

---

# **MaMoDaR Documentation**

**Kyanoush Yahosseini, Janine Straka, Luca Leipold**

**Jul 22, 2020**



---

## Contents:

---

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Installation</b>                    | <b>1</b> |
| <b>2</b> | <b>Requirements</b>                    | <b>3</b> |
| <b>3</b> | <b>Set Up</b>                          | <b>5</b> |
| 3.1      | Create a PostgreSQL database . . . . . | 5        |
| 3.2      | Set-Up . . . . .                       | 5        |
| <b>4</b> | <b>Run project</b>                     | <b>7</b> |
| <b>5</b> | <b>Indices and tables</b>              | <b>9</b> |



# CHAPTER 1

---

## Installation

---

The DataLinker can be installed on Linux, Windows and macOS.

The code is written in Java and Angular.

An installation of the DataLinker contains of three parts:

- 1) A frontend with the webserver (written in Angular)
- 2) A backend (written in Java)
- 3) A Postgres database to store the content, which is generated by the users of your DataLinker.

This chapter shows how these components are set up.



## CHAPTER 2

---

### Requirements

---

- Git: (<https://git-scm.com/download/win>)
- The database PostgreSQL <https://www.postgresql.org/download/windows>
- JDK version  $\geq 11$ : <https://adoptopenjdk.net/>
- Gradle: <https://gradle.org/install/#manually>
- Node.js: <https://nodejs.org/en/>
- We recommend Visual Studio Code as editor: <https://code.visualstudio.com/>
- **A RDMO server:**
  - Description: <https://rdmorganiser.github.io/>
  - Documentation: <https://rdmo.readthedocs.io/en/latest/>
- A token for the communication between RDMO and the DataLinker (<https://rdmo.readthedocs.io/en/latest/administration/api.html?highlight=token#authentication>)





### 3.1 Create a PostgreSQL database

1. Start a shell (*cmd.exe*)
2. Navigate to your PostgreSQL installation path *\$postgres* (i.e. *C:\USERPATH\bin\pgsql*)
3. Go into the *bin* subdirectory (*cd bin*)
4. Create a data directory (*mkdir data*)
5. Start PostgreSQL *pg\_ctl.exe start -D "C:\USERPATH\bin\pgsql\data"*
6. Create the *mamodar* database *createdb.exe mamodar*

### 3.2 Set-Up

1. Start Visual Studio Code
2. Install two *Extensions* (fifth icon on the left bar): 2.1 *Gradle Tasks* (richardwillis.vscode-gradle) 2.1 *npm* (eg2.vscode-npm-script)
3. Update *server\application.properties* 3.1. Fill in *spring.datasource.username* and *spring.datasource.password* using the PostgreSQL data 3.2. Update *rdmo.token* and *rdmo.url* to point to a previously setup RDMO server
4. In *Source Control* (third icon on the left bar) 4.1. Clone repository: <https://github.com/cuehs/mamodar.git> 4.2. Select a local folder to clone code to (no server!) 4.3. Open folder in Visual Studio Code
5. Go to *Terminal*, Run task: *npm: build -web*



## CHAPTER 4

---

### Run project

---

1. Start database (if not started already) 1.1. Start a shell (*cmd.exe*) 1.2. Navigate to your PostgreSQL installation path (i.e. *C:\USERPATH\bin\pgsql*) 1.3. Start PostgreSQL *pg\_ctl.exe start -D "C:\USERPATH\bin\pgsql\data"*
2. Start backend 2.1. In Visual Studio Code: Got to Terminal in the Taskbar at the top 2.2. Run task: *gradle:server:bootRun*
3. Start frontend 3.1. In Visual Studio Code: Got to Terminal in the Taskbar at the top 3.2. Run task: *npm: start -web*
4. Open a browser at <http://localhost:4200/>



## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`