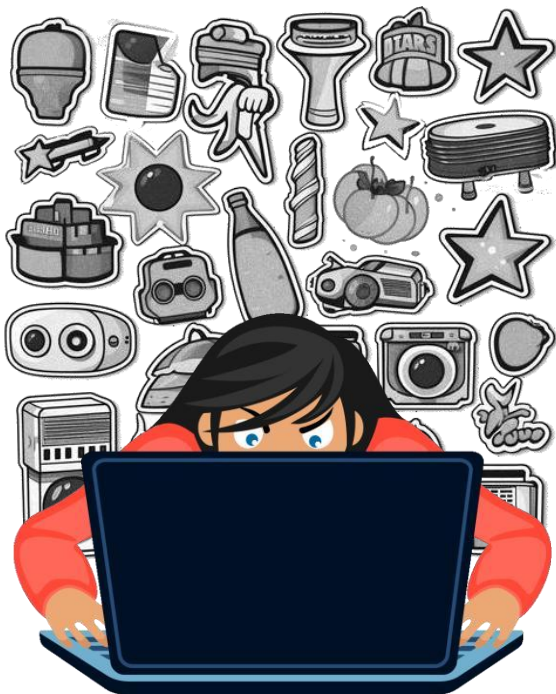




01

JSON

JSON



- O formato JSON (JavaScript Object Notation) é, como o nome sugere, uma forma de notação de objetos JavaScript, de modo que eles possam ser representados de uma forma comum a diversas linguagens.
- Facilmente trafegado entre aplicações em quaisquer protocolos, inclusive o HTTP.
- É um texto.

REPRESENTAÇÃO

{ }

```
{  
  "nome": "tênis",  
  "descricao": "tênis de corrida",  
  "quantidadeEstoque": 15,  
  "preco": 300  
}
```

VANTAGENS



CHECK LIST

Vantagens do JSON

- Leitura mais simples
- É tipado
- Arquivo com tamanho reduzido
- Velocidade maior na execução e transporte de dados
- Quem utiliza? Google, Facebook, Yahoo!, Twitter...



JSON

JSON – é um formato de dados leve e de fácil leitura utilizado para troca de informações entre sistemas computacionais.

Amplamente utilizado na web para representar dados estruturados.

- Oferece simplicidade, legibilidade, portabilidade e suporte amplo.
- JSON é muito utilizado na comunicação entre servidores e clientes em aplicações web, inclusive em APIs (Interface de Programação de Aplicativos), para transferir dados entre servidor e clientes de forma mais eficiente.

Manipulando arquivo JSON

Primeiro, adicione a dependência Gson ao seu projeto

Acesse o site:

<https://repo1.maven.org/maven2/com/google/code/gson/gson/>

Escolha a versão (2.13.1)

Baixe o arquivo gson-2.13.1.jar

OBS.: Deixei o arquivo no Teams e no github.

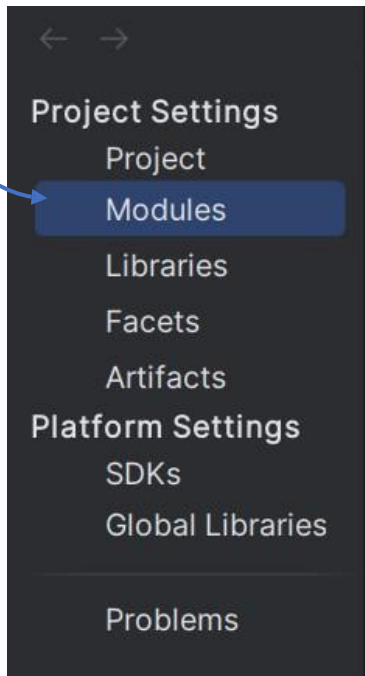
Agora precisamos adicionar a dependência no projeto.

1. Crie uma pasta lib no seu projeto.

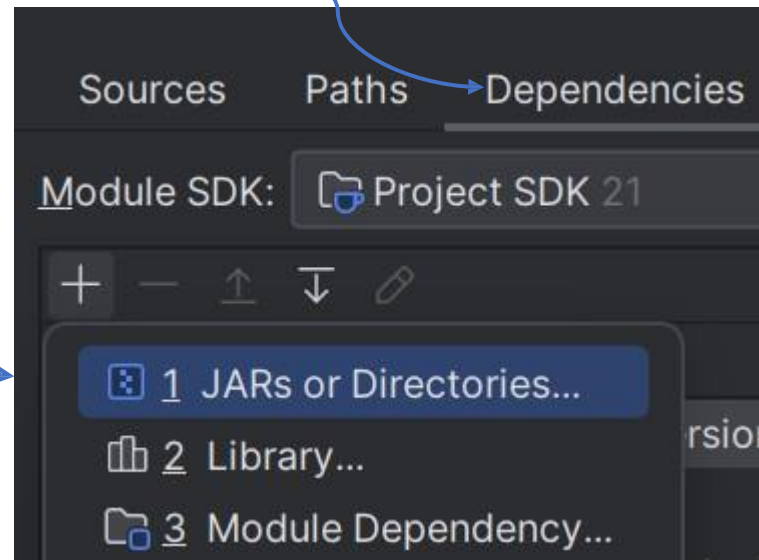
2. Precisamos adicionar o JAR que baixamos ao classpath no IntelliJ, para isso abra o menu File → Project Structure e siga os próximos passos.

Adicionando o JAR

Selecione a opção Modules



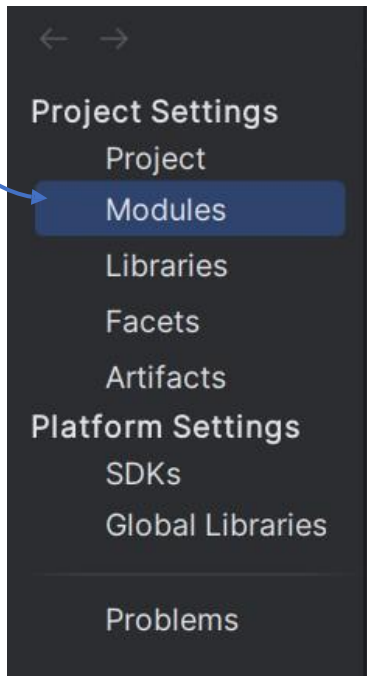
Clique no +
E em seguida na
opção 1 JARs



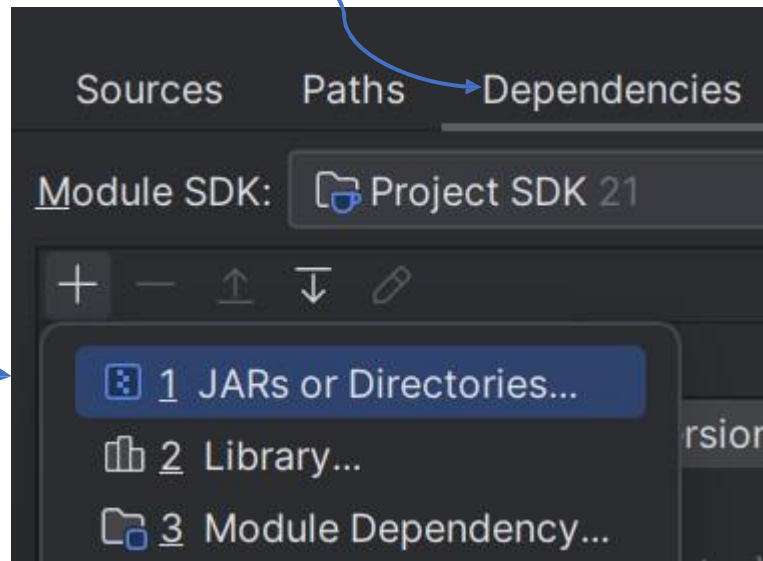
Clique na opção Dependencies

Adicionando o JAR

1. Selecione a opção Modules



2. Clique na opção Dependencies



3. Clique no +
E em seguida na
opção 1 JARs

4. Selecione o arquivo **gson-2.13.1.jar**.
Certifique-se que o Scope está como Compile.
Clique em OK para salvar.

ESCRITA DE JSON EM ARQUIVO

Usamos a biblioteca **Gson** para converter objetos Java em JSON e escrevê-los em arquivos. **setPrettyPrinting**: Gera JSON formatado de maneira legível.

```
public static void main(String[] args) {  
    Pessoa pessoa = new Pessoa("João", 30);  
    Gson gson = new GsonBuilder().setPrettyPrinting().create();
```



Gson: Facilita a conversão de objetos Java para JSON.

{ }

```
    try (FileWriter writer = new FileWriter("pessoa.json")) {  
        gson.toJson(pessoa, writer);  
        System.out.println("Arquivo JSON escrito com sucesso!");  
    } catch (IOException e) {  
        System.out.println("Ocorreu um erro ao escrever o arquivo  
JSON: " + e.getMessage());  
    }  
}
```

LEITURA DE JSON DE UM ARQUIVO

É possível ler um arquivo JSON e convertê-lo em um objeto Java.

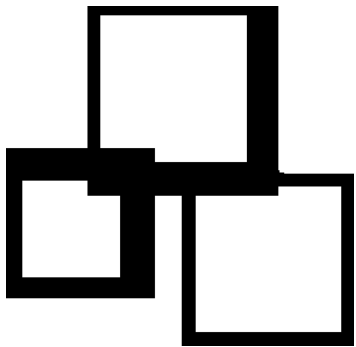
```
public static void main(String[] args) {  
    Gson gson = new Gson();  
  
    try (FileReader reader = new FileReader("pessoa.json")) {  
        Pessoa pessoa = gson.fromJson(reader, Pessoa.class);
```



fromJson: Converte Json em objeto Java

{ }

```
        System.out.println("Nome: " + pessoa.getNome());  
        System.out.println("Idade: " + pessoa.getIdade());  
    } catch (IOException e) {  
        System.out.println("Ocorreu um erro ao ler o arquivo  
JSON: " + e.getMessage());  
    }  
}
```



OBRIGADO

To be continued...

