

# Development

---

The following tools were used for development.

- JDK 11+ (Java Development Kit 11+)
- IDE: Eclipse or IntelliJ
- Java Framework: Springboot
- Postman: Generate JSON requests (for testing purposes)
- PostgreSQL: Data storage
- pgAdmin4: Admin console for PostgreSQL
- Docker: Contains image of postgresql and pgAdmin4

## Java Development Kit(JDK)

---

It is recommended to use JDK 11+.

Download and [install JDK](#).

Important: Make sure to [set up the environment variables](#).

## IDE: Eclipse or IntelliJ

---

We recommend using one of the following IDEs:

- [Eclipse](#): Eclipse IDE for Java Developers
- [IntelliJ](#): Community, Education or Ultimate edition. Important: Use version 2020.1 or later!

## Postman (optional)

---

[Postman](#) is a tool to create GET/POST/... requests. We recommend using it in order to debug the API.

## Docker

---

We use Docker to run our development database.

- [Download](#) for Windows
- [Download](#) for Mac

## How to use

---

- Install the below in-order
  - JDK 11+ and set environment variables
  - Install IDE and [set the JDK path](#) if need.
  - Install Docker
- Clone and pull the code
- Run `docker-compose up`
- Run the Digitact/Backend/Application.java file using your preferred IDE

# Example

---

Execute the following commands in Postman to verify your server is working correctly.

## POST request

---

Send a POST request to the following URL:

```
http://localhost:9090/api/controller/createApplicant
```

Example body of the request (important: send it as JSON)

```
{
  "firstName": "Lionel",
  "lastName": "Messi"
}
```

## GET request

---

After you have added at least one applicant using a POST request, you can retrieve them from the following URL:

```
http://localhost:9090/api/controller/getusers
```

The response should look like this:

```
[
  {
    "firstName": "Lionel",
    "lastName": "Messi"
  }
]
```

## pgAdmin4

---

In order to view the content of the database, we ship pgAdmin4 with our docker-compose file. Open `localhost:8080` to open pgAdmin4.

Log in using:

- Email Address: `example@example.com`
- Password: `example`

If you want to access the database, you are most likely prompted for a password. It is set to `postgres` by default.

## Create Applicant Restful API

---

```

{
  "firstName": String,
  "lastName": String,
  "industries": {
    "automotive": boolean,
    "finance": boolean,
    "commerce": boolean,
    "pharma_Healthcare": boolean,
    "public_sector": boolean
  },
  "positions": {
    "consultant_Business_Consultant": boolean,
    "iT_Consultant_Informationmanagement": boolean,
    "iT_Consultant_Java_JEE": boolean,
    "iT_Consultant_Data_Science": boolean,
    "iT_Consultant_Artificial_Intelligence": boolean,
    "internship_working_Student": boolean,
    "consultant_SAP": boolean
  },
  "educations": [
    {
      "university": String,
      "subject": String,
      "degree": String,
      "grade": double,
      "graduation_date": String
    }
  ],
  "keyCompetencies": [
    {
      "category": KeyCompetenciesCategory,
      "name": String,
      "rating": int
    }
  ]
}

```

KeyCompetenciesCategory is either `languages`, `professionalSoftware`, `databases` or `programmingLanguagesAndFrameworks`.