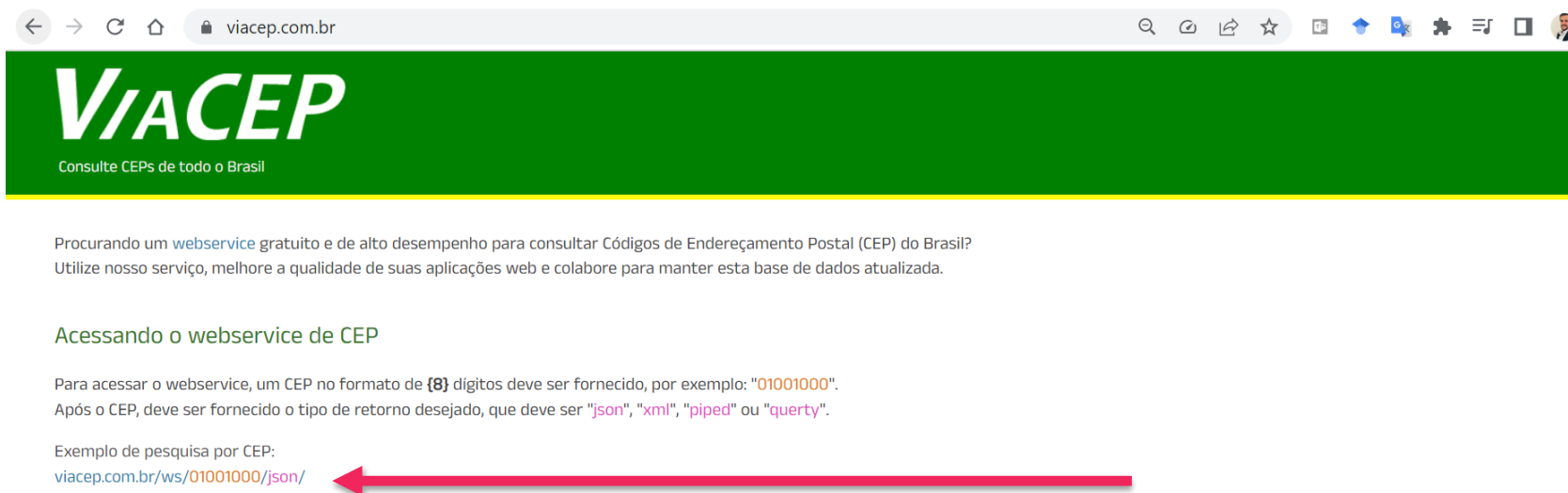


FIAP GRADUAÇÃO

# API consumindo CEP

# CEP dos correios

No site [www.viacep.com.br](http://www.viacep.com.br) tem as instruções de como enviar um cep e receber as informações, iremos focar no que está sendo apontado em vermelho na imagem.



Procurando um [webservice](#) gratuito e de alto desempenho para consultar Códigos de Endereçamento Postal (CEP) do Brasil? Utilize nosso serviço, melhore a qualidade de suas aplicações web e colabore para manter esta base de dados atualizada.

## Acessando o webservice de CEP

Para acessar o webservice, um CEP no formato de **{8}** dígitos deve ser fornecido, por exemplo: "01001000". Após o CEP, deve ser fornecido o tipo de retorno desejado, que deve ser "json", "xml", "piped" ou "querty".

Exemplo de pesquisa por CEP:

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

## Validação do CEP

Quando consultado um CEP de formato inválido, por exemplo: "950100100" (9 dígitos), "95010A10" (alfanumérico), "95010 10" (espaço), o código de retorno da consulta será um **400** (Bad Request). Antes de acessar o webservice, valide o formato do CEP e certifique-se que o mesmo possua **{8}** dígitos. Exemplo de como validar o formato do CEP em javascript está disponível nos exemplos abaixo.

Quando consultado um CEP de formato válido, porém inexistente, por exemplo: "99999999", o retorno conterá um valor de "erro" igual a "true". Isso significa que o CEP consultado não foi encontrado na base de dados. Veja como manipular este "erro" em javascript nos exemplos abaixo.

## Formatos de Retorno

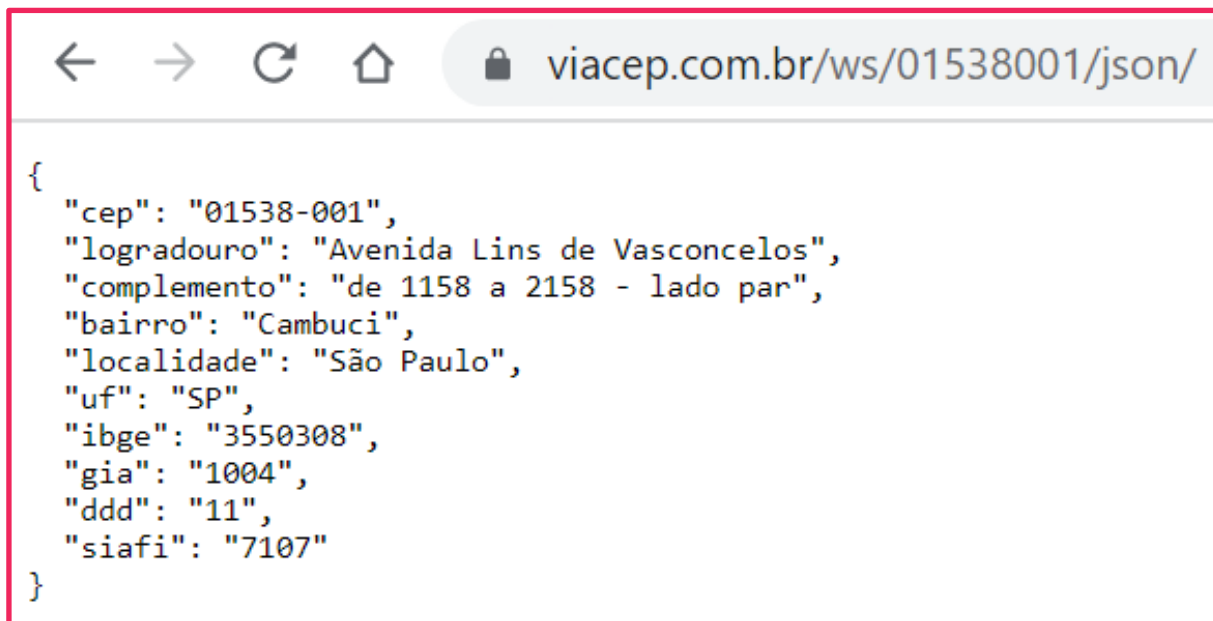
Veja exemplos de acesso ao webservice e os diferentes tipos de retorno:

## CEP dos correios

Para consultar um cep basta entrar no site com o link abaixo e substituir o que está em vermelho pelo CEP que deseja consultar o endereço.

<https://viacep.com.br/ws/01538001/json/>

- Após entrar no endereço abaixo o que o site exibiu de endereço, nesse caso é o endereço da FIAP da Aclimação e o CEP é 01538-001



```
{
  "cep": "01538-001",
  "logradouro": "Avenida Lins de Vasconcelos",
  "complemento": "de 1158 a 2158 - lado par",
  "bairro": "Cambuci",
  "localidade": "São Paulo",
  "uf": "SP",
  "ibge": "3550308",
  "gia": "1004",
  "ddd": "11",
  "siafi": "7107"
}
```

# Configurações do arquivo pom.xml – Parte 1

- <https://mvnrepository.com/open-source/http-clients>

– Esse link que é do maven repository disponibiliza configurações de dependências para os projetos que são feitas no pom.xml do maven.

- pom.xml

```
maven/pom.xml x
5 <artifactId>maven</artifactId>
6 <version>0.0.1-SNAPSHOT</version>
7
8 <name>maven</name>
9 <description>Add project description here</description>
10
11 <properties>
12   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
13 </properties>
14
15 <dependencies>
16   <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
17   <dependency>
18     <groupId>org.apache.httpcomponents</groupId>
19     <artifactId>httpclient</artifactId>
20     <version>4.5.14</version>
21   </dependency>
22
23   <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
24   <dependency>
25     <groupId>com.google.code.gson</groupId>
26     <artifactId>gson</artifactId>
27     <version>2.10.1</version>
28   </dependency>
29 </dependencies>
30
31 <build>
32   <plugins>
33     <plugin>
34       <groupId>org.apache.maven.plugins</groupId>
35       <artifactId>maven-compiler-plugin</artifactId>
36       <version>3.6.0</version>
37       <configuration>
38         <source>1.8</source>
39         <target>1.8</target>
40       </configuration>
41     </plugin>
42   </plugins>
43 </build>
```

Dentro de `<dependencies>`  
`</dependencies>` são as duas  
configurações que iremos inserir  
e que iremos aprender utilizando  
os dois próximos slides.

# Configurações do arquivo pom.xml – Parte 2

- <https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient/4.5.14>
  - Dependência http

**MVN REPOSITORY** Search for groups, artifacts, categories

Home » org.apache.httpcomponents » httpclient » 4.5.14

**Apache HttpClient » 4.5.14**  
HTTPClient provides an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.

License	Apache 2.0
Categories	HTTP Clients
Tags	network apache client http
HomePage	<a href="http://hc.apache.org/httpcomponents-client-ga">http://hc.apache.org/httpcomponents-client-ga</a>
Date	Nov 30, 2022
Files	pom (6 KB) jar (767 KB) View All
Repositories	Central Terrestis
Ranking	#36 in MvnRepository (See Top Artifacts) #1 in HTTP Clients
Used By	13,440 artifacts
Vulnerabilities	Vulnerabilities from dependencies: CVE-2020-15250

Testing Frameworks & Tools  
Android Packages  
Logging Frameworks  
Java Specifications  
JSON Libraries  
Core Utilities  
JVM Languages  
Mocking  
Language Runtime  
Web Assets  
Annotation Libraries  
Logging Bridges  
HTTP Clients  
Dependency Injection  
XML Processing  
Web Frameworks

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpclient</artifactId>
  <version>4.5.14</version>
</dependency>
```

☒ Include comment with link to declaration

```
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpclient</artifactId>
  <version>4.5.14</version>
</dependency>
```

# Configurações do arquivo pom.xml – Parte 3

- <https://mvnrepository.com/artifact/com.google.code.gson/gson/2.10.1>
  - Dependência para converter json em texto:

**MVN REPOSITORY** Search for groups, artifacts, categories Search

Home » com.google.code.gson » gson » 2.10.1

**Gson » 2.10.1**

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object.

License	Apache 2.0
Categories	JSON Libraries
Tags	google json gson format
Date	Jan 06, 2023
Files	pom (9 KB) jar (276 KB) View All
Repositories	Central
Ranking	#13 in MvnRepository (See Top Artifacts) #2 in JSON Libraries
Used By	19,675 artifacts

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

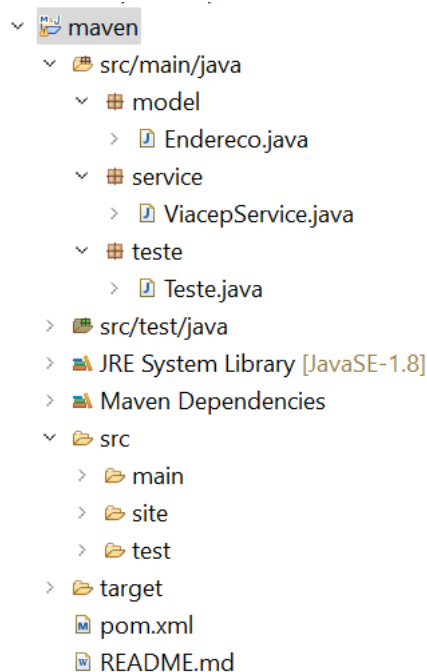
```
<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.10.1</version>
</dependency>
```

☒ Include comment with link to declaration

```
<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.10.1</version>
</dependency>
```

## Projeto para consultar um CEP

Criar um projeto Maven, conforme o professor ensinou, após criar a estrutura abaixo:





## Projeto para consultar um CEP – Arquivo pom.xml

Iremos adicionar as configurações no arquivo pom.xml que estão identificadas abaixo e foram explicadas nos slides anteriores.



```
maven/pom.xml x
5 <artifactId>maven</artifactId>
6 <version>0.0.1-SNAPSHOT</version>
7
8 <name>maven</name>
9 <description>Add project description here</description>
10
11 <properties>
12   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
13 </properties>
14
15 <dependencies>
16   <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
17   <dependency>
18     <groupId>org.apache.httpcomponents</groupId>
19     <artifactId>httpclient</artifactId>
20     <version>4.5.14</version>
21   </dependency>
22
23   <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
24   <dependency>
25     <groupId>com.google.code.gson</groupId>
26     <artifactId>gson</artifactId>
27     <version>2.10.1</version>
28   </dependency>
29 </dependencies>
30
31
32 <build>
33   <plugins>
34     <plugin>
35       <groupId>org.apache.maven.plugins</groupId>
36       <artifactId>maven-compiler-plugin</artifactId>
37       <version>3.6.0</version>
38       <configuration>
39         <source>1.8</source>
40         <target>1.8</target>
41       </configuration>
42     </plugin>
43   </plugins>
44 </build>
45 </pom.xml>
```

The image shows a screenshot of a pom.xml file in an IDE. Two red brackets with numbers 1 and 2 are used to highlight specific dependency entries. Bracket 1 points to the first dependency entry for org.apache.httpcomponents/httpclient. Bracket 2 points to the second dependency entry for com.google.code.gson/gson. A red line connects these two brackets to the explanatory text on the right.

Dentro de <dependencies>  
</dependencies> são as duas  
configurações que iremos inserir  
e que iremos aprender utilizando  
os dois próximos slides.

1 <https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient/4.5.14>

2 <https://mvnrepository.com/artifact/com.google.code.gson/gson/2.10.1>

# Projeto para consultar um CEP – Arquivo pom.xml

## Código do arquivo pom.xml:

```
<?xml version="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>br.com.cep</groupId>
<artifactId>maven</artifactId>
<version>0.0.1-SNAPSHOT</version>

<name>maven</name>
<description>Add project description here</description>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
<dependency>
<groupId>org.apache.httpcomponents</groupId>
<artifactId>httpclient</artifactId>
<version>4.5.14</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
<groupId>com.google.code.gson</groupId>
<artifactId>gson</artifactId>
<version>2.10.1</version>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.6.0</version>
<configuration>
<source>1.8</source>
<target>1.8</target>
</configuration>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-source-plugin</artifactId>
<version>3.0.1</version>
<executions>
<execution>
<id>attach-sources</id>
<goals>
<goal>jar</goal>
</goal>
</executions>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-javadoc-plugin</artifactId>
<version>3.0.1</version>
<executions>
<execution>
<id>attach-javadocs</id>
<goals>
<goal>jar</goal>
</goal>
</executions>
</plugin>
</plugins>
</build>

<reporting>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-project-info-reports-plugin</artifactId>
<version>3.0.1</version>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-javadoc-plugin</artifactId>
<version>3.0.1</version>
</plugin>
</plugins>
</reporting>
</project>
```

# Projeto para consultar um CEP – Arquivo Endereco.class

## Código:

```
package model;

public class Endereco {

    private String cep;
    private String logradouro;
    private String complemento;
    private String bairro;
    private String localidade;
    private String uf;
    private String ibge;
    private String gia;
    private String ddd;
    private String siafi;

    public String getCep() {
        return cep;
    }

    public void setCep(String cep) {
        this.cep = cep;
    }

    public String getLogradouro() {
        return logradouro;
    }

    public void setLogradouro(String logradouro) {
        this.logradouro = logradouro;
    }

    public String getComplemento() {
        return complemento;
    }

    public void setComplemento(String complemento) {
        this.complemento = complemento;
    }

    public String getBairro() {
        return bairro;
    }

    public void setBairro(String bairro) {
        this.bairro = bairro;
    }

    public String getLocalidade() {
        return localidade;
    }

    public void setLocalidade(String localidade) {
        this.localidade = localidade;
    }

    public String getUf() {
        return uf;
    }

    public void setUf(String uf) {
        this.uf = uf;
    }

    public String getIbge() {
        return ibge;
    }

    public void setIbge(String ibge) {
        this.ibge = ibge;
    }

    public String getGia() {
        return gia;
    }

    public void setGia(String gia) {
        this.gia = gia;
    }

    public String getDdd() {
        return ddd;
    }

    public void setDdd(String ddd) {
        this.ddd = ddd;
    }

    public String getSiafi() {
        return siafi;
    }

    public void setSiafi(String siafi) {
        this.siafi = siafi;
    }

    @Override
    public String toString() {
        return "Endereco [cep=" + cep + ", logradouro=" + logradouro + ", complemento=" + complemento + ", bairro="
            + bairro + ", localidade=" + localidade + ", uf=" + uf + ", ibge=" + ibge + ", gia=" + gia + ", ddd="
            + ddd + ", siafi=" + siafi + "];";
    }
}
```

## Imagem da classe:

```
Endereco.java ×
1 package model;
2
3 public class Endereco {
4
5     private String cep;
6     private String logradouro;
7     private String complemento;
8     private String bairro;
9     private String localidade;
10    private String uf;
11    private String ibge;
12    private String gia;
13    private String ddd;
14    private String siafi;
15
16    public String getCep() {
17        return cep;
18    }
19
20    public void setCep(String cep) {
21        this.cep = cep;
22    }
23
24    public String getLogradouro() {
25        return logradouro;
26    }
27
28    public void setLogradouro(String logradouro) {
29        this.logradouro = logradouro;
30    }
31
32    public String getComplemento() {
33        return complemento;
34    }
35
36    public void setComplemento(String complemento) {
37        this.complemento = complemento;
38    }
39
40    public String getBairro() {
41        return bairro;
42    }
43
44    public void setBairro(String bairro) {
45        this.bairro = bairro;
46    }
47
48    public String getLocalidade() {
49        return localidade;
50    }
51
52    public void setLocalidade(String localidade) {
53        this.localidade = localidade;
54    }
55
56    public String getUf() {
57        return uf;
58    }
59
60    public void setUf(String uf) {
61        this.uf = uf;
62    }
63
64    public String getIbge() {
65        return ibge;
66    }
67
68    public void setIbge(String ibge) {
69        this.ibge = ibge;
70    }
71
72    public String getGia() {
73        return gia;
74    }
75
76    public void setGia(String gia) {
77        this.gia = gia;
78    }
79
80    public String getDdd() {
81        return ddd;
82    }
83
84    public void setDdd(String ddd) {
85        this.ddd = ddd;
86    }
87
88    public String getSiafi() {
89        return siafi;
90    }
91
92    public void setSiafi(String siafi) {
93        this.siafi = siafi;
94    }
95
96    @Override
97    public String toString() {
98        return "Endereco [cep=" + cep + ", logradouro=" + logradouro + ", complemento=" + complemento + ", bairro="
99            + bairro + ", localidade=" + localidade + ", uf=" + uf + ", ibge=" + ibge + ", gia=" + gia + ", ddd="
100            + ddd + ", siafi=" + siafi + "];";
101    }
102
103 }
```

## Criar o método abaixo:

```
95
96 @Override
97 public String toString() {
98     return "Endereco [cep=" + cep + ", logradouro=" + logradouro + ", complemento=" + complemento + ", bairro="
99         + bairro + ", localidade=" + localidade + ", uf=" + uf + ", ibge=" + ibge + ", gia=" + gia + ", ddd="
100         + ddd + ", siafi=" + siafi + "];";
101 }
102
103 }
```

# Projeto para consultar um CEP – Arquivo ViacepService.class

## Código:

```
package service;

import java.io.IOException;

import org.apache.http.HttpEntity;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.http.util.EntityUtils;

import com.google.gson.Gson;

import model.Endereco;

public class ViacepService {

    public Endereco getEndereco(String cep) throws ClientProtocolException, IOException {

        Endereco endereco = null;

        HttpGet request = new
            HttpGet("https://viacep.com.br/ws/"+cep+"/json/");

        try(CloseableHttpClient httpClient =
            HttpClientBuilder.create().disableRedirectHandling().build();

            CloseableHttpResponse response = httpClient.execute(request)) {

            HttpEntity entity
                = response.getEntity();

            if(entity != null)

                String result = EntityUtils.toString(entity);

                Gson gson = new Gson();

                endereco = gson.fromJson(result, Endereco.class);

            }

            return endereco;

        }

    }

}
```

## Imagem da classe:

```
ViacepService.java x
1 package service;
2
3 import java.io.IOException;
16
17 public class ViacepService {
18
19     public Endereco getEndereco(String cep) throws ClientProtocolException, IOException {
20
21         Endereco endereco = null;
22
23         HttpGet request = new HttpGet("https://viacep.com.br/ws/"+cep+"/json/");
24
25         try(CloseableHttpClient httpClient = HttpClientBuilder.create().disableRedirectHandling().build();
26
27             CloseableHttpResponse response = httpClient.execute(request)) {
28
29             HttpEntity entity = response.getEntity();
30
31             if(entity != null) {
32
33                 String result = EntityUtils.toString(entity);
34
35                 Gson gson = new Gson();
36
37                 endereco = gson.fromJson(result, Endereco.class);
38
39             }
40
41             return endereco;
42         }
43     }
```

# Projeto para consultar um CEP – Arquivo teste.class (Tipo main)

## Código:

```
package teste;

import java.io.IOException;

import model.Endereco;
import service.ViacepService;

public class Teste {

    public static void main(String[] args) {

        ViacepService viacepservice = new ViacepService();

        try {

            Endereco

            endereco = viacepservice.getEndereco("01538001");

            String ddd =

            endereco.getDdd();

            String uf =

            endereco.getUf();

            System.out.println(endereco + "\n");

            System.out.println(ddd + "\n");

            System.out.println(uf + "\n");

        } catch (IOException e) {

            // TODO Auto-

            e.printStackTrace();

        }

    }

}
```

## Imagem da classe:

```
Teste.java ×
1 package teste;
2
3 import java.io.IOException;
4
5 public class Teste {
6
7     public static void main(String[] args) {
8
9         ViacepService viacepservice = new ViacepService();
10
11         try {
12             Endereco endereco = viacepservice.getEndereco("01538001");
13
14             String ddd = endereco.getDdd();
15             String uf = endereco.getUf();
16
17             System.out.println(endereco + "\n");
18             System.out.println(ddd + "\n");
19             System.out.println(uf + "\n");
20
21         } catch (IOException e) {
22             // TODO Auto-generated catch block
23             e.printStackTrace();
24         }
25     }
26 }
27
28
29 }
```



# I DÚVIDAS...

