Documento de utilização Autenticação Client

**Versão 1.1**

**Histórico de Revisões**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição da Revisão** | **Autor** |
| 25/09/2012 | 1.0 | Criação do Documento | Douglas Floriano |
| 26/09/2012 | 1.1 | Inclusão Módulo .Net(DLL) |  |
|  |  |  |  |
|  |  |  |  |

Conteúdo

[1. Utilização do Autenticação Client Jar 4](#_Toc336359996)

[1.1. Bibliotecas Necessárias 4](#_Toc336359997)

[1.2. Exemplo de como Utilizar 4](#_Toc336359998)

[1.3. Observações 9](#_Toc336359999)

[1.4. Requisitos do Sistema 9](#_Toc336360000)

# Utilização do Autenticação Client Jar

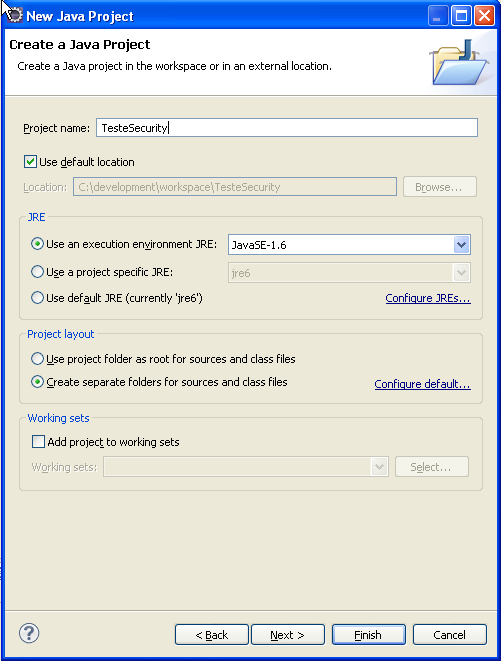
## Bibliotecas Necessárias

1. Para utilizar o Autenticação Client na projeto, será necessário as seguintes bibliotecas:
   1. AutenticacaoClient.jar
   2. commons-codec-1.6.jar
   3. commons-lang3-3.1.jar
2. biblioteca:

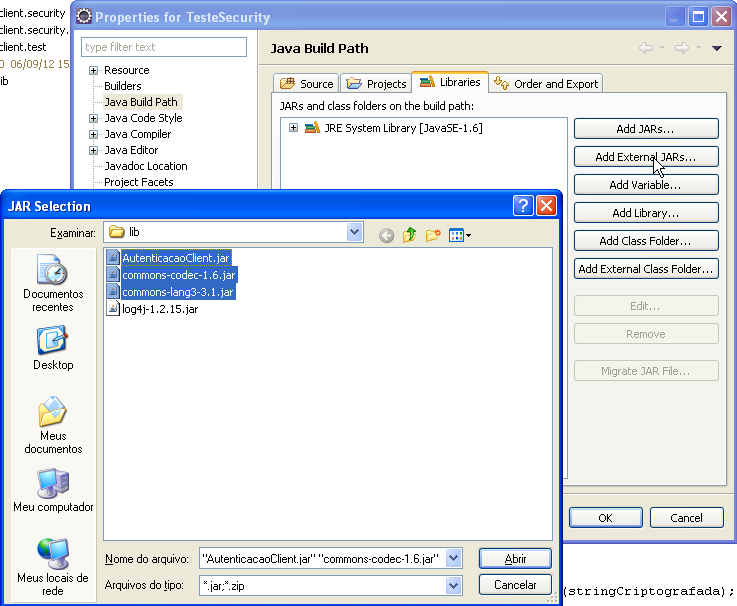


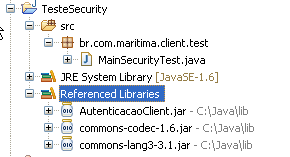
## Exemplo de como Utilizar

1. Criação de Projeto teste.

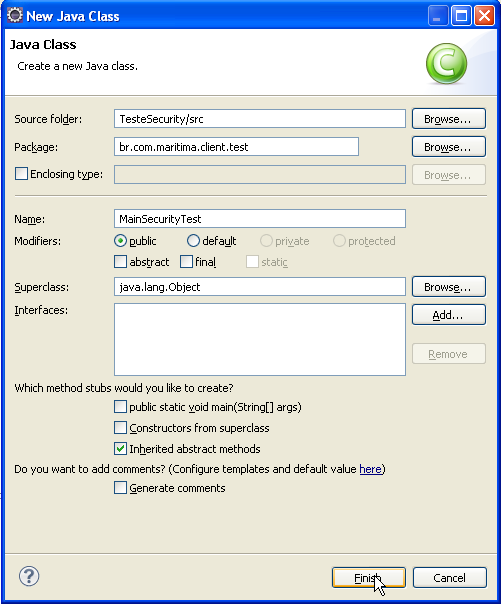


1. Importar Jar para o Projeto.

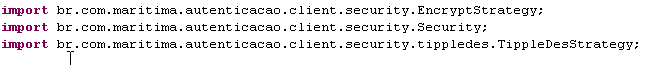




1. Criação Classe de Teste.



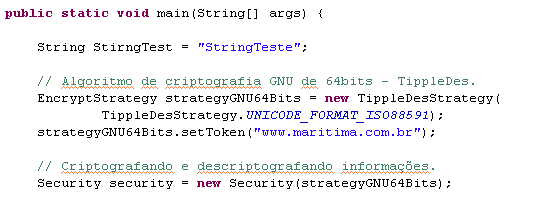
1. Import das classes necessárias.



Deverá criar uma Strategy.

A implementação da mesma está para o Padrão TippleDes.

1. Exemplo de Utilização.

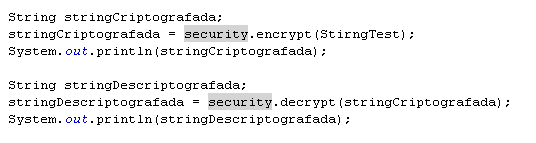


Padrão Unicode para a Criptografia.

Constante padrão para ISO88591.

Objeto para criptografia.

Recebe uma EncryptStrategy

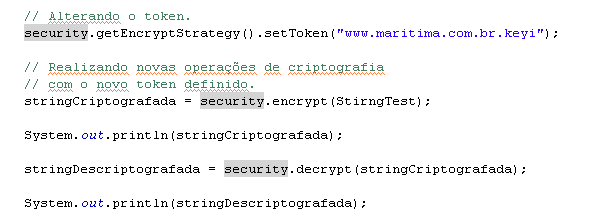


Descriptografando Texto criptografado.

Método decrypt para Security.

Criptografando Texto.

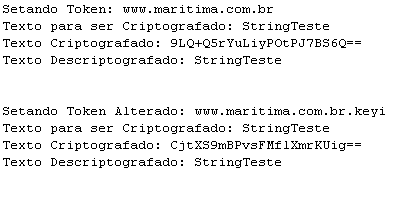
Método encrypt para Security.



Exemplo alterando Token.

1. Resultado Teste.

Texto Criptografado conforme Token Informado.



Texto Criptografado conforme Token Alterado Informado.

1. Código de Implantação para teste.

**package** br.com.maritima.client.test;

**import** br.com.maritima.autenticacao.client.security.EncryptStrategy;

**import** br.com.maritima.autenticacao.client.security.Security;

**import** br.com.maritima.autenticacao.client.security.tippledes.TippleDesStrategy;

**public** **class** MainSecurityTest {

**public** **static** **void** main(String[] args) {

// Algoritmo de criptografia GNU de 64bits - TippleDes.

EncryptStrategy strategyGNU64Bits = **new** TippleDesStrategy(

TippleDesStrategy.*UNICODE\_FORMAT\_ISO88591*);

strategyGNU64Bits.setToken("www.maritima.com.br");

System.*out*.println("Setando Token: www.maritima.com.br");

// Criptografando e descriptografando informações.

Security security = **new** Security(strategyGNU64Bits);

String StirngTest = "StringTeste";

System.*out*.println("Texto para ser Criptografado: " + StirngTest);

String stringCriptografada;

stringCriptografada = security.encrypt(StirngTest);

System.*out*.println("Texto Criptografado: " + stringCriptografada);

String stringDescriptografada;

stringDescriptografada = security.decrypt(stringCriptografada);

System.*out*.println("Texto Descriptografado: " + stringDescriptografada);

//Quebra de Linha

System.*out*.println("\n");

// Alterando o token.

security.getEncryptStrategy().setToken("www.maritima.com.br.keyi");

System.*out*.println("Setando Token Alterado: www.maritima.com.br.keyi");

System.*out*.println("Texto para ser Criptografado: " + StirngTest);

// Realizando novas operações de criptografia

// com o novo token definido.

stringCriptografada = security.encrypt(StirngTest);

System.*out*.println("Texto Criptografado: " + stringCriptografada);

stringDescriptografada = security.decrypt(stringCriptografada);

System.*out*.println( "Texto Descriptografado: " + stringDescriptografada);

}

}

1. Projeto Exemplo de Utilização.



## Observações

1. Atribuições Token.

* Sempre será necessária a atribuição de tolken.
* Tolken poderá ser passado vazio, mas não poderá ser passado nulo. Quando vazio, serão atribuídos 24 caracteres fixos, assim como um texto que contenha menos que 24 caracteres, onde serão completados com caracteres fixos somente os faltantes.
* Token deve ter no máximo 24 caracteres, todos os demais caracteres são desconsiderados na chave de encriptação e descriptação.

## Requisitos do Sistema

• Versão [Java: 1.6](http://www.java.com/pt_BR/download/) ou superior.

# Utilização do CripMulticalculo DLL

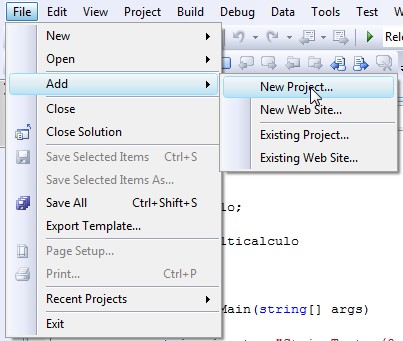
## Bibliotecas Necessárias

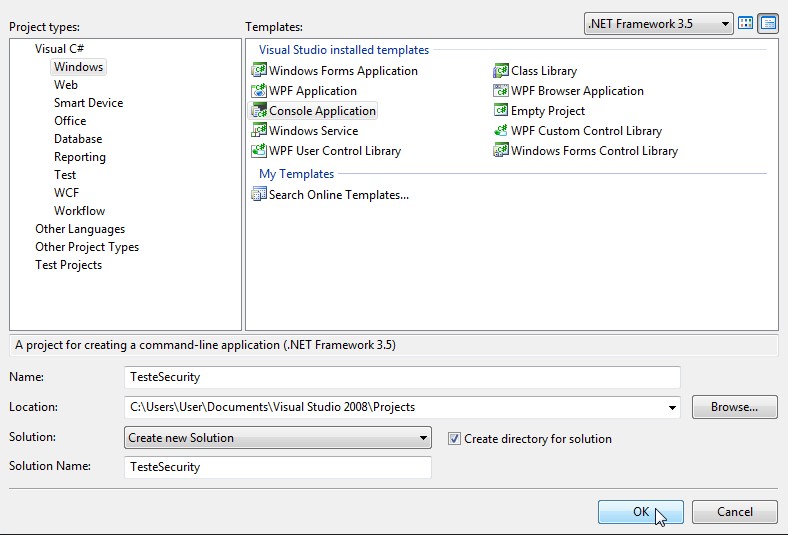
1. Para utilizar o CripMulticalculo no projeto, será necessário a seguinte biblioteca:
   1. CripMulticalculo.dll
2. biblioteca:



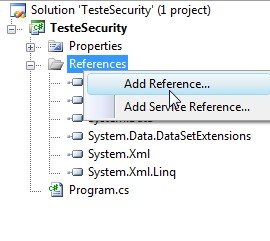
## Exemplo de como Utilizar

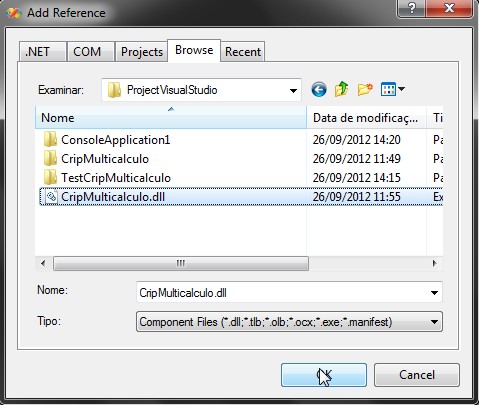
1. Criação de Projeto teste.



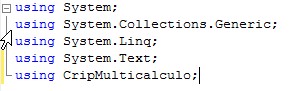


1. Importar Referencia CripMulticalculo(DLL) para o Projeto.





1. Import Referência necessária.



Objeto para criptografia “TippleDesService”.

Método “Encrypt” recebe o texto e o Token para retornar o texto Criptografado.

1. Exemplo de Utilização.



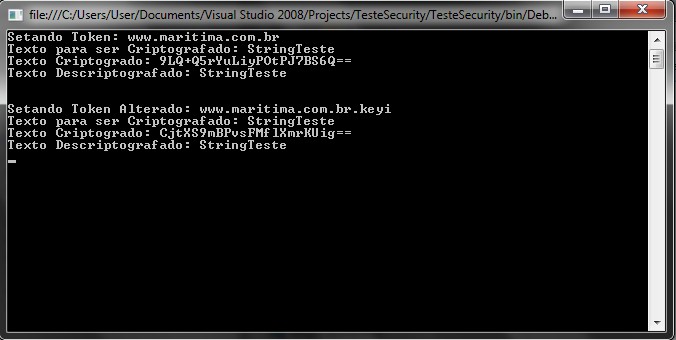
Exemplo alterando Token.

Método “Decrypt” recebe um texto criptografado e o Token para retornar o texto Descriptografado.



Texto Criptografado conforme Token Informado.

1. Resultado Teste.



Texto Criptografado conforme Token Alterado Informado.

1. Código de Implantação para teste.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using CripMulticalculo;

namespace TesteSecurity

{

class Program

{

static void Main(string[] args)

{

string stringTest = "StringTeste";

string token = "www.maritima.com.br";

Console.WriteLine("Setando Token: " + token);

Console.WriteLine("Texto para ser Criptografado: " + stringTest);

string resultadoCriptografado = TippleDesService.Encrypt(stringTest, token);

Console.WriteLine("Texto Criptogrado: " + resultadoCriptografado);

string resultadoDescriptografado = TippleDesService.Decrypt(resultadoCriptografado, token);

Console.WriteLine("Texto Descriptografado: " + resultadoDescriptografado);

Console.WriteLine("\n");

token = "www.maritima.com.br.keyi";

Console.WriteLine("Setando Token Alterado: " + token);

Console.WriteLine("Texto para ser Criptografado: " + stringTest);

resultadoCriptografado = TippleDesService.Encrypt(stringTest, token);

Console.WriteLine("Texto Criptogrado: " + resultadoCriptografado);

resultadoDescriptografado = TippleDesService.Decrypt(resultadoCriptografado, token);

Console.WriteLine("Texto Descriptografado: " + resultadoDescriptografado);

Console.Read();

}

}

}

1. Projeto Exemplo de Utilização.



## Observações

1. Atribuições Token.

* Sempre será necessária a atribuição de tolken.
* Tolken poderá ser passado vazio. Quando vazio, serão atribuídos 24 caracteres fixos, assim como um texto que contenha menos que 24 caracteres, onde serão completados com caracteres fixos somente os faltantes.
* Token deve ter no máximo 24 caracteres, todos os demais caracteres são desconsiderados na chave de encriptação e descriptação.
* No tolken não é permitido mais que 15 caracteres iguais e consecutivos por questão de segurança(Chave Fraca).