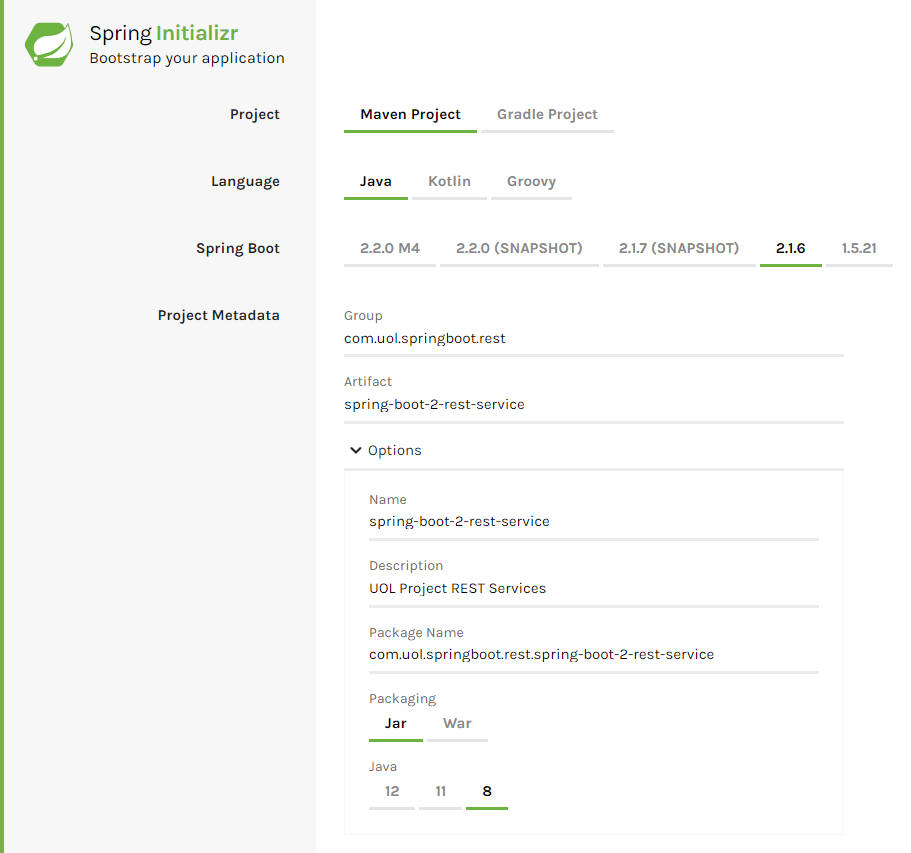
**Documentação UOL Project**

**1. Ferramentas Utilizadas**

1.1 – Eclipse: IDE utilizada para o desenvolvimento.

1.2 – Postman: Utilizado para fazer chamadas REST.

1.3 – Spring Initializr: Utilizado facilitar a geração do projeto Maven com Spring.



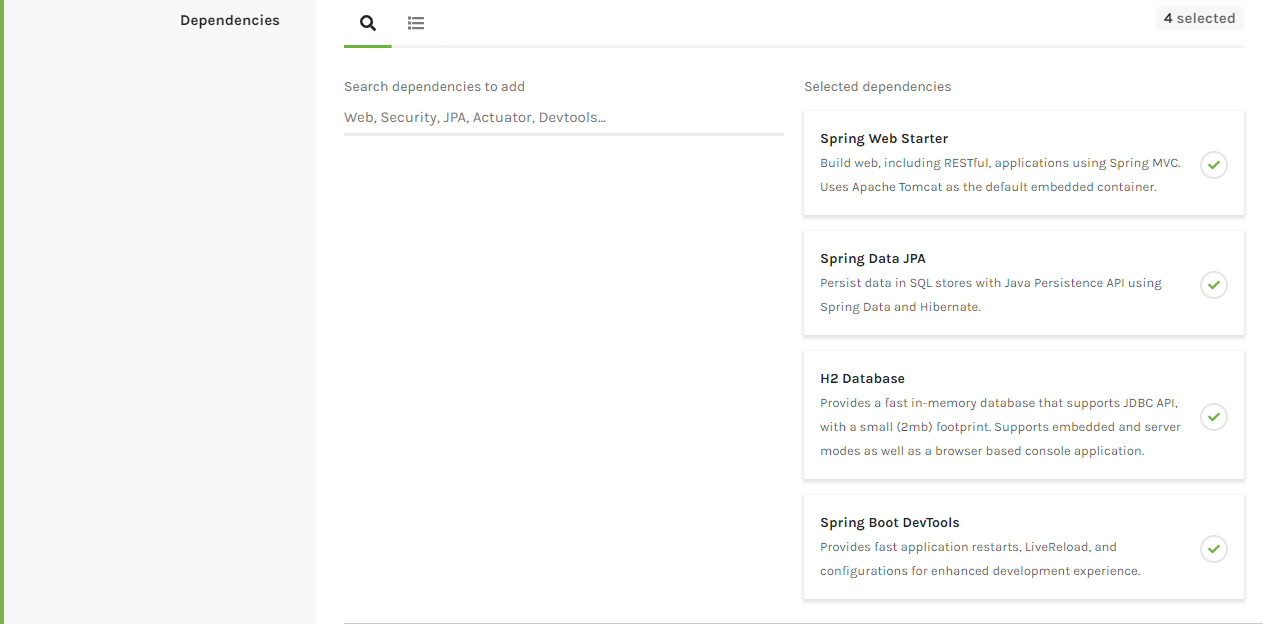
**2. Dependências**

2.1 – Spring Web Starter: Utilizado para construir aplicações RESTful.

2.2 – Spring Data JPA: Utilizado para persistir dados com o Java Persistance API usando Spring Data e Hibernate.

2.3 – H2 Database: Utilizado para prover um BD rápido, em memória e gerado em tempo de execução.

2.4 – Spring Boot DevTools: Utilizado para restarts e reloads rápidos na aplicação.



**3. Rodando e testando o projeto**

- Importe o projeto em sua IDE (Eclipse foi utilizado no desenvolvimento).

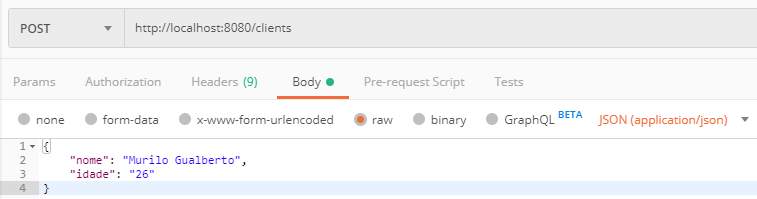
- No Package Explorer, vá até a classe principal em *com.uol.springboot.rest.springboot2restservice;* selecione a classe *SpringBoot2RestServiceApplication* com o botão direito e ‘Run As’ > ‘Java Application’.

- Abra o Postman e teste o GET chamando o endpoint <http://localhost:8080/clients>

- Mude a chamada para POST e mantenha o mesmo endpoint.

Na aba Body, selecione ‘raw’ e o tipo de envio como JSON. No corpo, digite:

*{ "nome": "Murilo Gualberto", "idade": "26" }*

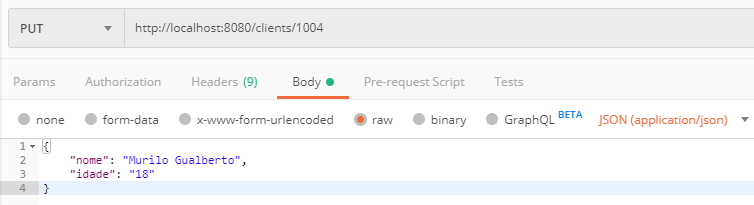


- Mude a chamada para GET chamando o endpoint <http://localhost:8080/clients>

- Mude a chamada para PUT, e mude o endpoint para <http://localhost:8080/clients/1004> .

Na aba Body, selecione ‘raw’ e o tipo de envio como JSON. No corpo, digite:

*{ "nome": "Murilo Gualberto", "idade": "18" }*



- Mude a chamada para GET chamando o endpoint <http://localhost:8080/clients/1004>

- Mude a chamada para DELETE chamando o endpoint <http://localhost:8080/clients/1004>

- Mude a chamada para GET chamando o endpoint <http://localhost:8080/clients>

- Volte para o Eclipse e veja no console as informações que foram salvas no cliente.

- Encerre a aplicação no Console.

**4. Empacotando e entregando o Projeto**

- Selecione com o botão direito a raiz do projeto (spring-boot-2-rest-service) com o botão direito, ‘Run as’ > ‘Run Configurations...’.

- Selecione com o botão direito ‘Maven Build’ > ‘New Configuration’.

- Em Name: coloque “*mvn clean install*”. Em baixo de base directory selecione o botão ‘Workspace’ e selecione o projeto ‘spring-boot-2-rest-service’ e clique ‘Ok’. Em ‘Goals’ coloque “*clean install*” (o Eclipse automaticamente usa o ‘mvn’).

- Clique em ‘Apply’ e depois em ‘Run’. Espere o Maven buildar o projeto e quando completar, o JAR estará disponível dentro da pasta ‘target’ dentro do projeto.