "string",
      "fieldValue": {},
      "type": "string"
    }
  ]
}
```

The filter should be provided as the following:

- **fieldName**: field of the model which you'd like to filter by
- **fieldValue**: value of which the stored entities will be compared to. This object can be more than just a basic type in cases of Date and Number filters. See the following example to determine how you should use in cases of different filter types:
    - *NUMBER*: you should provide the minimum and/or the maximum value of the field like the following:

      ```
      "fieldValue":

      {
            "min": 1000,
            "max": 2000
      }
      ```
    - *DATE*: you should provide the minimum and/or the maximum value of the field like the following:

      ```
      "fieldValue":

      {
            "min": "2018-12-11",
            "max": "2018-12-13"
      }
      ```
    - *EXACTMATCH*: you should provide the string value which you're exactly looking for. No similar values will be queried, only the strictly matching ones. Example:

      ```
      "fieldValue": "valueInString"
      ```

    - *PARTIALMATCH*: you should provide the string value which could occur in any substring of the value. If the field value contains the provided string in any way, it will be fetched by the application. Example:

      ```
      "fieldValue": "valueInString"
      ```

- **type**: type of the filter to use. Filter type can only be one of the following ones, determined by the type of the field which you'd like to query: ["NUMBER", "DATE", "EXACTMATCH", "PARTIALMATCH"]

Success Response:

- Code: **200**
- Response type: application/json.
- Example Response Content:

```
[
    {
        "foreName": "Ilona",
        "surName": "Kiss",
        "birthDate": "2018-11-23",
        "phoneNumber": "36205556478",
        "taxId": "123456791",
        "gender": "woman",
        "address": "23. Random Str. Hungary"
    }
]
```

Error Response:

- Code: **400**
- Response type: text/plain.
- Example Response Content:

```
Search request filters are not valid.
java.lang.IllegalArgumentException: Unable to locate Attribute  with
the the given name [invalidField] on this ManagedType
[hu.iit.me.model.Job]
```

Store applicant entity in application:

URL: /applicant

Request Method: POST

Request Params: *None*

Example Request Body:

```
{
  "address": "string",
  "birthDate": "string",
  "foreName": "string",
  "gender": "string",
  "phoneNumber": "string",
  "surName": "string",
  "taxId": "string"
}
```
Additional fields in the request body will be ignored.

Success Response:

- Code: **200**
- Response type: text/plain.
- Example Response Content:

  ```
  Applicant saved successfully
  ```

Error Response:

- Code: **400**
- Response type: text/plain.
- Example Response Content:

  ```
  Applicant fields are not valid.
  ```