***Criando uma aplicação para consumir endereço via CEP através de uma API:***

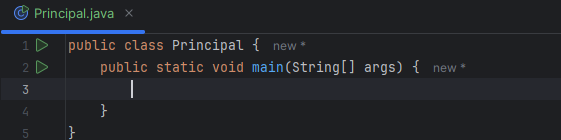
Objetivos:  
  
Criar uma aplicação para consultar a API ViaCep;

Menu para o usuário informar o CEP para busca;

Geração de um arquivo .Json com os dados do endereço.

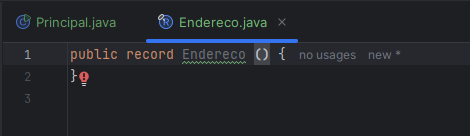
Passo a passo:

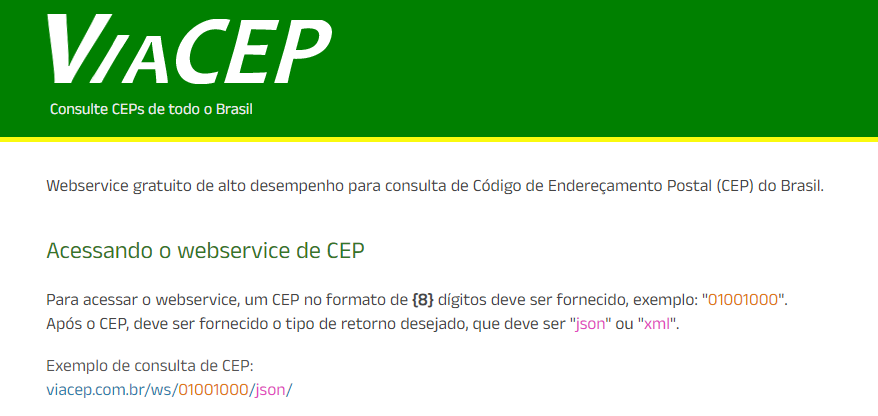
Primeiro passo: criar a classe principal onde vai receber o método main, que é o nosso ponto de entrada do nosso projeto e já chamei o método main.



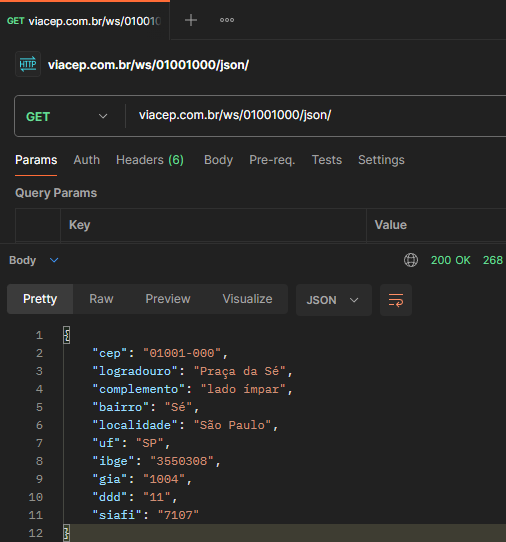
A intenção é buscar dados do endereço, logo preciso criar uma classe para representar o endereço. Como a intenção é representar e não setar valores, vou usar a classe record, usado para declarar classe imutável usadas para armazenar dados, assim ele já gera os métodos sem que eu precise setar.

Logo vou inserir uma nova classe e automaticamente ele vai entrar como public class, onde no lugar do class, vou alterar para record.

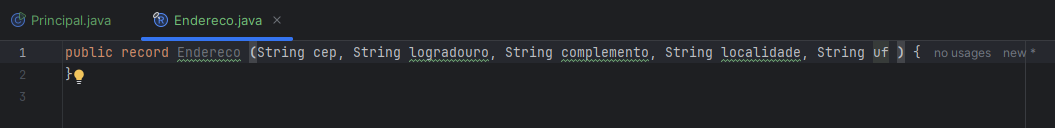


Consultei o site VIA CEP, que é um site pronto para consumir API/Webservice, peguei o link que eles oferecem como exemplo de consulta de CEP:  
  


E inseri no postman onde ele me retornou a consulta abaixo:



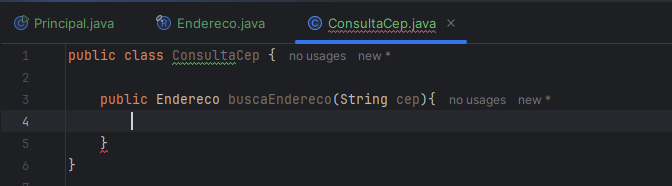
Com base nisso, eu começo a analisar quais os dados que quero para o meu buscador e acredito ser interessante.

Após essa análise, insiro meu dados evidenciando que são strings:  
  


Feito isso, meu record está pronto. Lembrando que ele já vem inserido os métodos.

A fim de deixar meu código mais organizado, criei uma nova classe ConsultaCep que vai ser o meu buscador de CEP.

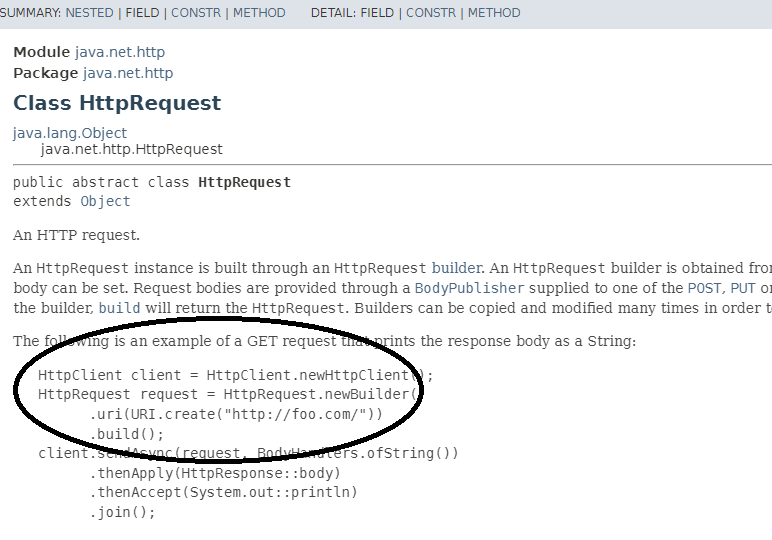
Pensando nisso, se torna necessário criar um método que eu insira um CEP e ele me devolva um endereço.



Criei o método public Endereco (endereço é a classe endereço e nomeei como buscaEndereco passando que ele vai buscar pelo CEP, ou seja, esse é o meu método para buscar o endereço (Buscar pelo CEP).

Nesse método nós vamos utilizar requisição e resposta, ou seja, eu vou fazer uma requisição através da busca pelo cep e vou ter uma resposta.

Para entender melhor o que estávamos fazendo e qual a forma de fazer, fomos consultar a documentação (java doc 17):  
  


No campo de busca digitei HTTPrequest, que nada mais é que uma requisição ao HTTP, ou seja, ao serviço web.:  
  


Nisso, ele me apresenta um código de exemplo e só preciso da primeira parte do código que é onde vou fazer a requisição ao serviço web e copio ele no meu código.

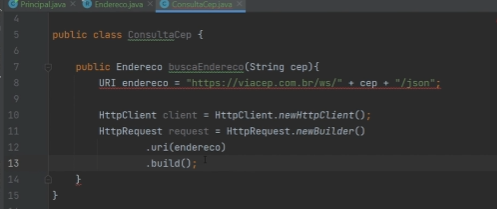


Vou fazer algumas alterações para deixar melhor para o que eu preciso.

A URI mencionada no código acima, serve para criar a interação com a WEB podendo ser uma URL também.

Em seguida eu preciso informar qual é a URI que estou usando para buscar o serviço na Web, que nada mais é que o endereço que está no Postman que foi inserido no passo mais acima.

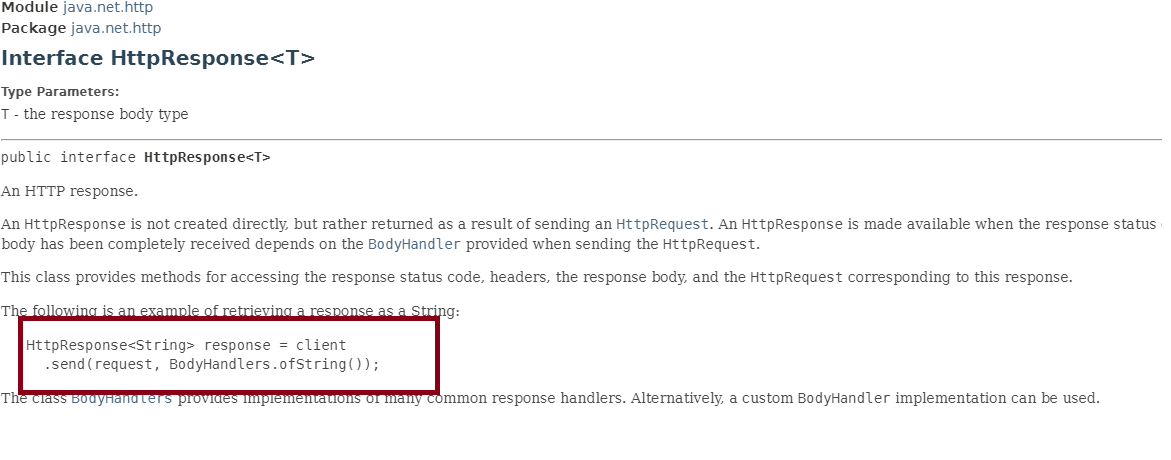
Note que extraímos a URI create, ou seja, digitei mais acima do HTTP, a URI e chamei de endereço e passei link do serviço web que estamos usando e concatenei com CEP.



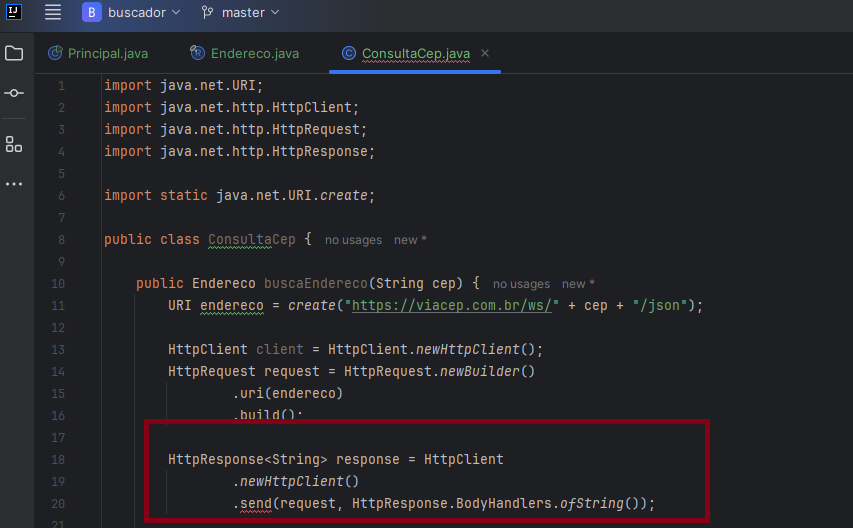
Obs.: meu link do serviço web estava assim:

[viacep.com.br/ws/01001000/json](http://viacep.com.br/ws/01001000/json)

Observe que tiramos os caracteres numéricos e deixamos o Json no final, que representa o formato de dados do arquivo, que é extremamente utilizado para troca de informações entre sistemas.

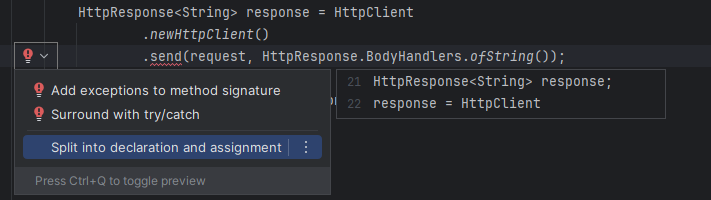
Assim como fiz a solicitação, preciso também de uma resposta. E faço o mesmo procedimento de verificar o que diz a documentação sobre o HTTPResponse:  
  


E ele me dá o código de exemplo de acima, que diz que eu posso ter uma resposta String através do meu client, que já foi inserido na solicitação:



Meu código está reclamando do send e se eu tenho verificar, percebo que ele está me falando das exceções:

Na primeira opção de Add exceptions to method signature, ele vai lançar as exceções pra mim e vai tratá-las:



Ele está reclamando do meu método send, que lançou exceções e ele está sugerindo que eu trate as exceções. Sempre quando isso acontece, eu preciso tratar as exceções ou declará-la e através disso o método vai lançar as minhas exceções.

Se eu usar a opção, ADD EXCEPTIONS TO METHOD SIGNATURE, automaticamente ele vai inserir a palavra chave TRHOWS ao lado do método, indicando que vai tratar as exceções, ou seja, o método vai tratar as exceções.

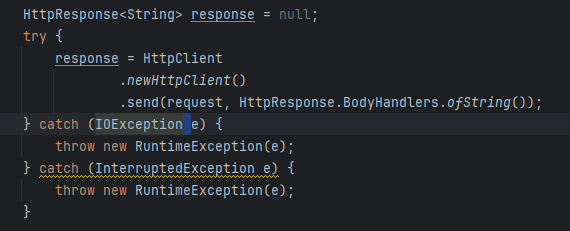


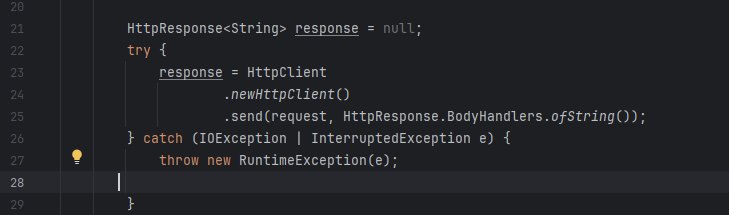
Mas eu quero tratar as exceções de forma mais personalizada, então vou usar a segunda opção de Try/cath.

Try: O bloco try indica o código que está gerando a exceção.

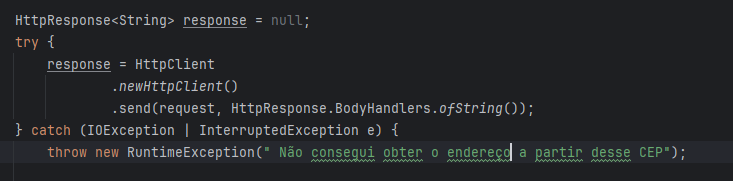
O bloco Catch é usado para tratar as exceções.

Quando clico nessa opção, automaticamente, existe uma modificação no meu código:



No primeiro cath eu posso concatenar as duas exceções (IOException e InterruptedException) e vai ficar da seguinte forma:  
  


Como concatenei as duas exceções, o cath da Linha 26, não preciso usar.

Caso ocorra algum problema na busca, coloquei uma resposta de que não consegui acessar o endereço…  
  


Mas caso ele consiga, ele já vai direto para a parte de conversão de dados com Gson.

IOException: é uma exceção que pode ser lançada durante operações de entrada e saída (Input/ Output, ou seja indica que ocorreu um erro durante essas operações.

InterruptedException: é uma exceção que pode ser lançada quando uma thread é interrompida, ou seja, um subprocesso do sistema.

Quando trato as exceções, eu consigo garantir que o sistema funcione corretamente.

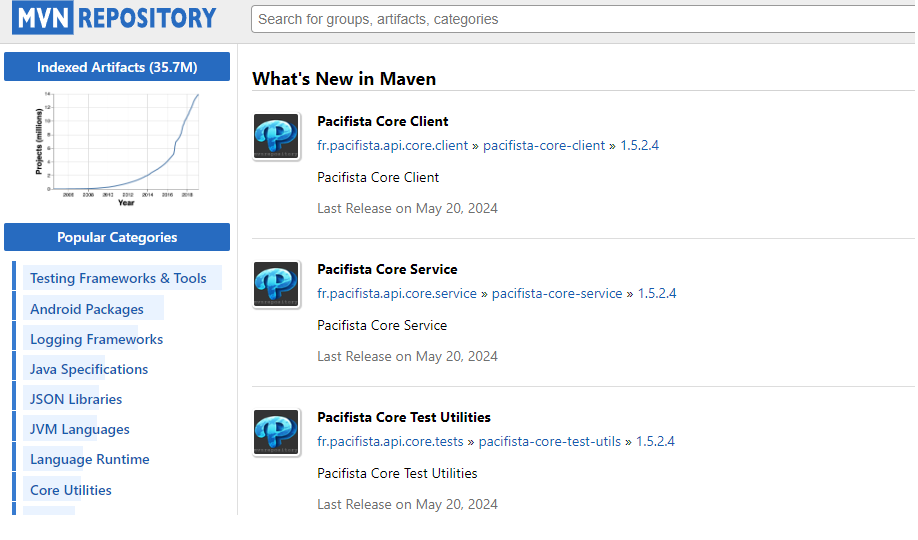
Nesse caso, eu garanto a integridade e funcionamento das minhas requisições HTTP e das minhas respostas.

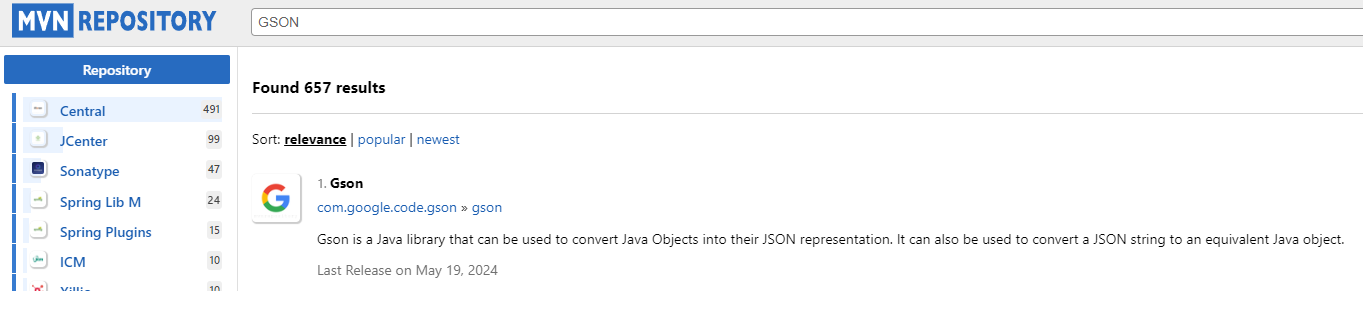
Na linha Throw new, utilizo essa expressão para lançar uma excessão explicitamente, ou seja, é usado para sinalizar que algo inesperado aconteceu e quando acontecer, lanço a exceção RunTimeException para dizer que o compilador não é obrigado a fazer o tratamento delas. E quando isso acontecer, a mensagem seguinte aparecerá: “Não consegui obter o endereço a partir desse CEP.”, permitindo que esse erro seja tratado isoladamente.

Desserealização:  
  
A minha resposta, vai vir em um formato Json, mas como eu quero manipular a minha resposta, eu preciso tranformar o meu formato de dados Json para um objeto.

Pensando nisso, preciso fazer a desserealização.

Foi usado o site <https://mvnrepository.com/> que é usado para buscar bibliotecas, pacotes, pois ele tem muitos repositórios úteis para o projeto.



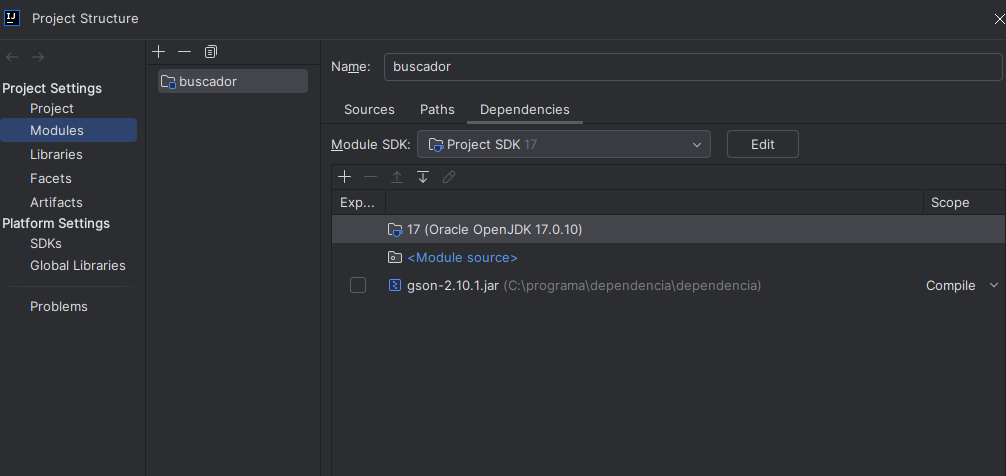
Nós vamos usar a biblioteca do google mesmo chamada GSON:  
  


Gson é uma biblioteca do google usada para serializalão e desserealização:  
  
**Serialização**: Converte objetos Java em JSON.

**Desserialização**: Converte JSON em objetos Java.

Precisa baixar a biblioteca, para isso vou selecionar a biblioteca, clicar na versão (geralmente é a última), e selecionar o JAR. (no caso é para o java).

Após baixado, precisamos inserir a biblioteca no projeto. Para isso, voltamos ao Intellij, File, Project Structure:

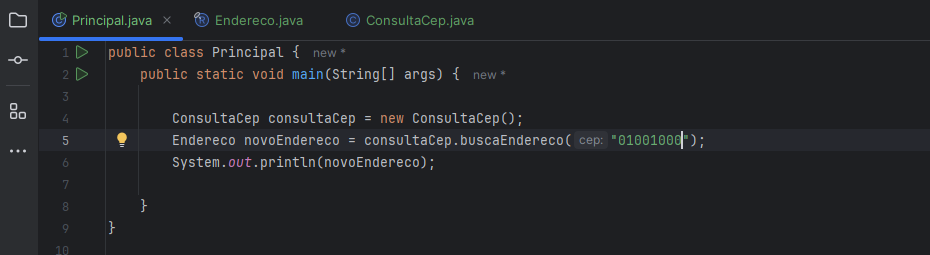


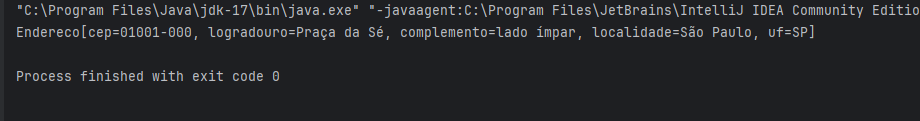
Na tela do Project Structure, vou clicar em Modules, em seguida Dependências e clicar no sinal de +.

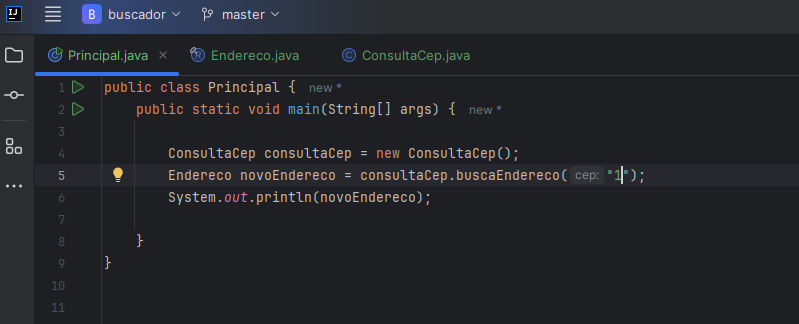
Vou localizar a pasta que salvei a minha biblioteca e clicar em OK. Pronto, minha dependência foi adicionada, ou seja, o meu projeto depende da biblioteca Gson para funcionar corretamente.

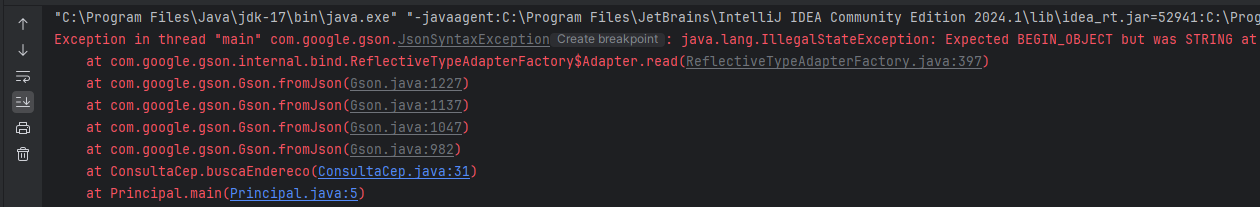
Agora que montei meu sistema, com requisições e exceções tratadas, preciso testar.

E para isso, vou usar a minha classe principal e tentar fazer uma busca.

Para isso vou instanciar a minha classe ConsultaCep e vou inserir um CEP que tinha de exemplo na minha consulta da API no Postman:  
  


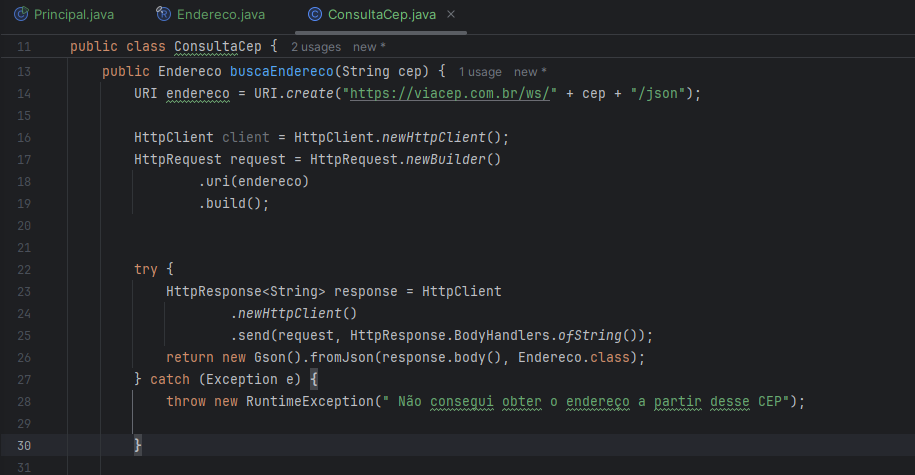
Na minha resposta no terminal, vai vir o endereço certinho:  
  


Porém se eu coloco um cep errado:  
  


ele traz uma outra exceção:  
  


Nesse caso, ele me indica que o erro está na classe Consulta Cep, na linha 31:  
  


Isso indica que se os meus dados que serão desserealizados estiver fora do bloco correto não vai funcionar, então, eu preciso inserir a linha 31 dentro do try, ou seja, dentro das exceções:



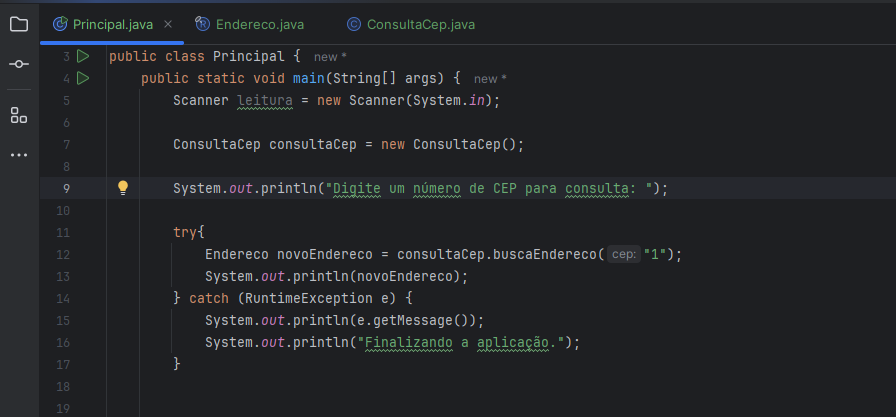
Note que foi realizado mudanças no catch, não para tratar apenas as exceções que tinha tratado e sim todas!

Nesse caso vou tratar as exceções na classe principal que se a parecer a RuntimeExceoption vou exibir a mensagem e finalizar a aplicação.

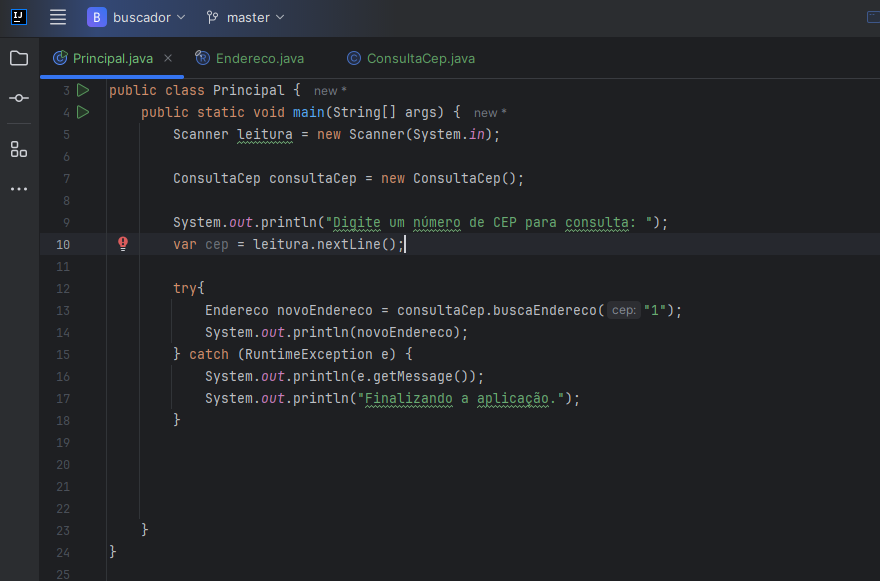


Após o teste finalizado, preciso criar o Scanner e solicitar o cep para o usuário.



Digitei na linha 9 a parte de leitura:  
  


E vou ler o que o usuário digitar e armazenar numa variável, conforme inserido em linha 10, usei o nextLine por que ela vai me retornar uma string:

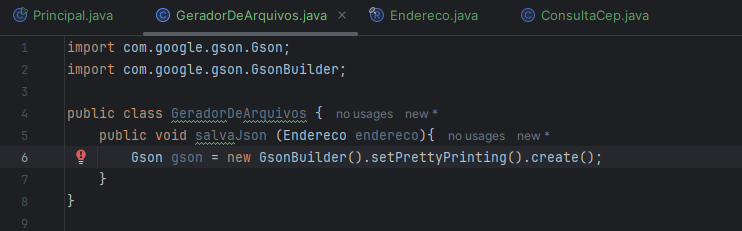


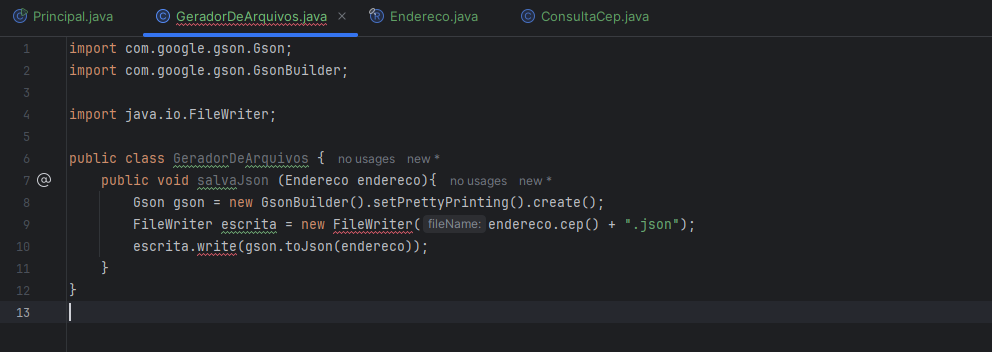
Feito isso, vou fazer um gerador de arquivos que vai ser uma classe responsável por criar arquivos em um formato especifico, como o Json e a finalidade é criar um arquivo com os dados consultados a fim de armazenar e compartilhar esses dados.

Para isso vou criar uma nova classe chamada Gerador de Arquivos e inserir o método void por que ela não vai ter retorno nenhum, apenas gerar arquivo.

No passo abaixo, criei o public void e chamei ele de salvaJson que vai pegar os dados da classe Endereco.

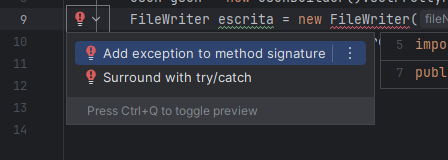
O Gson vai receber o Gson Builder (projeto padrão):



Em seguida, vou criar um objeto do tipo File Writer que vai ser responsável por escrever no arquivo (linha 10):  
  


e na linha 11 estou pedindo para a biblioteca Gson converter o objeto endereço em um formato Json usando o método toJson.

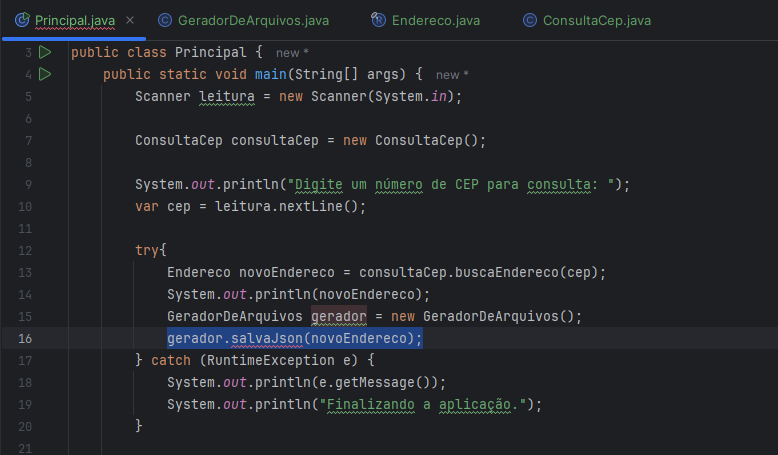
Perceba que ele está dando um errinho no meu FileWriter, por que ele está gerando exceções, e novamente tenho as duas opções:

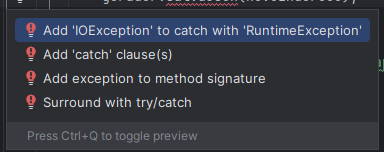


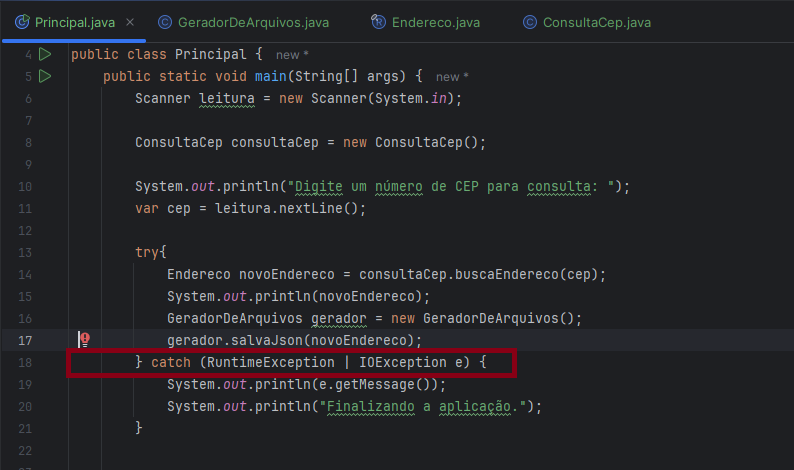
Nesse caso, optei pelo throw porque quero apenas sinalizar se ocorrer algo inesperado.

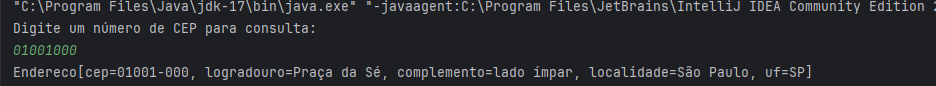
Para gerar o gerador de arquivos, preciso instanciar na classe principal também (linha 15).

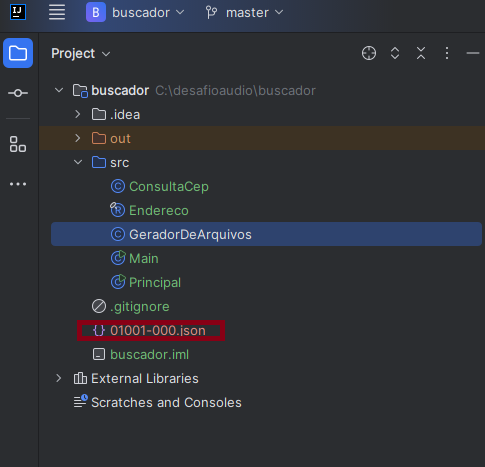
Inseri a linha 16 também para acionar a linha do método salvaJson para criar o arquivo Json com os dados do endereço consultado.



Meu código está reclamando do método salvaJson:  
  


E novamente ele me falando das exceções. Cliquei na primeira alternativa, já para ele incluir no cath e ele já concatenar as exceções:  
  


Feito tudo isso, vamos rodar o código e verificar se no terminal está tudo ok:  
  


e sim, está Ok. Porém a intenção é gerar o arquivo, então vamos ver no menu lateral se meu arquivo Json está lá.:  
  


e sim ele está.

Quando clico duas vezes, ele abre meu arquivo Json, com os campos que eu solicitei:  
  
