

## Variante 2: Lokale Installation

Voraussetzungen:

### Backend (Spring Boot):

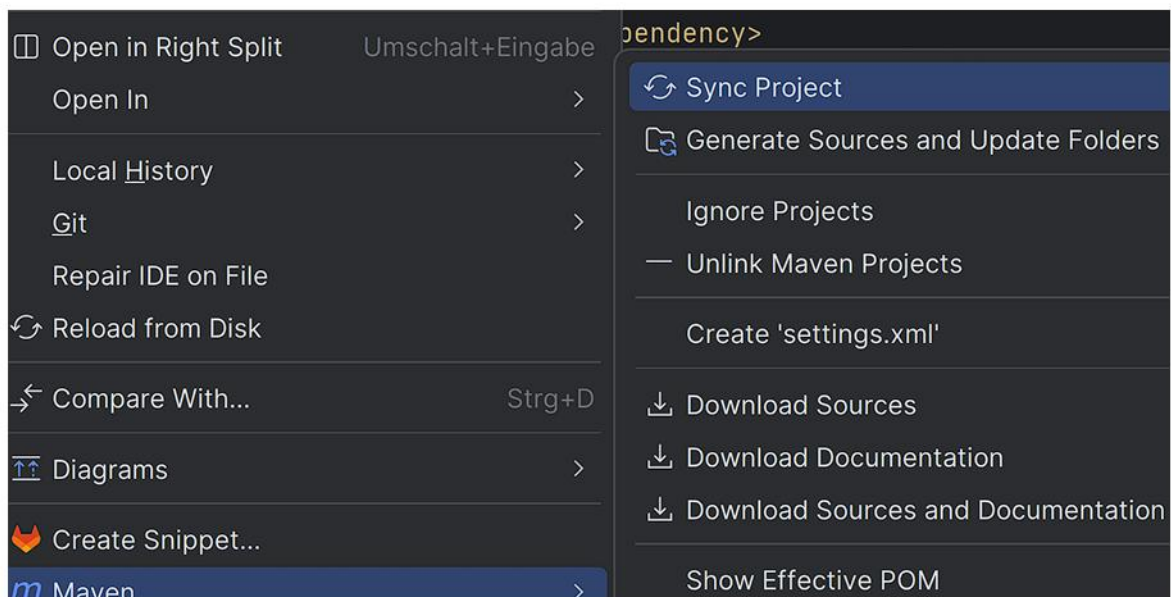
- **Java 21 JDK** - <https://adoptium.net/temurin/releases/?version=21>
- **Maven 3.8+** - <https://maven.apache.org/download.cgi>
- **PostgreSQL 15** - <https://www.postgresql.org/download/>
- **pgAdmin**: <https://www.pgadmin.org/download/>
- **Postman**: <https://www.postman.com/downloads/>
- **Eine IDE Ihrer Wahl (zum Beispiel IntelliJ)**: <https://www.jetbrains.com/idea/idea/download/?section=windows>

### Frontend (React):

- **Node.js 18+** - <https://nodejs.org/en/download/>
- **npm** (kommt mit Node.js)

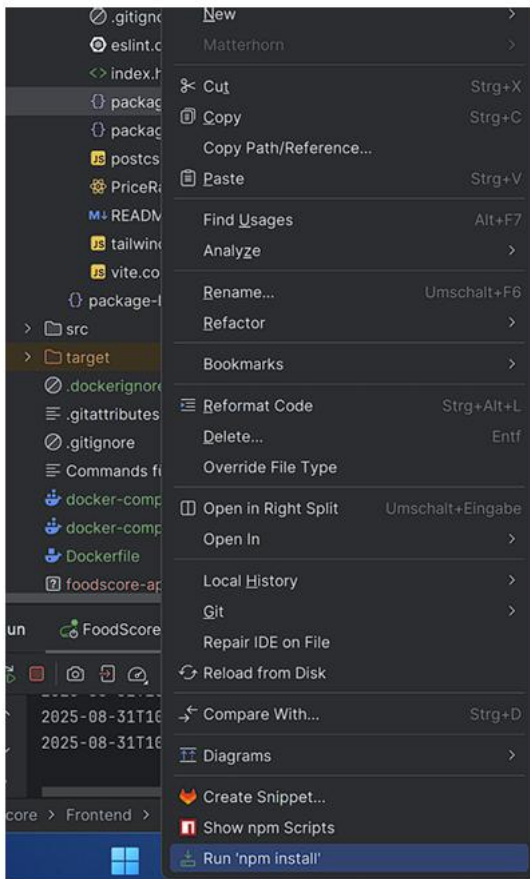
Entpacken Sie den Ordner „foodscore“ und öffnen Sie den Ordner mithilfe der IDE.

Navigieren Sie zu pom.xml und führen Sie mit Rechtsklick maven/sync\_project aus

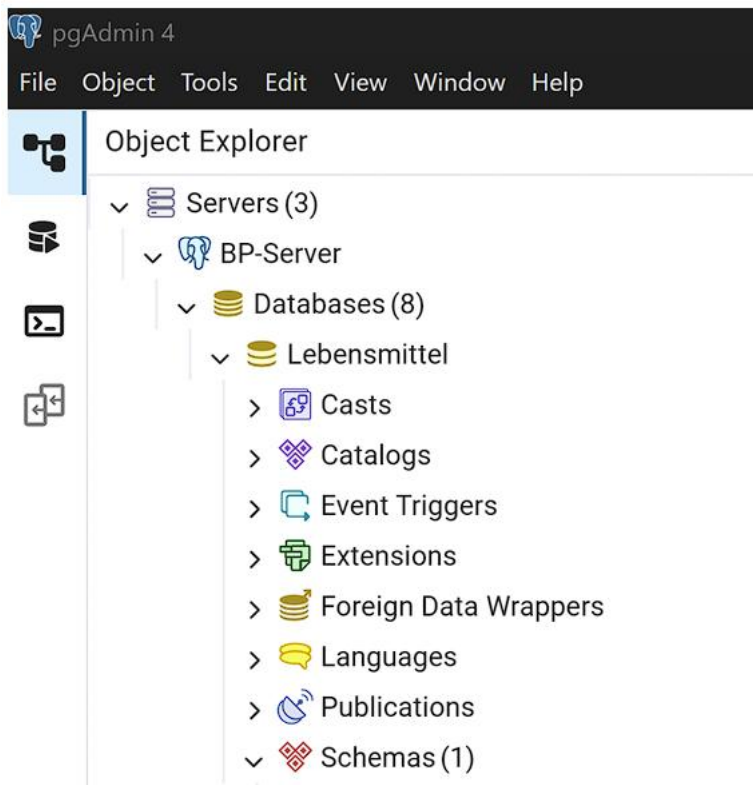


Dann zu Frontend/lebensmittel-score-frontend/package.json

Und führen Sie aus mit Rechtsklick „run npm install“ aus:



Jetzt müssen Sie in pgAdmin4 öffnen und einen neuen Server und Datenbank einspielen

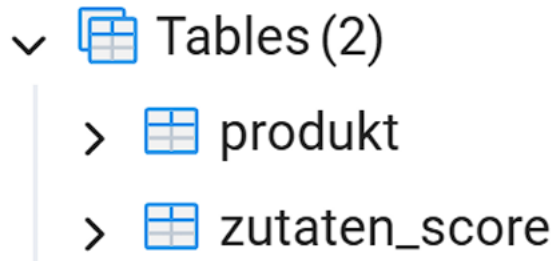


Dann wieder in der IDE die Datei: `src/main/resources/application.properties` So anpassen, dass die Datenbankverbindung steht z. B.

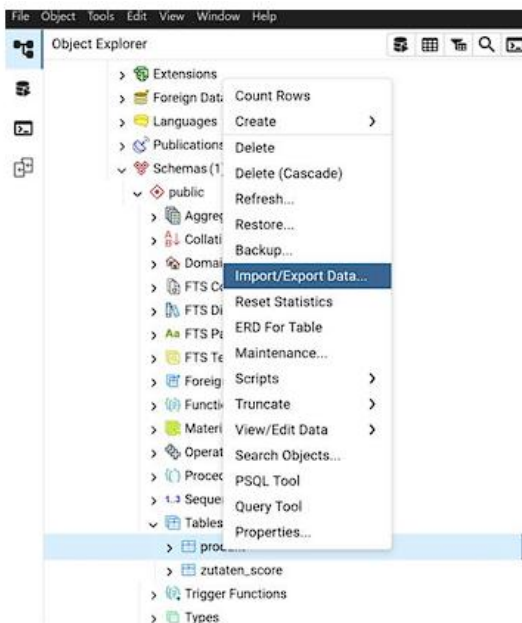
```
# PostgreSQL-Verbindung
spring.datasource.url=jdbc:postgresql://localhost:5432/LebensmittelV5
spring.datasource.username=postgres
spring.datasource.password=Gladbeck123
```

Dann einmal die Anwendung starten, indem man die Datei: `src/main/resources/application.properties` startet.

In pgAdmin4 sollte unter der Datenbank unterhalb von Schemas/public/tables 2 neue Tabellen: `produkt` und `zutaten_score` hinzugekommen sein



Mit Rechtsklick auf die Tabelle „produkt“ dann auf import/export klicken:



Eingaben wie folgt: Filename soll auf die Datei in

`src/main/java/com/example/Food Score/Persistenz/Daten/Lebensmitteldaten.csv` verweisen

Import/Export data - table 'produkt' ✕

General Options Columns

Import/Export

Filename

C:\Users\cenk-\Documents\foodscore\src\main\java\com\example\

Format

csv

Encoding

Select an item...

On Error

stop

Specifies how to behave when encountering an error converting a columns input value into its data type. An error\_action value of stop means fail the command, while ignore means discard the input row and continue with the next one. The default is stop. The ignore option is applicable only for COPY FROM when the FORMAT is text or csv.

Log Verbosity

default

Specify the amount of messages emitted by a COPY command: default or verbose. If verbose is specified, additional messages are emitted during processing. This is currently used in COPY FROM command when

Anschließend unter Options dann die Werte wie folgt auswählen:

Import/Export data - table 'produkt' ✕

General Options Columns

Header

☒

Freeze

☐

Requests copying the data with rows already frozen, just as they would be after running the VACUUM FREEZE command.

Delimiter

,

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Quote

"

Specifies the quoting character to be used when a data value is quoted. The default is double-quote. This must be a single one-byte character. This option is allowed only when using CSV format.

Escape

"

Specifies the character that should appear before a data character that

Dann im Abschnitt Columns dann die Spalten in dieser Reihenfolge angeben und auf ok klicken:

### Import/Export data - table 'produkt'



General Options Columns

Columns to import

name x marke x kategorie x zutaten x energie\_kj x  
 energie\_kcal x fett x gesaettigte\_fettsauren x  
 kohlenhydrate x zucker x eiweiss x salz x ean x  
 preis x quelle\_url x bild\_url x

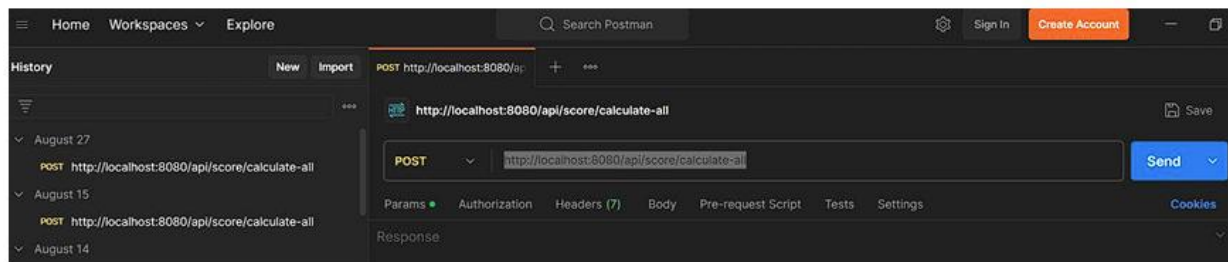
An optional list of columns to be copied. If no column list is specified, all

Die Anwendung kann nun gestartet werden, dazu muss in den Ordner Frontend/lebensmittel-score-frontend per Terminal navigiert werden und der Befehl „npm run dev“ ausführen

```
C:\Users\cenk-\Documents\foodscore\Frontend\lebensmittel-score-frontend>npm run dev
> lebensmittel-score-frontend@0.0.0 dev
> vite

VITE v7.0.0 ready in 792 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Jetzt müssen noch die Zutaten bewertet werden. Dazu muss per Postman eine POST Anfrage an diese Adresse gesendet werden: <http://localhost:8080/api/score/calculate-all>



Dann kann die Anwendung frei genutzt werden