

**zenvia** 

**Tutorial de  
Integração  
.NET**



## Sumário

Tutorial de Integração - .NET .....	3
Configurando Envio .....	4
Para envio de SMS individualmente .....	5
Para consulta de status de SMS individual.....	6
Para envio de Múltiplos SMS por Arquivo .....	7
Para envio de Múltiplos SMS em uma lista.....	9
Para consulta de status de Múltiplos SMS.....	10
Obtendo respostas das requisições .....	11
Glossário.....	12



## Tutorial de Integração - .NET

### Objetivo

O objetivo do tutorial de integração é guiar os desenvolvedores que desejam integrar suas aplicações ao gateway de SMS da Zenvia, automatizando assim seus envios de SMS conforme sua necessidade.

Mostraremos em simples passos como fazer suas aplicações .NET enviarem SMS por meio da API de integração que disponibilizamos para facilitar os envios.

### Pré-requisitos

Para executar a biblioteca de integração, basta ter instalado um servidor que interprete .NET, tal como IIS.

Nenhuma alteração nas configurações do .NET será necessária.

### Entendendo a biblioteca .NET

A biblioteca .NET é um conjunto de classes que tem como principal função fazer requisições HTTP da forma mais organizada e simplificada possível, fazendo com que poucas informações sejam passadas por sua aplicação. A biblioteca foi desenvolvida na linguagem C#, porém este tutorial terá exemplo nas linguagens VB e C#.



## Configurando Envio

### 1º Passo

Faça o *download* da biblioteca de integração por meio do *link* <http://desenvolvedores.zenvia.com/bibliotecas/NET.zip>

### 2º Passo

Adicionar a dll na pasta de lib do seu projeto.

### 3º Passo

Referencie a biblioteca HumanAPIClient.dll.

### 4º Passo

Para qualquer operação do *gateway* de SMS, é preciso identificar-se com sua conta e código de acesso.

Trabalharemos com as instâncias das classes SimpleSending e MultipleSending para manipulação de SMS (envio e consulta), passando por parâmetros de seu construtor a sua conta e o código de acesso. Tais dados são cedidos pela Zenvia juntamente com a negociação do contrato firmado com a empresa.

Exemplo:

C#

```
MultipleSending cliente = new MultipleSending("conta", "senha");
```

VB

```
Dim cliente As MultipleSending = New MultipleSending("conta", "senha")
```

## Para envio de SMS individualmente

### 1º Passo

Instancie a classe *SimpleSending*, informando a sua conta e senha.

Exemplo:

C#

```
SimpleSending cliente = new SimpleSending("conta", "senha");
```

VB

```
Dim cliente As SimpleSending = New SimpleSending("conta", "senha")
```

### 2º Passo

Instancie a classe *SimpleMessage* e adicione as informações *To* (obrigatório), *Message* (obrigatório), *From* (opcional), *Id* (opcional), *Schedule* (opcional) e *Callback* (opcional).

Exemplo:

C#

```
SimpleMessage mensagem = new SimpleMessage();  
mensagem.To = "555199990101";  
mensagem.Message = "Minha mensagem de teste";  
mensagem.Schedule = "17/05/2011 10:00:00";
```

VB

```
Dim mensagem As SimpleMessage = New SimpleMessage();  
mensagem.To = "555199990101";  
mensagem.Message = "Minha mensagem de teste";  
mensagem.Schedule = "17/05/2011 10:00:00";
```

### 3º Passo

Chame o método *SimpleSending.send*, passando por parâmetro a classe *SimpleMessage*.

Exemplo:

C#

```
List<String> retornos = cliente.send(mensagem);
```

VB

```
Dim retornos As List(Of String) = cliente.send(message)
```

### 4º Passo

Para obter as respostas das requisições, veja o item "[Obtendo respostas das requisições](#)".

## Para consulta de status de SMS individual

### 1º Passo

Instancie a classe *SimpleSending*, informando a sua conta e senha.

Exemplo:

C#

```
SimpleSending cliente = new SimpleSending("conta", "senha");
```

VB

```
Dim cliente As SimpleSending = New SimpleSending("conta", "senha")
```

### 2º Passo

Chame o método *SimpleSending.query*, passando por parâmetro o *id* do SMS que deseja consultar.

Exemplo:

C#

```
String id = "010";  
List<String> retornos = cliente.query(id);
```

VB

```
Dim id As String = "010"  
Dim retornos As List(Of String) = cliente.query(id)
```

### 3º Passo

Para obter as respostas das requisições, veja o item ["Obtendo respostas das requisições"](#).

## Para envio de Múltiplos SMS por Arquivo

### 1º Passo

Instancie a classe *MultipleSending*, informando a sua conta e senha.

Exemplo:

C#

```
MultipleSending cliente = new MultipleSending("conta", "senha");
```

VB

```
Dim cliente As MultipleSending = New MultipleSending("conta", "senha")
```

### 2º Passo

Instancie a *FileResource*, informando o arquivo a ser enviado e o [tipo de layout](#) do arquivo. Você pode, também, informar o [tipo de callback](#) para todos os SMS.

Exemplo:

C#

```
MultipleMessage fileMessage = new FileResource(new FileInfo("nome-do-arquivo.txt"),  
LayoutTypeEnum.TYPE_A);  
  
fileMessage.Callback = CallbackTypeEnum.FULL;
```

VB

```
Dim fileMessage As FileResource = New FileResource(New FileInfo("nome-do-arquivo.txt"),  
LayoutTypeEnum.TYPE_A)  
  
fileMessage.Callback = CallbackTypeEnum.FULL
```

### 3º Passo

Chame o método *MultipleSending.send*, passando por parâmetro a classe *FileResource*.

Exemplo:

C#

```
List<String> retornos = cliente.send(fileMessage);
```

VB

```
Dim retornos As List(Of String) = cliente.send(fileMessage)
```

#### 4ºPasso

Para obter as respostas das requisições, veja o item ["Obtendo respostas das requisições"](#).

## Para envio de Múltiplos SMS em uma lista

### 1º Passo

Instancie a classe *MultipleSending*, informando a sua conta e senha.

Exemplo:

C#

```
MultipleSending cliente = new MultipleSending("conta", "senha");
```

VB

```
Dim cliente As MultipleSending = New MultipleSending("conta", "senha")
```

### 2º Passo

Instancie a *ListResource*, informando uma *string* com a lista de SMS a serem enviados e o *tipo de layout* da lista. Você pode, também, informar o *tipo de callback* para todos os SMS; todavia, não é obrigatório. Em nosso exemplo, escolheremos o *layout A*, que consiste em "**to;message**". Para adicionar mais de uma mensagem, inclua o caractere **\n** entre as *strings*.

Exemplo:

C#

```
String data = "555199990101;Mensagem de teste\n555199990102;Mensagem de teste";  
MultipleMessage listMessage = new ListResource(data, LayoutTypeEnum.TYPE_A);
```

```
listMessage.Callback = CallbackTypeEnum.FULL;
```

VB

```
Dim dados As String = "555199990101;Mensagem de teste\n555199990102;Mensagem de teste";
```

```
Dim listMessage As ListResource = New ListResource(dados, LayoutTypeEnum.TYPE_A)
```

```
listMessage.Callback = CallbackTypeEnum.FULL
```

### 3º Passo

Chame o método *MultipleSending.send*, passando por parâmetro a classe *ListResource*.

Exemplo:

C#

```
List<String> retornos = cliente.send(listMessage);
```

VB

```
Dim retornos As List(Of String) = cliente.send(listMessage)
```

#### 4º Passo

Para obter as respostas das requisições, veja o item ["Obtendo respostas das requisições"](#).

### Para consulta de status de Múltiplos SMS

#### 1º Passo

Instancie a classe *MultipleSending*, informando a sua conta e senha.

Exemplo:

C#

```
MultipleSending cliente = new MultipleSending("conta", "senha");
```

VB

```
Dim cliente As MultipleSending = New MultipleSending("conta", "senha")
```

#### 2º Passo

Chame o método *MultipleSending.query*, passando por parâmetro um *array* de *ids* dos SMS que deseja consultar.

Exemplo:

C#

```
String[] ids = new String[]{"010", "009", "008"};  
List<String> retornos = cliente.query(ids);
```

VB

```
Dim ids As String() = New String[]{"010", "009", "008"};  
Dim retornos As List(Of String) = cliente.query(ids)
```

#### 3º Passo

Para obter as respostas das requisições, veja o item ["Obtendo respostas das requisições"](#).

## Obtendo respostas das requisições

O retorno do método de envio é uma lista de [string](#). Nesse texto é possível resgatar o código e a descrição da requisição e do estado dos SMS enviados.

Exemplo:

C#

```
foreach (String resp in retornos)
{
    Console.WriteLine(resp);
}
```

VB

```
For Each resp As String In retornos
    msgResponse.Text = resp
Next
```

## Glossário

Item	Propriedade	Descrição
1	<i>To</i>	Número de telefone do SMS no formato DDI + DDD + Telefone (Exemplo: 555199990101).
2	<i>Message</i>	Mensagem que será enviada ao telefone. Terá no máximo 150 caracteres (sem o campo From).
3	<i>From</i>	Identificação do remetente que será atribuído à mensagem. O campo "Message" + o campo "From" devem ter tamanho máximo de 150 caracteres.
4	<i>Id</i>	Sua identificação do SMS para fins de consulta.
5	<i>Schedule</i>	Data de agendamento de envio do SMS.
6	Tipos de <i>callback</i>	<p>Tipo de callback que a sua aplicação terá como retorno. São eles:</p> <ul style="list-style-type: none"><li>● <b>INACTIVE</b>(0) (Padrão): Não será enviada mensagem de <i>callback</i> para a sua aplicação.</li><li>● <b>FINAL</b>(1): Será enviada a mensagem de <i>callback</i> com somente o estado final de cada mensagem enviada.</li><li>● <b>FULL</b>(2): Será enviada a mensagem de <i>callback</i> com os estados intermediários e o estado final de cada mensagem enviada.</li></ul> <p><b>Observação:</b> Para os <i>callbacks</i> Final e Full, deverá ser cadastrado no Atendimento da Zenvia um <i>host</i> ao qual serão enviadas as requisições.</p>
7	<i>String</i>	Conjunto de caracteres.
8	Tipos de <i>layout</i>	<p>Tipo A: to;message Tipo B: to;message;from Tipo C: to;message;id Tipo D: to;message;id;from Tipo E: to;message;from;id;schedule</p>