

Rule Engine Application Documentation

Table of Contents

1. [Project Structure](#)
 2. [Backend README](#)
 3. [Frontend README](#)
 4. [Packaging](#)
 5. [Testing the Entire Application](#)
-

Project Structure

```
css
Copy code
ApplicationOne
├── Backend
│   ├── src
│   │   ├── main
│   │   │   ├── java
│   │   │   │   └── com
│   │   │   │       └── yourpackage
│   │   │   │           ├── controller
│   │   │   │           ├── service
│   │   │   │           └── model
│   │   │   └── resources
│   │   │       ├── application.properties
│   │   │       └── static
│   │   └── test
│   ├── pom.xml
│   └── README.md
└── Frontend
    ├── src
    │   ├── components
    │   │   ├── CreateRule.js
    │   │   └── EvaluateRule.js
    │   ├── App.js
    │   ├── App.css
    │   ├── index.js
    │   └── index.css
    ├── public
    │   └── index.html
    ├── package.json
    └── README.md
```

Backend README

Rule Engine Backend

Overview

This is the backend service for a Rule Engine application developed using Spring Boot. It allows for the creation and evaluation of rules based on user attributes.

Requirements

- Java 11 or higher
- Maven
- MySQL database

Setup

1. Clone the repository:

```
bash
Copy code
git clone <repository-url>
```

2. Navigate to the Backend directory:

```
bash
Copy code
cd ApplicationOne/Backend
```

3. Update application.properties with your MySQL database credentials:

```
properties
Copy code
spring.datasource.url=jdbc:mysql://localhost:3306/your_database_name
spring.datasource.username=your_username
spring.datasource.password=your_password
```

4. Build the application:

```
bash
Copy code
mvn clean install
```

5. Run the application:

```
bash
Copy code
mvn spring-boot:run
```

API Endpoints

POST /api/rules/create_rule

Creates a new rule.

Request Body:

json

Copy code

```
{
  "rule_string": "((age > 30 AND department = 'Sales') OR (age < 25 AND
department = 'Marketing')) AND (salary > 50000 OR experience > 5)"
}
```

Response:

json

Copy code

```
{
  "message": "Rule created successfully"
}
```

POST /api/rules/evaluate_rule

Evaluates a rule against user data.

Request Body:

json

Copy code

```
{
  "rule_string": "your_rule_string_here",
  "data": {
    "age": 35,
    "department": "Sales",
    "salary": 60000,
    "experience": 3
  }
}
```

Response:

json

Copy code

```
{
  "result": true
}
```

Testing

You can test the endpoints using Postman or any other API testing tool.

Frontend README

Rule Engine Frontend

Overview

This is the frontend application for the Rule Engine developed using React. It allows users to create and evaluate rules through a simple interface.

Requirements

- Node.js (v14 or higher)
- npm

Setup

1. Clone the repository:

```
bash
Copy code
git clone <repository-url>
```

2. Navigate to the Frontend directory:

```
bash
Copy code
cd ApplicationOne/Frontend
```

3. Install dependencies:

```
bash
Copy code
npm install
```

4. Start the development server:

```
bash
Copy code
npm start
```

5. Access the application: Open your browser and navigate to `http://localhost:3000`.

Features

- Create rules using a simple text area.
- Evaluate rules against user data and see the results.

Testing

Ensure that the backend server is running at `http://localhost:8080` before starting the frontend application.

Packaging

- **Backend:** Package your Spring Boot application using Maven. You can create a runnable JAR file by running:

```
bash
Copy code
mvn clean package
```

- **Frontend:** You can build your React app for production using:

```
bash
Copy code
npm run build
```

This will create an optimized build in the `build` folder.

Testing the Entire Application

After packaging both the frontend and backend:

1. **Run the backend server** to ensure it's working correctly with the MySQL database.
2. **Start the frontend application** and make sure you can access the features.
3. **Test the API calls** through the frontend to create and evaluate rules.