

# Componentes do JsWebGets v1.0 – Referência Completa

## Diagrama de Componentes

### JsWebGets

- JsGeneral
- JsTranslation
- JsObject
- JsApplication

### Data

- JsDataConnector
- JsHttpRequest
- JsWebServiceConnector
- JsXML

### Widgets

- JsWidget

### Visual

#### Container

- JsDialog
- JsTab
- JsWebWrapper
- JsWidgetGrid
- JsWidgetStack

#### Display

- JsCalendar
- JsImage
- JsLabel
- JsMenu
- JsMenuBar
- JsMenuItem
- JsToolBar

#### View

- JsListView
- JsListViewItem

### Form

- JsInput

## **Button**

- JsCheckBox
- JsImageButton
- JsPushButton
- JsRadioButton
- JsRadioButtonItem
- JsToolBarButton

## **Field**

- JsCNPJEdit
- JsComboBox
- JsCPFEdit
- JsDateEdit
- JsIPEdit
- JsLineEdit
- JsLineEditAdv
- JsListBox
- JsRichTextEdit
- JsSpinBox
- JsTextEdit
- JsTimeEdit
- JsUpload

# Componentes do JsWebGets v1.0 – Referência Completa

A referência colocada a partir desta página pretende servir como consulta rápida aos métodos públicos dos objetos.

Para melhor compreensão sobre a forma como foi documentada a referência, por favor, atenha-se aos padrões abaixo mostrados. O uso da biblioteca fora desses parâmetros poderá acarretar em mal funcionamento das aplicações que utilizem JsWebGets como biblioteca de interface. Embora o Javascript não faça o controle de atributos públicos ou privados nativamente, é extremamente importante, para o sucesso de sua aplicação, que você limite-se pelos parâmetros aqui definidos.

Por “**Métodos Públicos**” fica definido as APIs dos objetos aos quais os programadores devem ter acesso. Embora há muito mais métodos que os documentados aqui, tais métodos não documentados são para uso interno dos objetos. Para compreendê-los melhor, consulte a “*Referência para Desenvolvedores do JsWebGets*” junto à IDSL.

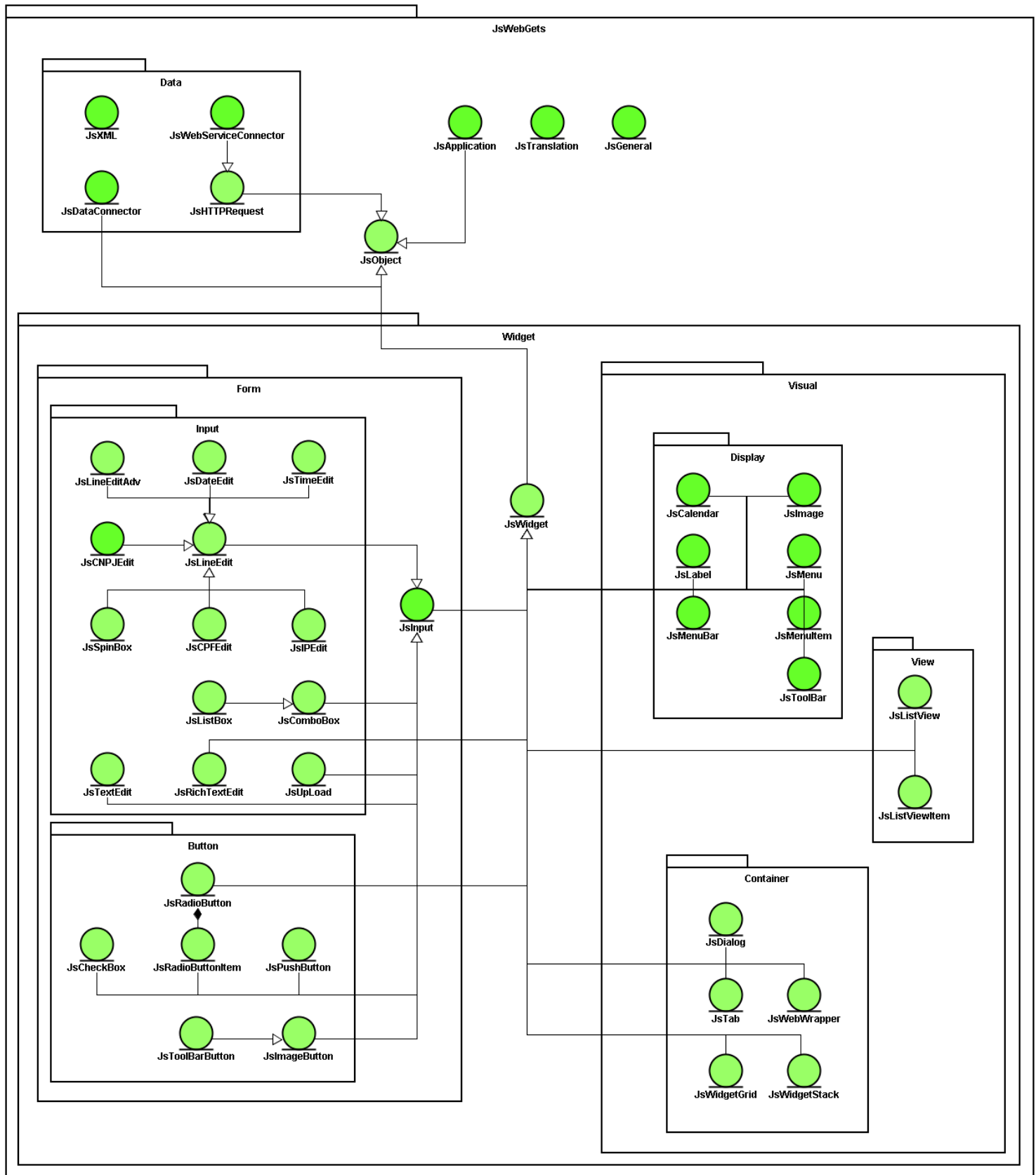
Os **Métodos Herdados** de classe pai são documentados nas classes filhas apenas quando há alteração nos parâmetros informados ou no retorno do método. Nas classes filhas, métodos herdados serão apenas listados, sendo necessário remeter-se à classe pai para ver a definição do método, há menos que haja alguma observação especial a fazer na classe filha (como os casos onde o método, mesmo presente, não tem utilização).

Não há nenhum **Atributo Público** nos objetos, com exceção do JsListView que contém duas referências para seus itens (item seleciona e último inserido). Embora, todos os atributos existentes possam ser alterados e lidos diretamente, reforçamos o pedido para que não o faça para não colocar em risco a compatibilidade e usabilidade futura da biblioteca.

Há ainda, a definição de tipos de variáveis, que segue a lista abaixo

<i>str</i>	= <i>String</i>
<i>int</i>	= <i>Inteiro</i>
<i>bool</i>	= <i>Booleano</i>
<i>any</i>	= <i>Qualquer tipo básico (int, str, bool)</i>
<i>jsobj</i>	= <i>um objeto JsWebGet</i>

# Diagrama de Componentes



## JsGeneral

**Descrição:** JsGeneral refere-se ao arquivo JsGeneral.js. Não é um objeto em si, mas uma série de funções gerais de apoio ao uso da JsWebGets. Define funções e variáveis gerais como browserType, randomizer(), entre outras.

**Arquivo:** JsWebGets/JsGeneral.js

**Tipo de Objeto:** Geral

**Quantidade de Instâncias Simultâneas:** não pode ser instanciado.

## JsTranslation

**Descrição:** JsTranslation refere-se ao arquivo JsTranslation.js. Não é um objeto em si, mas uma série de arrays com tradução para diversos idiomas de itens de interface, como os meses do ano, mensagens de erro, etc.

**Arquivo:** JsWebGets/JsTranslation.js

**Tipo de Objeto:** Geral

**Quantidade de Instâncias Simultâneas:** não pode ser instanciado.

# JsObject

**Descrição:** Classe abstrata que define apenas alguns métodos que estarão presentes em todos os objetos do JsWebGets.

**Arquivo:** JsWebGets/JsObject.js

**Tipo de Objeto:** Geral

**Quantidade de Instâncias Simultâneas:** classe abstrata, não pode ser instanciada.

## Métodos Públicos:

*setValue(any value)*

Define o valor do Objeto.

*getValue(void)*

Retorna o valor definido para o Objeto.

*setAttribute(str name, any value)*

Define o valor de um atributo, novo ou pré-existente para o objeto.

*getAttribute(str name)*

Retorna o valor de um atributo pré-existente para o objeto.

# JsApplication

**Descrição:** é o objeto principal, ao qual todos os outros são amarrados. Representa o documento carregado pela janela, e é responsável por aninhar toda a representação HTML dos objetos, além de carregar o tema para a aplicação.

**Arquivo:** JsWebGets/JsApplication.js

**Tipo de Objeto:** Geral

**Superclasse:** JsObject

**Quantidade de Instâncias Simultâneas:** 1 por janela.

## Métodos Públicos:

*setLanguage(str value)*

Define a língua na qual as mensagens de erro deverão ser fornecidos. Deve estar de acordo com os tipos definidos no arquivo JsTranslation.js

*setTitle(str title)*

Define o título da janela do navegador onde está sendo executada a aplicação.

*setTheme(str theme)*

Define o tema (folha de estilo e imagens) a ser utilizado pela biblioteca.

*addItem(jsobj any)*

Adiciona um item filho ao objeto.

*delItem(jsobj any)*

Remove um item filho do objeto.

*setMenu(jsobj menuobj)*

Define um menu de contexto (botão direito) para o objeto.

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

Sem uso no contexto.

*getAttribute(str name)*

Sem uso no contexto.



# JsDataConnector

**Descrição:** Classe para envio e recepção de dados do servidor. Define um iframe oculto com um formulário que envia os dados para a página remota definida como *handler*. O retorno do *handler* deve ser sempre colocado no nível top ou parent da janela, por tratar-se de iframe oculto. Suas vantagens sobre o uso de HTTP Request e métodos Ajax é permitir a realização de Upload de arquivos.

**Arquivo:** JsWebGets/Data/JsDataConnector.js

**Tipo de Objeto:** Transmissão/Recepção de Dados

**Superclasse:** JsObject

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setHandler(str url)*

Define a página remota no servidor que irá receber os dados postados.

*setFieldValue(str fieldname, any value)*

Define um campo e um valor ao mesmo para envio ao handler definido.

*setFile(jsobj uploadobj)*

Adiciona um arquivo para upload na lista dos itens a serem enviados ao handler. Pode haver *n* arquivos para upload.

*postData(void)*

Executa a ação de envio dos dados para o handler.

*clearData(void)*

Limpa o DataConnector dos valores enviados anteriormente. É chamado automaticamente pelo postData().

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

Sem uso no contexto.

*getAttribute(str name)*

Sem uso no contexto.

# JsHttpRequest

**Descrição:** Classe para envio e recepção de dados do servidor. Define um objeto HTTP Request que envia os dados para a página ou webservice remoto definido como *handler*. O retorno do *handler* pode ser tanto código Javascript a ser utilizado diretamente pela aplicação no mesmo nível da janela, ou um objeto XML. Não é permitida a realização de Upload de arquivos por motivos de segurança.

**Arquivo:** JsWebGets/Data/JsHttpRequest.js

**Tipo de Objeto:** Transmissão/Recepção de Dados

**Superclasse:** JsObject

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setHandler(str url)*

Define a página ou webservice remoto no servidor que irá receber os dados postados

*createHandler(function)*

Define a função a ser executada quando o envio e recepção dos dados é executado com sucesso.

*setFieldValue(str fieldname, any value)*

Define um campo e um valor ao mesmo para envio ao handler definido.

*setXML(jsobj xmlobj)*

Define o XML a ser enviado ao handler.

*postData(void)*

Executa a ação de envio dos dados para o handler.

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

Sem uso no contexto.

*getAttribute(str name)*

Sem uso no contexto.

# JsWebServiceConnector

**Descrição:** Classe para criação de interface de comunicação com WebServices, onde o WSDL é utilizado para a criação de API similar ao do WebService para chamada direta dentro do código Javascript.

**Arquivo:** JsWebGets/Data/JsWebServiceConnector.js

**Tipo de Objeto:** Transmissão/Recepção de Dados

**Superclasse:** JsHttpRequest

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*bind(str wsurl, function callback)*

Conecta o objeto ao WebService correspondente na URL, construindo a API logo em seguida. O parâmetro callback é a função que deve ser executada assim que o objeto estiver pronto para utilização. Pode ser deixado em branco (não recomendado por risco de haver chamada de métodos antes da API estar pronta para uso).

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

Sem uso no contexto.

*getAttribute(str name)*

Sem uso no contexto.

*setHandler(str url)*

Sem uso no contexto.

*createHandler(function)*

Sem uso no contexto.

*setFieldValue(str fieldname, any value)*

Sem uso no contexto.

*setXML(jsobj xmlobj)*

Sem uso no contexto.

*postData(void)*

Sem uso no contexto.

# JsXML

**Descrição:** Implementa um objeto XML comum às duas implementações dos navegadores Internet Explorer e Mozilla/Firefox, permitindo a compatibilidade de código entre os navegadores.

**Arquivo:** JsWebGets/Data/JsXML.js

**Tipo de Objeto:** Objeto XML

**Superclasse:** DOMDocument(Internet Explorer)/XMLDocument (Mozilla/Firefox).

**Quantidade de Instâncias Simultâneas:** n

**ATENÇÃO** - os métodos e atributos listados aqui são os únicos suportados por ambos os navegadores. Utilizar métodos específicos de um navegador que não estejam listados abaixo não garantirá o funcionamento de sua aplicação no outro.

## Atributos Públicos:

*xml*

Retorna o objeto XML como uma string (implementado como extensão do objeto XMLElemento no JsWebGets)

*async*

Define se o processamento deve ser assíncrono ou não (default = false)

*nodeName*

Nome do nó

*nodeValue*

Valor do nó

*nodeType*

Tipo do nó

*namespaceURI*

URI do namespace do objeto XML

*prefix*

prefixo

*doctype*

tipo do documento

*implementation*

Implantação do XML

*documentElement*

Pega o nó raiz do objeto XML atual

*parentNode*

Referência para o nó pai

*childNodes*

Referência para os nós filhos

*firstChild*

Referência para o primeiro filho

*lastChild*

Referência para o último filho

*previousSibling*

Referência para o nó anterior

*nextSibling*

Referência para o nó posterior

*attributes*

Atributos do nó

*ownerDocument*

Referência para o objeto XMLDocument ao qual o nó atual refere-se.

### **Métodos Públicos:**

*getElementsByTagName(str tagname)*

Retorna um array de nós identificados pela tag informada

*insertBefore(new node, old node)*

Insere um nó antes de outro

*replaceChild(new node, old node)*

Substitui um nó por outro

*removeChild(node)*

Remove um nó

*appendChild(node)*

Adiciona um nó ao final da childNodes

*hasChildNodes()*

Retorna um booleano indicando se tem filhos ou não

*cloneNode()*

Clona um nó

*createElement(str nodeType)*

Cria um elemento (nó) avulso

*createDocumentFragment()*

Cria um fragmento de documento

*createTextNode(str text)*

Cria um nó de texto com o texto especificado.

*createComment(str comment)*

Cria um nó de comentário

*createCDATASection()*

Cria uma seção CDATA

*createProcessingInstruction()*

Cria uma instrução de processamento

*createAttribute()*

Cria um atributo

*createEntityReference()*

Cria uma referência para uma entidade

# JsWidget

**Descrição:** Classe abstrata, de onde são geradas todas as outras classes de widgets que terão representação HTML visível na página.

**Arquivo:** JsWebGets/Widget/JsWidget.js

**Tipo de Objeto:** Interface/Geral

**Superclasse:** JsObject

**Quantidade de Instâncias Simultâneas:** classe abstrata, não pode ser instanciada.

## Métodos Públicos:

*addItem(jsobj any)*

Adiciona um item filho ao objeto.

*delItem(jsobj any)*

Remove um item filho do objeto.

*setMenu(jsobj menuobj)*

Define um menu de contexto (botão direito) para o objeto.

*setToolTip(str value)*

Define o texto do Tooltip do Widget.

*setEvent(str eventtype, function)*

Define uma função a ser executada para um evento dado sobre o objeto em questão. Como parâmetros, aceita a string do evento que disparará a função, e a função em si, sem parênteses ou parâmetros. A string pode ser qualquer uma definida no DOM, sem a utilização do prefixo "on".

Ex: obj.setEvent("click",funcao);

*unsetEvent(str eventtype, function)*

Remove uma função a ser executada para um evento dado sobre o objeto em questão. Como parâmetros, aceita a string do evento que disparará a função, e a função em si, sem parênteses ou parâmetros. A string pode ser qualquer uma definida no DOM, sem a utilização do prefixo "on".

Ex: obj.unsetEvent("click",funcao);

*disableEvent(str eventtype, function)*

Desabilita uma função a ser executada para um evento dado sobre o objeto em questão. Como parâmetros, aceita a string do evento que disparará a função, e a função em si, sem parênteses ou parâmetros. A string pode ser qualquer uma definida no DOM, sem a utilização do prefixo "on".

Ex: obj.unsetEvent("click",funcao);

*setXPos(int x)*

Define o posicionamento absoluto na horizontal em relação ao objeto pai.

*setYPos(int y)*

Define o posicionamento absoluto na vertical em relação ao objeto pai.

*setHeight(int h)*

Define a altura do objeto.

*setWidth(int w)*

Define a largura do objeto.

*setClass(str classname)*

Define a classe CSS do objeto.

*setCursor(str cursortype)*

Define o ponteiro do mouse, quando estiver sobre o objeto.

*setFont(str fontfamily)*

Define a família da fonte do texto.

*setFontColor(str fontcolor)*

Define a cor da fonte do texto.

*setFontSize(str fontsize)*

Define o tamanho da fonte do texto.

*setFontWeight(str fontweight)*

Define o peso da fonte do texto.

*setFontStyle(str fontstyle)*

Define o estilo da fonte do texto.

### **Métodos Herdados:**

*setValue(any value)*

Define o valor do Objeto.

*getValue(void)*

Retorna o valor definido para o Objeto.

*setAttribute(str name, any value)*

Define o valor de um atributo, novo ou pré-existente para o objeto.

*getAttribute(str name)*

Retorna o valor de um atributo pré-existente para o objeto.



## JsDialog

**Descrição:** Classe para construção de diálogos, a fim de evitar bloqueadores de Pop-up, e facilitar a codificação mantendo todos os elementos no mesmo nível da janela do navegador. Pode se auto-ocultar, e aparecer já com posicionamento no centro da tela

**Arquivo:** JsWebGets/Widget/Container/JsDialog.js

**Tipo de Objeto:** Interface - Container

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

*hideDialog(void)*

Oculto o diálogo.

*showDialog(void)*

Mostra o diálogo com posicionamento absoluto em relação à janela, e centralizado.

### Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

*delItem(jsobj any)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsTab

**Descrição:** Classe para a criação de abas (tabulações) similares às encontradas em sistemas desktop, permitindo uma organização melhorada do conteúdo na tela.

**Arquivo:** JsWebGets/Widget/Container/JsTab.js

**Tipo de Objeto:** Interface - Container

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*addTab(str tablabel)*

Adiciona uma tabulação ao objeto.

*delTab(int tabindex)*

Remove a tabulação indicada pelo índice.

*addItem(jsobj any)*

Adiciona um objeto à última tabulação criada.

*addItemToTab(jsobj any, int tabindex)*

Adiciona um objeto à uma tabulação especificada, indicada pelo índice.

*showTab(int tabindex)*

Mostra a tabulação indicada pelo índice

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsWebWrapper

**Descrição:** Classe que cria um iframe para visualização de conteúdo externo à aplicação (como páginas web).

**Arquivo:** JsWebGets/Widget/Container/JsWebWrapper.js

**Tipo de Objeto:** Interface - Container

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setPage(str url)*

Define a página web a ser mostrada no iframe.

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

Sem uso no contexto.

*getAttribute(str name)*

Sem uso no contexto.

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

Sem uso no contexto.

*setToolTip(str value)*

Sem uso no contexto.

*setEvent(str eventtype, function)*

Sem uso no contexto.

*unsetEvent(str eventtype, function)*

Sem uso no contexto.

*disableEvent(str eventtype, function)*

Sem uso no contexto.

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

Sem uso no contexto.

*setCursor(str cursortype)*

Sem uso no contexto.

*setFont(str fontfamily)*

Sem uso no contexto.

*setFontColor(str fontcolor)*

Sem uso no contexto.

*setFontSize(str fontsize)*

Sem uso no contexto.

*setFontWeight(str fontweight)*

Sem uso no contexto.

*setFontStyle(str fontstyle)*

Sem uso no contexto.

# JsWidgetGrid

**Descrição:** Classe utilizada para auxiliar no posicionamento dos elementos na tela de forma tabular. Funciona de maneira similar a uma tabela (de fato é uma tabela).

**Arquivo:** JsWebGets/Widget/Container/JsWidgetGrid.js

**Tipo de Objeto:** Interface - Container

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*addRow(void)*

Adiciona uma linha ao Grid.

*addCell(int width, int height, str align, str valign, int colspan, int rowspan, str bgcolor, str url\_bgimage)*

Adiciona uma célula à última linha adicionada. É preciso adicionar uma linha antes de chamar este método.

*addItem(jsobj any)*

Adiciona um item à última célula criada na linha.

*clearGrid(void)*

Remove todas as linhas do grid.

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



# JsWidgetStack

**Descrição:** Cria uma pilha de divs invisíveis cujo conteúdo é mostrado um a um. É utilizado dentro do objeto JsTab para permitir a sobreposição e apresentação das tabulações criadas, e pode ser utilizado sem problema para a criação de outros elementos como wizards.

**Arquivo:** JsWebGets/Widget/Container/JsWidgetStack.js

**Tipo de Objeto:** Interface - Container

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*addStack(void)*

Adiciona um stack à pilha.

*delStack(int stackindex)*

Remove o stack indicado pelo índice.

*showStack(int stackindex)*

Apresenta o stack indicado pelo índice, ocultando os outros.

*addItem(jsobj any)*

Adiciona um item ao último stack adicionado.

*addItemToStack(jsobj any, int stackindex)*

Adiciona um item ao stack indicado pelo índice.

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsCalendar

**Descrição:** Classe que cria um calendário. Trás, por padrão, o valor do dia de hoje.

**Arquivo:** JsWebGets/Widget/Container/JsCalendar.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*build(void)*

Monta o calendário com o valor de data definido pelo setValue().

*getValue(void)*

Retorna o valor da data selecionada no calendário no formato YYYY-MM-DD.

*setValue(isodate datevalue)*

Define o valor da data selecionada no calendário no formato YYYY-MM-DD.

*setDay(int day)*

Define apenas o dia para o calendário, mantendo mês e ano.

*setMonth(int month)*

Define apenas o mês para o calendário, mantendo dia e ano.

*setYear(int year)*

Define apenas o ano para o calendário, mantendo dia e mês.

## Métodos Herdados:

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsImage

**Descrição:** Classe que apresenta uma imagem dada qualquer.

**Arquivo:** JsWebGets/Widget/Container/JsImage.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setSource(str urlimage)*

Define a imagem que será mostrada pelo objeto.

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

Sem uso no contexto.

*setFontColor(str fontcolor)*

Sem uso no contexto.

*setFontSize(str fontsize)*

Sem uso no contexto.

*setFontWeight(str fontweight)*

Sem uso no contexto.

*setFontStyle(str fontstyle)*

Sem uso no contexto.

# JsLabel

**Descrição:** Classe que cria um rótulo de texto, que pode conter um ícone, e ser definido como editável através de duplo clique.

**Arquivo:** JsWebGets/Widget/Container/JsLabel.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setIcon(str utlimage, str align)*

Define o ícone e alinhamento do mesmo em relação ao texto.

*setEditable(void)*

Define o objeto para entrar em modo de edição se clicado duplamente.

## Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



# JsMenu

**Descrição:** Classe que cria um objeto do tipo Menu. Este objeto pode ser adicionado a uma barra de menus, a um outro menu como submenu, ou ainda como menu de contexto, ativado com o botão direito em qualquer objeto. No caso de adicionar um JsMenu como menu de contexto de outro objeto, o mesmo JsMenu poderá ser adicionado a vários objetos sem perda de referência ou duplicação de menus.

**Arquivo:** JsWebGets/Widget/Container/JsMenu.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setValue(str value)*

Define o valor que o Label do menu terá. O label é o texto apresentado na barra de menus ou no menu pai.

*setIcon(str url\_icon)*

Define o ícone apresentado no Label do menu.

*addItem(jsobj menuobj), addItem(jsobj menuitemobj)*

Adiciona outros objetos ao menu. Caso o objeto seja outro menu, você terá adicionado um submenu.

*addDiv(void)*

Adiciona uma linha divisória entre os menus.

## Métodos Herdados:

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

*setMenu(jsobj menuobj)*

Sem uso no contexto.

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsMenuBar

**Descrição:** Classe que cria uma barra de menu para a aplicação, que pode ser posicionada em qualquer local da tela, mas é normalmente colocada no alto.

**Arquivo:** JsWebGets/Widget/Container/JsMenuBar.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

*addItem(jsobj menuobj)*

Adiciona um objeto do tipo JsMenu à barra de menu da aplicação.

### Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

Sem uso no contexto.

*getAttribute(str name)*

Sem uso no contexto.

*delItem(jsobj any)*

*setMenu(jsobj menuobj)*

Sem uso no contexto.

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsMenuItem

**Descrição:** Classe que define os itens de um menu.

**Arquivo:** JsWebGets/Widget/Container/JsMenuItem.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsLabel

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setValue(str value)*

Define o texto apresentado.

*setIcon(str url\_icon)*

Define o ícone apresentado.

## Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

*delItem(jsobj any)*

*setIcon(str utlimage, str align)*

*setEditable(void)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsToolBar

**Descrição:** Classe que cria uma barra de ferramentas, geralmente utilizado logo abaixo do JsMenuBar.

**Arquivo:** JsWebGets/Widget/Container/JsToolBar.js

**Tipo de Objeto:** Interface - Display

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*addItem(jsobj toolbarobj)*

Adiciona um JsToolBarButton à barra de ferramentas.

*addDiv(void)*

Adiciona uma linha divisória entre os botões

## Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*getValue(void)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

*setMenu(jsobj menuobj)*

Sem uso no contexto.

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*  
Sem uso no contexto.

*setFontColor(str fontcolor)*  
Sem uso no contexto.

*setFontSize(str fontsize)*  
Sem uso no contexto.

*setFontWeight(str fontweight)*  
Sem uso no contexto.

*setFontStyle(str fontstyle)*  
Sem uso no contexto.



## JsListView

**Descrição:** Classe que cria um Listview que serve tanto como gridview (tabela) quanto treeview, com colunas dimensionáveis e ordenáveis.

**Arquivo:** JsWebGets/Widget/Container/JsListView.js

**Tipo de Objeto:** Interface - View

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

### Atributos Públicos:

*selectedItem*

Referência para o objeto JsListViewItem que estiver selecionado no Listview.

*lastInsertedItem*

Referência para o último objeto JsListViewItem que adicionado ao Listview.

### Métodos Públicos:

*setTreeView(void)*

Define que o JsListview funcionará como treeview. Deve ser definido antes de adicionar-se qualquer JsListViewItem.

*addColumn(str headerstr, int headerwidth, str url\_icon)*

Adiciona uma coluna ao JsListView. O único parâmetro não obrigatório é o ícone.

*delColumn(int colindex)*

Remove a coluna identificada pelo índice informado.

*addItem(jsobj listviewitemobj)*

Adiciona um JsListViewItem ao JsListView.

*delItem(jsobj listviewitemobj)*

Remove um item. Geralmente é utilizado referenciando o item selecionado. Ex:  
obj\_listview.delItem(obj\_listview.selectedItem);

*getValue(void)*

Retorna o valor dos itens dentro do JsListView no formato de um array multidimensional.

*clearData(void)*

Limpa o JsListView removendo todos os itens filhos.

### Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

Sem uso no contexto.

*setFontColor(str fontcolor)*

Sem uso no contexto.

*setFontSize(str fontsize)*

Sem uso no contexto.

*setFontWeight(str fontweight)*

Sem uso no contexto.

*setFontStyle(str fontstyle)*

Sem uso no contexto.

## JsListViewItem

**Descrição:** Classe que define os itens de um JsListView. São as linhas dentro de um ListView. A cada célula da linha, corresponde outro objeto JsWebGet, que pode ser um JsLabel, JsComboBox, JsRadioButton, JsCheckBox, ou o que se faça necessário.

**Arquivo:** JsWebGets/Widget/Container/JsListViewItem.js

**Tipo de Objeto:** Interface - View

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

O método addItem tem 3 formas de ser utilizado, cada um explicado abaixo.

*self.addItem(jsobj listviewitemobj)*

Se o argumento passado for outro JsListViewItem, adiciona este como nó filho do Item pai dentro de uma estrutura em árvore.

*self.addItem(jsobj anyobj)*

Se o argumento for um outro objeto qualquer JsWebGet, ele será adicionado como célula do item, e a altura do item será ajustada a ele, se necessário.

*self.addItem(str value, str url\_icon, bool editable)*

A terceira forma de uso é passando apenas uma string, isso criará automaticamente um objeto JsLabel com aquela string. Outros dois argumentos são permitidos, sendo eles, o ícone do JsLabel, e se a célula será editável através de duplo clique.

*getValue(void)*

Retorna um array multidimensional, onde cada item do array representa uma célula, e o último item do array representa os valores dos nós filhos (outros JsListViewItems que tenham sido adicionados a este).

### Métodos Herdados:

*setValue(any value)*

Sem uso no contexto.

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsInput

**Descrição:** Classe abstrata, de onde são geradas todas as outras classes de widgets que terão representação HTML visível na página e que possuem um objeto do tipo input dentro, permitindo receber entrada de dados do usuário.

**Arquivo:** JsWebGets/Widget/Form/JsInput.js

**Tipo de Objeto:** Interface – Form/Geral

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*disable(bool value)*

Desabilita o campo input, bloqueando a entrada. O valor passado deve ser true or false, indicando se o campo deve ou não ser desabilitado.

## Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsCheckBox

**Descrição:**

**Arquivo:** JsWebGets/Widget/Form/Button/JsCheckBox.js

**Tipo de Objeto:** Interface – Form – Button

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

`setLabel(str value)`

Define o rótulo do checkbox.

`setChecked(bool value)`

Define se o rótulo está checado ou não

**Métodos Herdados:**

*setValue(any value)*

*getValue(void)*

Retorna falso se o campo não estiver selecionado

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



# JsImageButton

**Descrição:** Classe que cria um botão do tipo imagem.

**Arquivo:** JsWebGets/Widget/Form/Button/JsImageButton.js

**Tipo de Objeto:** Interface – Form – Button

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setSource(str url\_image)*

Define a url da imagem a ser apresentada.

## Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsPushButton

**Descrição:** Classe que cria um botão

**Arquivo:** JsWebGets/Widget/Form/Button/JsPushButton.js

**Tipo de Objeto:** Interface – Form – Button

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setLabel(str value)*

Define o texto do botão

## Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsRadioButton

**Descrição:** Cria um grupo de radiobuttons.

**Arquivo:** JsWebGets/Widget/Form/Button/JsRadioButton.js

**Tipo de Objeto:** Interface – Form – Button

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setOrientation(str (“v” | “h”))*

Define a orientação na tela. “v” = vertical, “h” = horizontal.

*addItem(any value, any label, bool selected)*

Adiciona uma nova opção à listagem de radios.

## Métodos Herdados:

*setValue(any value)*

Define como marcado o radio que tiver aquele valor

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*deleteItem(jsobj any)*

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsRadioButtonItem

**Descrição:** São os itens do JsRadioButton. Geralmente criados automaticamente a partir do método `addItem` do JsRadioButton.

**Arquivo:** JsWebGets/Widget/Form/Button/JsRadioButtonItem.js

**Tipo de Objeto:** Interface – Form – Button

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

*setLabel(str value)*

Define o texto de um JsRadioButtonItem.

### Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



## JsToolBarButton

**Descrição:** Classe que cria um botão para a barra de ferramentas.

**Arquivo:** JsWebGets/Widget/Form/Button/JsToolBarButton.js

**Tipo de Objeto:** Interface – Form – Button

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setSource(str url\_image)*

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsCNPJEdit

**Descrição:** Classe que cria um campo de entrada de valores do tipo CNPJ, com aplicação de máscara e validação de CNPJ.

**Arquivo:** JsWebGets/Widget/Form/Input/JsCNPJEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setMaxLength(int value)*  
Sem uso no contexto.

*setPassword(void)*  
Sem uso no contexto.

*setNumeric(void)*  
Sem uso no contexto.

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*  
Sem uso no contexto.

*delItem(jsobj any)*  
Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsComboBox

**Descrição:** Classe que cria uma lista do tipo drop-down para seleção de opções.

**Arquivo:** JsWebGets/Widget/Form/Input/JsComboBox.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*addItem(any value, any label, str url\_icon, bool selected)*

Adiciona um item à lista. O parâmetro “icon” foi mantido apenas para compatibilidade reversa, mas deve ser reimplementado na versão 2.0 da biblioteca.

*clearData(void)*

Limpa a lista de itens do JsComboBox

*getLabel()*

Pega o texto do item selecionado

## Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

*delItem(jsobj any)*

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsCPFEdit

**Descrição:** Classe que cria um campo de entrada de valores do tipo CPF, com aplicação de máscara e validação de CPF.

**Arquivo:** JsWebGets/Widget/Form/Input/JsCPFEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setMaxLength(int value)*  
Sem uso no contexto.

*setPassword(void)*  
Sem uso no contexto.

*setNumeric(void)*  
Sem uso no contexto.

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*  
Sem uso no contexto.

*delItem(jsobj any)*  
Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



## JsDateEdit

**Descrição:** Classe que define um campo para entradas do tipo Data. Aceita apenas saída ou entrada de dados automática no formato iso. A entrada do usuário deve ser no formato brasileiro (dd-mm-yyyy).

**Arquivo:** JsWebGets/Widget/Form/Input/JsDateEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setMaxLength(int value)*  
Sem uso no contexto.

*setPassword(void)*  
Sem uso no contexto.

*setNumeric(void)*  
Sem uso no contexto.

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*  
Sem uso no contexto.

*delItem(jsobj any)*  
Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsIPEdit

**Descrição:** Classe que cria um campo para entrada de valores do tipo IP, com validação da entrada.

**Arquivo:** JsWebGets/Widget/Form/Input/JsIPEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

*setNetMask()*

Define a máscara de IP a ser utilizada para validação.

### Métodos Herdados:

*setMaxLength(int value)*

Sem uso no contexto.

*setPassword(void)*

Sem uso no contexto.

*setNumeric(void)*

Sem uso no contexto.

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsLineEdit

**Descrição:** Campo simples para entrada de dados. Pode ainda ser convertido para campo do tipo senha, ter o máximo de caracteres definido, e ser definido como campo apenas numérico.

**Arquivo:** JsWebGets/Widget/Form/Input/JsLineEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

*setMaxLength(int value)*

Define comprimento máximo do campo em caracteres.

*setPassword(void)*

Define o campo como tipo senha.

*setNumeric(void)*

Define campo como tipo numérico apenas.

### Métodos Herdados:

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsLineEditAdv

**Descrição:** Campo de texto com auto-completação. É preciso definir o *handler* da consulta das lista a ser carregada, que pode ser tanto uma página quanto um webservice, acessível por HTTPRequest.

**Arquivo:** JsWebGets/Widget/Form/Input/JsLineEditAdv.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

### Métodos Públicos:

*setMinSearchLength(int value)*

Define o mínimo de caracteres para efetuar consulta ao *handler*.

*setHandler(str url)*

Define o *para* consulta, que pode ser tanto uma página web quanto um webservice. O *handler* deve estar pronto para receber as seguintes variáveis enviadas por método POST:

*searchkey* – chave a ser buscada na fonte de dados consultada pelo *handler*

*leadv\_name* – nome do objeto solicitante (para gerar resposta)

*addItem(any value, any label, str url\_icon, bool selected)*

Adiciona itens na lista de seleção.

### Métodos Herdados:

*setMaxLength(int value)*

*setPassword(void)*

Sem uso no contexto.

*setNumeric(void)*

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



# JsListBox

**Descrição:** Classe que define uma lista de múltipla escolha.

**Arquivo:** JsWebGets/Widget/Form/Input/JsListBox.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsComboBox

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*resetSelecteds(void)*

Desmarca todos os itens selecionados.

*getValue(void)*

Retorna um array multidimensional, onde cada item é composto de um par contendo o valor e o texto do item selecionado.

## Métodos Herdados:

*addItem(any value, any label, str url\_icon, bool selected)*

Adiciona um item à lista. O parâmetro “icon” foi mantido apenas para compatibilidade reversa, mas deve ser reimplementado na versão 2.0 da biblioteca.

*clearData(void)*

Limpa a lista de itens do JsComboBox

*getLabel()*

Sem uso no contexto.

*setValue(any value)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*delItem(jsobj any)*

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsRichTextEdit

**Descrição:** Editor tipo Rich Text.

**Arquivo:** JsWebGets/Widget/Form/Input/JsRichTextEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsSpinBox

**Descrição:** Campo numérico com botões laterais para auto-incremento, de acordo com o *raise factor* definido.

**Arquivo:** JsWebGets/Widget/Form/Input/JsSpinBox.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

## Métodos Públicos:

*setRaiseFactor(int value)*

Define o *raise factor* – fator de incremento – utilizado quando os botões laterais são clicados para mais ou para menos.

## Métodos Herdados:

*setMaxLength(int value)*

*setPassword(void)*

Sem uso no contexto.

*setNumeric(void)*

Sem uso no contexto.

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

## JsTextEdit

**Descrição:** Campo para texto longo, com quebras de linha, mas sem edição Rich Text, tipo textarea.

**Arquivo:** JsWebGets/Widget/Form/Input/JsTextEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsWidget

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*



## JsTimeEdit

**Descrição:** Campo para entrada de dados do tipo hora, com validação e máscara automáticas.

**Arquivo:** JsWebGets/Widget/Form/Input/JsTimeEdit.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsLineEdit

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setValue(any value)*

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*

# JsUpload

**Descrição:** Campo para upload de arquivos. Deve ser utilizado apenas com JsDataConnector, pois os métodos Javascript não possuem acesso a disco para realizar upload via XMLHttpRequest.

**Arquivo:** JsWebGets/Widget/Form/Input/JsUpload.js

**Tipo de Objeto:** Interface – Form – Input

**Superclasse:** JsInput

**Quantidade de Instâncias Simultâneas:** n

**Métodos Públicos:**

**Métodos Herdados:**

*setValue(any value)*

Sem uso no contexto, pois é bloqueado por todos os navegadores por segurança.

*getValue(void)*

*setAttribute(str name, any value)*

*getAttribute(str name)*

*addItem(jsobj any)*

Sem uso no contexto.

*delItem(jsobj any)*

Sem uso no contexto.

*disable(bool value)*

*setMenu(jsobj menuobj)*

*setToolTip(str value)*

*setEvent(str eventtype, function)*

*unsetEvent(str eventtype, function)*

*disableEvent(str eventtype, function)*

*setXPos(int x)*

*setYPos(int y)*

*setHeight(int h)*

*setWidth(int w)*

*setClass(str classname)*

*setCursor(str cursortype)*

*setFont(str fontfamily)*

*setFontColor(str fontcolor)*

*setFontSize(str fontsize)*

*setFontWeight(str fontweight)*

*setFontStyle(str fontstyle)*