

# Web Application Development

## Endpoint-reference

### Javorek Dénes

---

# Advertise Job (Operator)

Create a new Job with the given properties, if the same exact Job entity doesn't exist already.

**URL:** /operator/addJob

**Method:** POST

## Body Parameters:

```
<advertiseJobRequest>
  <job>
    <name>[string] </name>
    <type>['ACCOUNTING', 'SALES', 'MANAGEMENT', 'RESEARCH',
'MANUFACTURING'] </type>
    <city>[string] </city>
    <wage>[integer] </wage>
    <description>[string] </description>
    <requirements>
      <!--Zero or more repetitions:-->
      <requirement>[string] </requirement>
    </requirements>
  </job>
</advertiseJobRequest>
```

## Example Request:

```
curl --request POST \
  --url http://localhost:8080/operator/addJob \
  --header 'Content-Type: application/xml' \
  --data '<advertiseJobRequest>
<job>
  <name>Example Job</name>
  <type>RESEARCH</type>
  <city>Pécs</city>
  <wage>800</wage>
  <description>Example description for the Job</description>
  <requirements>
    <requirement>Being a good example</requirement>
    <requirement>Having multiple requirements</requirement>
  </requirements>
</job>
</advertiseJobRequest>'
```

## Successful Response

**Condition:** If everything is OK and the requested Job didn't exist yet

**Code:** 200 OK

**Response example:**

```
{
  "model": {
    "name": "Example Job",
    "type": "MANAGEMENT",
    "city": "Szolnok",
    "wage": 1800,
    "description": "Example description for the Job",
    "requirements": {
      "requirement": [
        "Being a good example",
        "Having multiple requirements"
      ]
    }
  }
}
```

## Error Response

**Condition:** If the same Job already exists

**Code:** 409 Conflict

# List Application (Operator)

List all applications that was created before

**URL:** `/operator/ listApplication`

**Method:** `GET`

## Successful Response

**Condition:** `If everything is OK.`

**Code:** `200 OK`

## Example Request:

```
curl --request GET \  
--url http://localhost:8080/operator/listApplication
```

## Response example:

```
{
  "model": {
    "application": [
      {
        "job": {
          "name": "Superior AdvertiseJob endpoint operator",
          "type": "MANAGEMENT",
          "city": "Miskolc",
          "wage": 50,
          "description": "So you can do this, instead of me!",
          "requirements": {
            "requirement": [
              "WINDOWS/linux knowledge",
              "Postman"
            ]
          }
        },
        "applicant": {
          "firstname": "Luke",
          "lastname": "Skywalker",
          "email": "lukeskywalker@notexisting.not",
          "phonenumber": "43464365466"
        },
        "creationDate": "2018-11-01 10:00:00",
        "comment": "Sounds good, can I earn some money?"
      }
    ]
  }
}
```

## Notes

- If there is no application to show, empty application list will be returned

# List Applications (Worker)

List all applications that was created before

**URL:** `/worker/:applicantId/ listApplication`

**URL Params:**

**Required:**

`applicantId=[integer]`

**Method:** `GET`

## Successful Response

**Condition:** `If everything is OK.`

**Code:** `200 OK`

**Example Request:**

```
curl --request GET \  
--url http://localhost:8080/worker/1/listApplications
```

## Response example:

```
{
  "model": {
    "application": [
      {
        "job": {
          "name": "Junior Trash Relocating Worker",
          "type": "MANUFACTURING",
          "city": "Budapest",
          "wage": 600,
          "description": "Sounds great, isn't it? ;)",
          "requirements": {
            "requirement": [
              "Endurance",
              "Loyalty",
              "Being morning-person"
            ]
          }
        },
        "applicant": {
          "firstname": "Thomas",
          "lastname": "Test",
          "email": "testUser01@donotreply.org",
          "phonenumber": "00998887766"
        },
        "creationDate": "2017-04-02 15:00:00",
        "comment": "Trash is my life, pls choose me"
      }
    ]
  }
}
```

## Notes

- If there is no application to show, empty application list will be returned

# Search Applications (Worker)

Search from those applications that was created by a given applicant (worker). Search parameters can be provided with URL params, if you choose to not give any filter, then all available applications will be returned.

**URL:** `/worker/:applicantId/ listApplication`

## URL Params:

**Required:**

`applicantId=[integer]`

**Optional:**

`applicationId=[integer]`

`jobId=[integer]`

`createdAfter=[date, format="yyyyMMdd"]`

`createdBefore=[date, format="yyyyMMdd"]`

`hasComment=[boolean]`

**Method:** `GET`

## Successful Response

**Condition:** `If everything is OK.`

**Code:** `200 OK`

## Example Request:

`curl --request GET \`

`--url`

`'http://localhost:8080/worker/1/searchApplications?applicationId=2&jobId=2&createdAfter=20170401&createdBefore=20170403&hasComment=true'`



## Response example:

```
{
  "model": {
    "application": [
      {
        "job": {
          "name": "Junior Trash Relocating Worker",
          "type": "MANUFACTURING",
          "city": "Budapest",
          "wage": 600,
          "description": "Sounds great, isn't it? ;)",
          "requirements": {
            "requirement": [
              "Endurance",
              "Loyalty",
              "Being morning-person"
            ]
          }
        },
        "applicant": {
          "firstname": "Thomas",
          "lastname": "Test",
          "email": "testUser01@donotreply.org",
          "phonenumber": "00998887766"
        },
        "creationDate": "2017-04-02 15:00:00",
        "comment": "Trash is my life, pls choose me"
      }
    ]
  }
}
```

## Notes

- If there is no application to show, empty application list will be returned
- If no filter provided with url params, it gives the same result as *List Applications (Worker)* endpoint