

Exercício: consumindo uma API WEB

Embora tenhamos definido os termos “serviço” e “API”, é comum encontramos os dois termos sendo usados como sinônimos.

O “serviço” chamado ViaCEP (www.viacep.com.br) oferece duas funcionalidades:

- Validação de CEP: neste envia-se um CEP Brasileiro e o serviço retorna um JSON com o endereço correspondente.
- Busca de CEP: envia-se a informação de estado, cidade e logradouro, nesta ordem, e o serviço responde com um JSON contendo uma lista de CEPs/endereços que atendem aos dados informados.

Objetivo da tarefa

Criar um programa Java que apresenta duas opções para o usuário: validar um CEP ou pesquisar um CEP. Na primeira opção o programa solicita um CEP e imprime na tela o endereço correspondente ao CEP se ele for valido ou uma mensagem de erro caso seja um CEP inexistente. Na segunda opção o programa deve solicitar a sigla do estado, o nome da cidade e o logradouro e imprimir a lista de CEPs/endereços correspondentes aos dados informados ou uma mensagem de erro caso os dados não correspondam a nenhum CEP.

Para verificar e/ou pesquisar os CEPs deve ser usado o serviço “ViaCEP” descrito acima, isto é, o programa a ser desenvolvido será um “cliente” deste serviço WEB.

Para a comunicação com o serviço o programa deverá explorar, obrigatoriamente, as classes *HttpRequest* e *HttpResponse* da API de Java. O Retorno do serviço é uma string JSON que pode ser exibida na tela sem tratamento.

Detalhes sobre o protocolo REST e formas de tratar as strings JSON serão objeto das próximas aulas.

A figura que segue apresenta um exemplo de código Java que faz uma requisição HTTP.

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.time.Duration;

try {
    HttpClient httpClient = HttpClient.newBuilder()
        .connectTimeout(Duration.of(1, MINUTES))
        .build();

    httpRequest = HttpRequest.newBuilder()
        .GET()
        .uri(URI.create(url))
        .build();
```

```
    httpResponse = httpClient.send(httpRequest, HttpResponse.BodyHandlers.ofString());

    System.out.println(httpResponse.body());
} catch (IOException e) {
    e.printStackTrace();
    throw new RuntimeException(e.getMessage());
} catch (InterruptedException e) {
    e.printStackTrace();
    throw new RuntimeException(e.getMessage());
}
```

Exemplo de código Java que faz uma requisição HTTP.