# AgendaAi

## Micro serviços usando Java 21 & Spring

## Dependências/Requisitos

* JDK 21
* Postgres
* Spring 3.3.0
  + Spring Web
  + PostgreSQL Driver
  + Spring Data JPA
  + Lombok
  + Validation

# Como rodar

## VSCode

Com jdk 21 instalado no sistema, instale as extensões [Spring Boot Extension Pack](https://marketplace.visualstudio.com/items?itemName=vmware.vscode-boot-dev-pack)

Então, instalando o [Extension Pack for Java](https://marketplace.visualstudio.com/items?itemName=vscjava.vscode-java-pack), é possível executar o projeto clicando no ícone com título *Run* no canto superior do ambiente de desenvolvimento. Caso não apareça, tente reiniciar ou selecionar o arquivo *main*, presente na raiz do pacote principal.

## Conexão com o Postgres

Dentro da pasta *resources* existe um arquivo chamado *application.properties.*

Configure ele para as informações presentes na sua máquina.

Informe seu nome de usuário do postgres e a senha para conectar com ele. Dentro do PGAdmin crie as tabelas com o nome *ms-customer*, *ms-company*, *ms-product*, *ms-schedule*. Não é necessário adicionar nenhuma coluna ou propriedade já que isso é feito de forma automática. Caso deseje mudar o nome da tabela, o novo nome deve ser identificado em *resources\nome\_do\_arquivo.properties*, ao final da url presente no campo *spring.datasource.url=*

spring.application.name=customer

server.port=8081

spring.datasource.url=jdbc:postgresql://localhost:5432/ms-customer

spring.datasource.username=postgres

spring.datasource.password=senha123

spring.jpa.hibernate.ddl-auto=update

# Testes

É possível integrar o backend a algum front atráves de linguagens e fazer requisições usando fetch, entretanto também é possivel usar o [Postman](https://www.postman.com/downloads/) (Baixe a versão para desktop para acessar localhost).

* Selecione o método HTTP desejado (Apenas Post, Get, Put, Delete são suportados atualmente).
* Informe a url "<http://localhost:808x/customers>"
* Clique em Body e vá em Raw
* Escreva um json baseado no que deseja inserir ou alterar.

Se tanto o cliente quanto o servidor estiverem rodando na mesma máquina, é possível que você enfrente erros de [*CORS*](https://pt.wikipedia.org/wiki/Cross-origin_resource_sharing). É possível contorná-los através de um novo tipo de arquivo [*configuration*](https://cursos.alura.com.br/forum/topico-cors-113963) e rodando o seu cliente em um servidor HTTP. Um jeito prático é fazer esse processo em Python:

cd E:\MSAGENDAAI\html\

python -m http.server 3000