

Akatus

Biblioteca para Integração com .NET

Caique Dourado – <http://www.caiquedourado.com.br>

SUMÁRIO

| | |
|--|----|
| Configuração | 3 |
| Carrinho | 4 |
| Processar Transação | 4 |
| Meios de Pagamento..... | 6 |
| Obter os meios de pagamento ativos em sua conta Akatus | 6 |
| Opções de Parcelamento..... | 7 |
| Obter as opções de parcelamento para um determinado meio de pagamento e valor do pedido | 7 |
| Consulta Status | 8 |
| Obter o status de uma transação | 8 |
| NIP – Notificação Instantânea de Pagamento | 9 |
| Recebe uma notificação da Akatus informando a mudança do status de pagamento de uma transação | 9 |
| Tratamento de Erros | 10 |

CONFIGURAÇÃO

Adicione no Web.Config sua API Key, o Tóken NIP, o E-mail utilizado no cadastro da Akatus, e o Ambiente ('producao' ou 'testes').

```
<appSettings>

    <!-- Akatus - Ambiente ('producao' ou 'testes')-->
    <add key="AkatusAmbiente" value="testes"/>

    <!-- Akatus - API Key -->
    <add key="AkatusApiKey" value="SUA-API-KEY"/>

    <!-- Akatus - Tóken NIP -->
    <add key="AkatusTokenNIP" value="SEU-TOKEN-NIP"/>

    <!-- Akatus - E-mail da Conta -->
    <add key="AkatusEmail" value="emailcadastro@empresa.com.br"/>

</appSettings>
```

CARRINHO

PROCESSAR TRANSAÇÃO

```
Akatus.Carri nho. Carri nho carri nho = new Akatus.Carri nho. Carri nho();
```

```
//Nome e E-mail do Comprador
```

```
carri nho. Pagador. Nome = "NOME CLIENTE";  
carri nho. Pagador. Email = "email@cliente.com";
```

```
//Adiciona Endereço do Comprador
```

```
Akatus.Carri nho. PagadorEndereco endereco = new Akatus.Carri nho. PagadorEndereco();  
endereco. Tipo = Akatus.Enums. TipoEndereco. entrega;  
endereco. Logradouro = "Rua Teste da Silva";  
endereco. Numero = 0;  
endereco. Bairro = "CENTRO";  
endereco. Cidade = "Salvador";  
endereco. Estado = "BA";  
endereco. Pais = "BRA";  
endereco. CEP = "40000000";
```

```
carri nho. Pagador. Enderecos. Add(endereco);
```

```
//Adiciona Telefone do Comprador
```

```
Akatus.Carri nho. PagadorTelefone telefone = new Akatus.Carri nho. PagadorTelefone();  
telefone. Tipo = Akatus.Enums. TipoTelefone. celular;  
telefone. Numero = "7199990000";
```

```
carri nho. Pagador. Telefones. Add(telefone);
```

```
//Adiciona Produto
```

```
Akatus.Carri nho. Produto produto = new Akatus.Carri nho. Produto();  
produto. Codigo = "ABC1234567";  
produto. Descricao = "Caixa de bombons sortidos";  
produto. Quantidade = 1;  
produto. Preco = 32.25m;  
produto. Frete = 0;  
produto. Peso = 0;  
produto. Desconto = 0;
```

```
carri nho. Produtos. Add(produto);
```

```
//Forma de Pagamento (Boleto)
```

```
//carri nho. Transacao. MeioDePagamento = Akatus.Enums. MeioDePagamento. boleto;  
//carri nho. Transacao. DescontoTotal = 0;  
//carri nho. Transacao. PesoTotal = 0;  
//carri nho. Transacao. FreteTotal = 0;  
//carri nho. Transacao. Moeda = "BRL";  
//carri nho. Transacao. Referencia = "OFP12345";
```

```
//Forma de Pagamento (Cartão de Crédito)
```

```
carri nho. Transacao. MeioDePagamento = Akatus.Enums. MeioDePagamento. cartao_master;  
carri nho. Transacao. Referencia = "OFP12345";  
carri nho. Transacao. Cartao. Numero = "545301000066167";  
carri nho. Transacao. Cartao. NumeroParcelas = 2;  
carri nho. Transacao. Cartao. CodigoSeguranca = "123";  
carri nho. Transacao. Cartao. Expiracao = "05/2018";  
carri nho. Transacao. Cartao. Portador. Nome = "AUTORI ZAR";  
carri nho. Transacao. Cartao. Portador. CPF = "721.726.663-78";  
carri nho. Transacao. Cartao. Portador. Telefone = "7199990000";
```

```
carri nho.Transacao.DescontoTotal = 0;  
carri nho.Transacao.PesoTotal = 0;  
carri nho.Transacao.FreteTotal = 0;  
carri nho.Transacao.Moeda = "BRL";  
  
//Envia carrinho  
Akatus.Carri nho.Retorno retorno = carri nho.processaTransacao();
```

MEIOS DE PAGAMENTO

OBTER OS MEIOS DE PAGAMENTO ATIVOS EM SUA CONTA AKATUS

```
Akatus.ConsultaMeiosPagamento.Consulta consulta = new  
Akatus.ConsultaMeiosPagamento.Consulta();
```

```
//Consulta meios de pagamento  
List<Akatus.ConsultaMeiosPagamento.Retorno> retorno =  
consulta.consultaMeiosDePagamentoDisponiveis();
```

OPÇÕES DE PARCELAMENTO

OBTER AS OPÇÕES DE PARCELAMENTO PARA UM DETERMINADO MEIO DE PAGAMENTO E VALOR DO PEDIDO

```
Akatus.ConsultaParcelamento.Consulta consulta = new Akatus.ConsultaParcelamento.Consulta();  
  
//Consulta opções de parcelamento  
Akatus.ConsultaParcelamento.Retorno retorno = consulta.consultaParcelamento(10.39m,  
Akatus.Enums.MeioDePagamento.cartao_visa);
```

CONSULTA STATUS

OBTER O STATUS DE UMA TRANSAÇÃO

```
Akatus.ConsultaStatus.Consulta consulta = new Akatus.ConsultaStatus.Consulta();  
  
//Consulta status da transação  
Akatus.ConsultaStatus.Retorno retorno = consulta.consultaStatusTransacao("00000000-0000-0000-  
0000-0000000000");
```


NIP – NOTIFICAÇÃO INSTANTÂNEA DE PAGAMENTO

RECEBE UMA NOTIFICAÇÃO DA AKATUS INFORMANDO A MUDANÇA DO STATUS DE PAGAMENTO DE UMA TRANSAÇÃO

```
//Pega parâmetros postados
string token = Request.Form["token"];
string transacao_id = Request.Form["transacao_id"];
string status = Request.Form["status"];
string referencia = Request.Form["referencia"];

//Processa retorno da transação
Akatus.NotificacaoPagamento.Retorno retorno =
Akatus.NotificacaoPagamento.Notificacao.processaRetorno(token, transacao_id, status,
referencia);
```

TRATAMENTO DE ERROS

```
try
{
    //CODE
}
catch (Akatus.RestException ex)
{
    //Show http status
    Response.Wri te(ex.StatusCode);
    Response.Wri te(ex.StatusDesciption);

    //Show error messages
    foreach (string error in ex.ErrorMessage)
    {
        Response.Wri te("<br />" + error);
    }
}
catch (Exception ex) {

    //Show exception message
    Response.Wri te(ex);

}
```