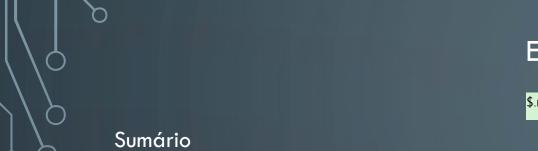


TODOS OS DIREITOS

- Fontes: https://notifyjs.jpillora.com/
- Tradução: Thiago dos Santos

- Lista de estilos
 - <u>Bootstrap</u>
 - <u>metro</u>



Exemplo

\$.notify("Hello World");



Notify.js é um plugin jQuery para fornecer notificações simples, mas totalmente personalizáveis. Os trechos de código javascript nesta documentação com a borda verde podem ser executados clicando neles.

USO BÁSICO

Notificações de elemento

Você pode colocar notificações em qualquer elemento:

Exemplo

\$(".elem-demo").notify("Hello Box");



USO BÁSICO

Notificações globais

Se você não especificar um elemento, a notificação aparecerá no canto superior esquerdo (a menos que você especifique uma posição diferente - consulte Posicionamento)

Exemplos

\$.notify("I'm over here !");

- I'm over here!
- ☑ I'm over here!
- I'm over here!
- I'm over here!

USO BÁSICO

Estilos de notificação

Cada estilo pode definir um conjunto de classes a serem usadas para substituir a notificação. A versão pré-empacotada inclui um estilo de notificação de inicialização (veja mais abaixo em Estilo). Essas classes incluem:

Exemplos

```
Success
$.notify("Access granted", "success");
Info
$.notify("Do not press this button", "info");
Warning
$.notify("Warning: Self-destruct in 3.. 2..", "warn");
Error
$.notify("BOOM!", "error");
Note: This page has set the default class to "success" with $.notify.defaults({ className: "success"});.
```

POSICIONAMENTO

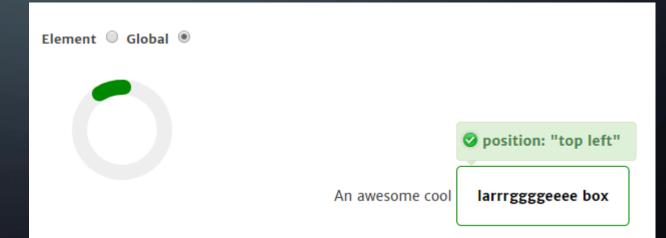
A opção string de posição é usada para descrever o alinhamento vertical e horizontal. As notificações de elemento e as notificações globais podem ser reposicionadas verticalmente em: "superior", "central" ou "inferior" e reposicionadas horizontalmente em: "esquerda", "central" ou "direita". Se não fornecermos uma opção de posição, os alinhamentos padrão serão definidos nas configurações padrão: globalPosition e elementPosition. Quando apenas um alinhamento é fornecido, o alinhamento vertical é o meio e o alinhamento horizontal é o centro.

Exemplos

\$(".pos-demo").notify("I'm to the right of this box", { position:"right" });

We can position the notification around this box V I'm to the right of this box

- Use esta ferramenta de posicionamento para ver todas as combinações possíveis de posições.
 - position: "top left"
 position: "top left"
 position: "top left"
 position: "top left"



<u>API</u>

O QUE É UMA API

Uma API (Application Programming Interface) é um conjunto de comandos, funções, protocolos e objetos que os programadores podem usar para criar softwares ou interagir com sistemas externos. Ela fornece aos desenvolvedores comandos para executar operações comuns para que eles não tenham que escrever um código do zero.

\$.notify(string|object, [options]) string|object - global notification data options - an options object or class name string

\$.notify(element, string|object, [options])
element - a jquery element
string|object - element notification data
options - an options object or class name string

\$(selector).notify(string|object, [options]) selector - jquery selector string|object - element notification data options - an options object or class name string

\$.notify.addStyle(styleName, styleDefinition) styleName - string (the style option references this name) styleDefinition - style definition object (see Styling below)

\$.notify.defaults(options)
options - an options object (updates the defaults listed below)

OPÇÕES

O objeto de opções listado acima está no formulário abaixo. Este objeto abaixo é o real usado para verificar a validade das opções e definir os padrões.

```
// whether to hide the notification on click
clickToHide: true,
// whether to auto-hide the notification
autoHide: true,
// if autoHide, hide after milliseconds
autoHideDelay: 5000,
// show the arrow pointing at the element
arrowShow: true,
// arrow size in pixels
arrowSize: 5,
// position defines the notification position though uses the defaults below
position: '...',
// default positions
elementPosition: 'bottom left', globalPosition: 'top right',
// default style style: 'bootstrap',
// default class (string or [string])
className: 'error',
// show animation
showAnimation: 'slideDown',
// show animation duration
showDuration: 400,
// hide animation
hideAnimation: 'slideUp',
// hide animation duration
hideDuration: 200,
// padding between element and notification
gap: 2
```

GUIA DE ESTILO PERSONALIZADO

Você pode adicionar seus próprios estilos ao Notify.js com o método \$.notify.addStyle. Veja API.

Os objetos de definição de estilo estão no formato:

GUIA DE ESTILO PERSONALIZADO

html - Se você possui apenas um elemento HTML que precisa modificar por notificação, deve atribuir a esse elemento um atributo de datanotify-text ou data-notify-html. Use data-notify-html se desejar exibir HTML arbitrário dentro da notificação; caso contrário, use data-notify-text, pois é mais seguro.

Caso contrário, se você desejar modificar vários elementos HTML por notificação, precisará atribuir a cada elemento um dos dois atributos acima, além de um valor. Para um exemplo disso, consulte o Exemplo avançado abaixo.

classes - Este objeto será usado para construir css. Ele deve estar no formato descrito acima e há um exemplo abaixo.

css - Essa string é simplesmente css bruto. Use essa propriedade se desejar manter sua definição de estilo dentro de seus arquivos js.

Convenção de nomenclatura de classe

Quando cada notificação é construída, seu contêiner (o elemento mais externo definido no seu estilo html) aplicará automaticamente a classe:

GUIA DE ESTILO PERSONALIZADO

notifyjs- <nome do estilo> -base

Quando você usa a opção de nome da classe (className), a classe:

notifyjs- <nome do estilo> - <nome da classe>

Serão aplicados. Portanto, se você definir seus estilos em um arquivo CSS externo ou na propriedade css do estilo, deverá definir suas regras CSS usando esta convenção de nomenclatura.

Exemplo Simples

• Se você definir o estilo:

Ao executar esse código, o objeto de classes neste exemplo será convertido no seguinte css:

```
.notifyjs-happyblue-base {
  white-space: nowrap;
  background-color: lightblue;
  padding: 5px;
} .notifyjs-happyblue-superblue {
  color: white;
  background-color: blue;
}
```

- Você pode confirmar isso agora inspecionando o DOM (procure o elemento de estilo marcado na cabeça)
- Agora você pode usar seu novo estilo com:

GUIA DE ESTILO PERSONALIZADO

Você pode confirmar isso agora inspecionando o DOM (procure o elemento de estilo marcado na cabeça)

Agora você pode usar seu novo estilo com:

e você pode usar a classe superblue com:

\$.notify('hello !!', { style: 'happyblue' });

⊕ hello ‼⊕ ⊕ hello ‼⊕

⊕hello ‼⊕

⊚hello‼⊚

⊚hello‼⊚

⊕ hello ‼⊕

⊕hello‼⊕

⊚hello‼⊚

⊕ hello ‼⊕

\$.notify('HELLO !!!!', { style: 'happyblue', className: 'superblue' });

⊕ HELLO !!!!⊕

⊕ HELLO !!!!⊕

⊕ HELLO !!!!⊕

⊕ HELLO !!!! ⊕

⊕ HELLO !!!! ⊕

e HELLO !!!!e

⊕ HELLO !!!!⊕

⊕ HELLO !!!!⊕

⊕HELLO !!!!⊕

⊕ HELLO !!!!⊕

⊕ HELLO !!!!⊕

⊕ HELLO !!!! ⊕

⊕ HELLO !!!!⊕

EXEMPLO AVANÇADO

Digamos que você queira usar botões nas suas notificações, então você pode fazer algo como:

```
//add a new style 'foo'
$.notify.addStyle('foo', {
html:
"<div>" +
 "<div class='clearfix'>" +
 "<div class='title' data-notify-html='title'/>" +
 "<div class='buttons'>" + "<button class='no'>Cancel</button>" +
 "<button class='yes' data-notify-text='button'></button>" +
 "</div>" + "</div>" + "</div>" });
//listen for click events from this style
$(document).on('click', '.notifyjs-foo-base .no', function() {
//programmatically trigger propogating hide event
$(this).trigger('notify-hide');
$(document).on('click', '.notifyjs-foo-base .yes', function() {
//show button text
alert($(this).text() + " clicked!");
//hide notification
$(this).trigger('notify-hide');
```

EXEMPLO AVANÇADO

Observe que não há propriedade de classes definida acima. Como o CSS neste exemplo não é trivial, usaremos um arquivo CSS externo:

Nota: você também pode converter esse código CSS em uma string JavaScript e usálo com a propriedade css. Não é muito legível, embora seja melhor para distribuição.

```
.notifyjs-foo-base {
opacity: 0.85;
width: 200px;
background: #F5F5F5;
padding: 5px;
border-radius: 10px;
 .notifyjs-foo-base .title {
width: 100px;
float: left; margin: 10px 0 0 10px;
 text-align: right;
.notifyjs-foo-base .buttons {
width: 70px;
float: right;
font-size: 9px;
padding: 5px;
margin: 2px;
notifyjs-foo-base button {
font-size: 9px;
padding: 5px;
margin: 2px;
width: 60px;
```

EXEMPLO AVANÇADO

Agora você pode usar este estilo com:

```
$.notify({
  title: 'Would you like some Foo ?',
  button: 'Confirm'
}, {
  style: 'foo',
  autoHide: false,
  clickToHide: false
});
```



EXEMPLO EXTRA

Se o uso dos métodos acima ainda não fornecer o que você precisa, pode passar os objetos jQuery diretamente para a notificação (desde que o elemento tenha o atributo data-notify-html):

```
var h5 = $("<h5/>").append("You MUST have some Foo !")
$.notify({
  title: h5,
  button: 'YES !'
}, {
  style: 'foo',
  autoHide: false,
  clickToHide: false
});
```



CONTRIBUINDO COM MAIS ESTILOS

Contribuir com mais estilos é fácil!

Clique neste link

Entre no GitHub (se você ainda não tiver)

Clique em 'Fork'

Digite seu novo estilo

Clique em 'Confirmar novo arquivo'

Clique em 'Enviar solicitação de recebimento'!

- Para iniciar as pessoas com algumas idéias de estilos, confira este post em vários estilos de Growl.
- Nota: Se você deseja contribuir com a biblioteca corrigindo erros ou adicionando recursos, consulte a pasta src. Além disso, a fonte está no CoffeeScript, no entanto, novos estilos podem ser JavaScript ou CoffeeScript. Estou usando o Grunt para compilar, minificar e concatenar as compilações. Você pode fazer isso com:
- Fork https://github.com/notifyjs/notifyjs
- cd notifyjs
- npm install -g grunt-cli
- npm install
- Grunt
- Pull-request