

# Webservice - Consulta de CEP via PEX

Neste tópicos veremos como fazer uma requisição do **webservice** de CEP via **PEX**.

O Webservice que será usado é o da [ViaCep](#), webservice gratuito que busca todos os CEP's do Brasil.

## Conceito:

CEP - Código postal ou Código de Endereçamento Postal é um código desenvolvido pelas administrações postais e criado com o intuito de facilitar a organização logística e localização espacial de um endereço postal.

Acessando o site da **ViaCep** você encontra toda a documentação necessária para realizar as requisições **JSON, XML e etc.**

## Retorno JSON:

```
https://viacep.com.br/ws/88811578/json <-- Exemplo do retorno
```

```
{
  "cep": "88811-578",
  "logradouro": "Rua Leandro Martignago",
  "complemento": "", "bairro": "Pio Corrêa",
  "localidade": "Criciúma",
  "uf": "SC",
  "unidade": "",
  "ibge": "4204608",
  "gia": ""
}
```

No exemplo a seguir será feito um processo com os campos que o Webservice retorna por exemplo, logradouro, complemento, localidade, uf, unidade, ibge e gia.

image-1640260099791.png

## Criando o Processo.

1º - Crie um formulário do tipo Dissertativa e vincule uma variável por exemplo **/\*CEP\*/** do tipo

texto para aplicar a máscara de CEP.

[image-1640260129434.png](#)

Image not found or type unknown

Neste formulário marque a opção **Ao sair** e logo após nos três pontos na direita:

[image-1640260153130.png](#)

Image not found or type unknown

Logo após insira o código conforme a imagem a seguir:

[image-1640260195858.png](#)

Image not found or type unknown

```
const
  cs_logradouro      = '2' ;
  cs_complemento     = '3' ;
  cs_bairro          = '4' ;
  cs_localidade      = '5' ;    // <--- ID dos formularios
  cs_uf              = '6' ;
  cs_ibge             = '7' ;
  cs_gia              = '8' ;
  cs_unidade         = '9' ;

var
  lsCorpoRequisicao,
  lscaminhoURL,
  Result,
  lsCEP,
  lsSQL              : String;
  loRESTClient       : TRESTClient;
  loRESTRequest      : TRESTRequest;    // <--- Variaveis essenciais para o
funcionamento.
  loRESTResponse     : TRESTResponse;
  loAuthBasica       : THTTPBasicAuthenticator;
  lojson              : tjsonobject;
  loCDS               : Tlibcds;
begin
```

```

lsCEP := aoFormularios.GetJSON('1').GetStr('TEXT0');           // <- Jogando o valor do
campo para a variavel.

//inicia integração com web service

Result      := '';
loRESTClient := nil;
loRESTRequest := nil;
loRESTResponse := nil;
lscaminhoURL := 'http://www.viacep.com.br/ws/' + lsCEP + '/json/'; //<- Inserindo
o valor do campo no link para buscar o JSON
if length(lsCEP) > 1 then
begin
    //executa o requisição
    try
        loRESTClient := TRESTClient .Create(nil);
        loRESTRequest := TRESTRequest .Create(nil);
        loRESTResponse := TRESTResponse.Create(nil);

        loRESTClient.RaiseExceptionOn500 := False;
        loRESTClient.BaseURL              := lscaminhoURL;           //<- Cria os objetos e
conexões necessarias.
        loRESTClient.ContentType          := '';

        loRESTRequest.Method              := rmGet;
        loRestRequest.Resource             := '';
        loRESTRequest.Client               := loRESTClient;
        loRESTRequest.Response             := loRESTResponse;
        try
            loRESTRequest.Execute;
        except
        end;

        if TSTR.of_IsNotNullEma( loRESTResponse.JSONText) then
        begin
            Result := loRESTResponse.JSONText;
            lojson := tjson.op_StringToJson( loRESTResponse.JSONText);
            aoFormularios.GetJSON( cs_logradouro).SetStr('TEXT0',
lojson.GetStr('logradouro'));

```

```

aoFormularios.GetJSON(cs_complemento).SetStr('TEXT0', lojson.GetStr('complemento'));
    aoFormularios.GetJSON(cs_bairro).SetStr('TEXT0', lojson.GetStr('bairro'));
    aoFormularios.GetJSON(cs_localidade).SetStr('TEXT0',
lojson.GetStr('localidade'));
    aoFormularios.GetJSON(cs_uf).SetStr('TEXT0', lojson.GetStr('uf'));    //<- Caso tenha
algo no JSON retorna preenche nos campos.
    aoFormularios.GetJSON(cs_unidade).SetStr('TEXT0', lojson.GetStr('unidade'));
    aoFormularios.GetJSON(cs_ibge).SetStr('TEXT0', lojson.GetStr('ibge'));
    aoFormularios.GetJSON(cs_gia).SetStr('TEXT0', lojson.GetStr('gia'));
end
else
begin
    Result := loRESTResponse.Content;
end;
finally
    loAuthBasica .Free;
    loRESTClient .Free;    //<- Obrigatorio para destruir as variaveis e não ficar
com lixo de memoria.
    loRESTRequest .Free;
    loRESTResponse.Free;
end;
end
else
begin
    aoMensagem.SetStr(' MENSAGEM', 'Para poder efetuar a busca, você precisa informar um CEP
válido');
    aoMensagem.SetInt(' FOCO', 43);
    aoMensagem.SetStr(' TIPO', ' INFO');    // <- Caso o valor seja invalido.
    aoMensagem.SetInt(' TIMEOUT', 6000);
end;
    // Abaixo faz a validação dos campos, caso tenha algum valor preenchido pelo fica
somente leitura se não fica editavel para ser preenchido

    aoFormularios.GetJSON(cs_logradouro).SetBol(' SOMENTELEITURA',
aoFormularios.GetJSON(cs_logradouro).GetStr('TEXT0') <> '' );
    aoFormularios.GetJSON(cs_complemento).SetBol(' SOMENTELEITURA',
aoFormularios.GetJSON(cs_complemento).GetStr('TEXT0') <> '');
    aoFormularios.GetJSON(cs_bairro).SetBol(' SOMENTELEITURA',
aoFormularios.GetJSON(cs_bairro).GetStr('TEXT0') <> '');
    aoFormularios.GetJSON(cs_localidade).SetBol(' SOMENTELEITURA',

```

```
aoFormularios.GetJSON(cs_localidade).GetStr(' TEXTO' ) <> '' );
    aoFormularios.GetJSON(cs_uf).SetBol(' SOMENTELEITURA' ,
aoFormularios.GetJSON(cs_uf).GetStr(' TEXTO' ) <> '' );
    aoFormularios.GetJSON(cs_unidade).SetBol(' SOMENTELEITURA' ,
aoFormularios.GetJSON(cs_unidade).GetStr(' TEXTO' ) <> '' );
    aoFormularios.GetJSON(cs_ibge).SetBol(' SOMENTELEITURA' ,
aoFormularios.GetJSON(cs_ibge).GetStr(' TEXTO' ) <> '' );
    aoFormularios.GetJSON(cs_gia).SetBol(' SOMENTELEITURA' ,
aoFormularios.GetJSON(cs_gia).GetStr(' TEXTO' ) <> '' );

end;
```

[image-1640260301004.gif](#)

Image not found or type unknown

Não é necessário ter todos os campos, você pode adapta-lo a sua necessidade. Os campos quando não tem nenhum valor retornado pelo **JSON** ficam editáveis para ser preenchidos manualmente.

**Versão Homologada : 12.6**

---

Revisão #1

Criado 23 December 2021 08:44:03 por Nicolly Andrielly

Atualizado 6 April 2022 09:07:47 por Nicolly Andrielly