



gNotify let's you send Notifications from your Xojo Web Application.

gNotify is a pure javascript/css solution and is compatible with all major recent browsers.

The demo project demonstrates most gNotify features.

If you build the demo project (or your application using gNotify from demo project) the Notifications will be displayed but the titles and texts will be overridden.

You need to buy gNotify to build your own application.

What are Notifications ?

Notifications are information messages displayed over the current page. Notifications texts, colors and icons are informing the user that something happened.

With gNotify these messages can be triggered and controlled by your application.

Features

- Elegant, responsive, flexible notifications,
- Lightweight, pure javascript and CSS, NO dependencies
- Use one of 5 pre-defined styles (info, success, warning, error, question)
- Or create your own style at runtime
- Define Title & Message Texts, Color and Size
- Set Notification timeout
- Illustrate Timeout with a Progress Bar
- Define Progress Bar color, optional pause or reset on hover
- Define Buttons and be informed when these are clicked
- Optional Close icon, close on escape
- Use FontAwesome (free) icons
- Use Google Material icons
- Overlay support (creates a modal notification)
- Right-to-left (RTL) support

System requirements

Xojo release 2016r2.1 and following.

Web Applications only (no Desktop, Raspberry or iOS application).



Licensing

gNotify is developed by Gate61 see <http://gate61.com> for contacts
gNotify is released under BSD license, Copyright © 2018, Olivier Colard - Gate61
All rights reserved.

gNotify is based on dolce/iziToast (see <http://izitoast.marcelodolce.com> and <https://github.com/dolce/iziToast>)
dolce/iziToast is licensed under the Apache License 2.0

Xojo is © 2018 Xojo, Inc.



Installation

gNotify is delivered as a project containing the gNotify controls.
gNotify classes are encrypted or not, depending on your buying option.

To install gNotify in your application, proceed as follow:

- create or open your own Xojo Web project
- open the gNotify demo project
- copy the gNotifyFolder
- paste the gNotifyFolder within your own project

To use gNotify

- drag the gNotify class to your WebPage or a WebContainer (*)

(*) Remark:

You can place gNotify within a WebContainer but the Notifications are always displayed on the WebPage at the Notification defined position.

gNotify does not show in the IDE Editor, it's a « tray » control.
There are no IDE options.

gNotifyFolder contains the following:

- gNotify (WebPage control)
- gNotifyConfig (configuration class)
- gNotifyGlobal (module)



Creating a Notification

Notifications essentially have:

- Title
- Message

Optionally have:

- Icon
- Buttons

You can create as much notifications as you want.

You can create your own « predefined » notifications and reuse them multiple times.

A Notification is defined by creating a **gNotifyConfig instance**.

A Notification is displayed by calling the gNotify instance **Notify** method or one of the method for pre-defined notification styles.

For example

gNotify1 is an instance of gNotify placed on my WebPage.

To create a simple « Hello World » Notification with pre-defined « info » style:

```
Dim notif As New gNotifyConfig
notif.titleText = "Hello"
notif.messageText = "World"
gNotify1.Info(notif)
```



gNotifyGlobal

If you plan to use FontAwesome or Google Material Icons, you **must activate** them by setting the following gNotifyGlobal properties to True:

- gNotifyUseFontAnwesome
- gNotifyUseMaterialIcons

These options must be set **before the page is displayed** the first time.

gNotify

Methods

`Notify(config As gNotifyConfig)`

Display the Notifications as defined by the config properties.

You can also display a notification using one of the 5 pre-defined styles:

`Info(config As gNotifyConfig)`

`Success(config As gNotifyConfig)`

`Warning(config As gNotifyConfig)`

`Error(config As gNotifyConfig)`

`Question(config As gNotifyConfig)`

Each of these methods will display the Notifications as defined by the gNotifyConfig instance properties but **override** the backgroundColor and icon settings with the predefined settings.

Events

`ButtonClicked(notificationID As String, buttonCaption As String)`

A button was clicked.

This event is not raised when the « close » button (see properties) is clicked

`Opened(notificationID As String)`

Notification is displayed.

`Closed(notificationID As String)`

Notification is closed.



gNotifyConfig

Pre-Defined Styles

Pre-Defined Styles	
Info	 gNotify a Notification control for Xojo Web Applications ×
Success	 gNotify a Notification control for Xojo Web Applications ×
Warning	 gNotify a Notification control for Xojo Web Applications ×
Error	 gNotify a Notification control for Xojo Web Applications ×
Question	 gNotify a Notification control for Xojo Web Applications ×

Icons

Pre-defined icons:

- ico-info
- ico-success
- ico-warning
- ico-error
- ico-question

Font Awesome icons

Only « free » FontAwesome icons are available.
Icon styles are restricted to Solid and Regular.
Icon list can be found here: <https://fontawesome.com/icons?d=gallery&s=regular,solid&m=free>

See demo project (wDemo, init & dspFeatures method) for FontAwesome examples.

Material Design icons

Icon list can be found here: <https://material.io/icons/>

See demo project (wDemo, showTitle method) for Material Icon example.



Properties

`animateInside` (Boolean = True)

Animate the Text and Message when displaying.

`backgroundColor` (Color)

Background color of the Notification.

`className` (String)

Use if you want to add your own class to the DOM element.
Set by gNotify but not used.

`closeOnEscape` (Boolean = False)

When True, the Notification will be closed when the user will press the Escape key.

`closeOnOverlayClick` (Boolean = False)

When True, the Notification will be closed when the user click somewhere on the Overlay around the Notification.
Ignored when `withOverlay = False`.

`ID` (String)

Unique ID of the Notification.
You need it if you want to close the Notification by program.
Read-only property.

`maxWidth` (Integer)

Maximum Notification width in pixels.

`messageColor` (Color)

Color of the message Text.

`messageFontSize` (Integer)

Message Font size in px.

`messageLineHeight` (Integer)

Message Line Height in px.

`messageColor` (Color)

Color of the message Text.

`messageText` (String)

Text of the message.

`overlayColor` (Color)

Color of the Overlay around the Notification.
Ignored if `withOverlay = False`.



`pauseProgressOnHover` (Boolean = True)
Pause timeout when mouse over the Notification.
Ignored if timeout is disabled (timeout = 0)
Ignored if `resetProgressOnHover` = True

`position` (String = topRight)
Position of the Notification.
Possible values are:
- bottomRight
- bottomLeft
- topRight
- topLeft
- topCenter
- bottomCenter
- center

`progressBarColor` (Color)
Color of the Progress Bar.
Ignored if `withProgressBar` = False.

`raiseCloseEvent` (Boolean = False)
Closed event raised only if True.

`raiseOpenEvent` (Boolean = False)
Opened event raised only if True.

`resetProgressOnHover` (Boolean = False)
Reset timeout when mouse over the Notification.
Ignored if timeout is disabled (timeout = 0)

`timeout` (Integer = 5000)
Timeout in milliseconds.
Timeout is disabled when 0.

`titleColor` (Color)
Color of the title Text.

`titleFontSize` (Integer)
Title Font size in px.

`titleLineHeight` (Integer)
Title Line Height in px.

`titleColor` (Color)
Color of the title Text.

`titleText` (String)
Text of the title.



`transitionIn (String)`

Notification animation when opening.

Values are `bounceInLeft`, `bounceInRight`, `bounceInUp`, `bounceInDown`, `fadeIn`, `fadeInDown`, `fadeInUp`, `fadeInLeft`, `fadeInRight` or `flipInX`.

`transitionOut (String)`

Notification animation when closing.

Values are `fadeOut`, `fadeOutUp`, `fadeOutDown`, `fadeOutLeft`, `fadeOutRight`, `flipOutX`.

`withBalloon (Boolean = False)`

Adds a balloon.

Balloon color matches the background color.

`withCloseButton (Boolean = True)`

Adds a close button.

`withOverlay (Boolean = False)`

Adds an full page overlay around the Notification.

Looks like a « modal » Notification.

`withProgressBar (Boolean = True)`

Display a Progress Bar.

Ignored if timeout disabled.

`zIndex (Integer = 9999)`

zIndex of the Notification.

Must be high enough to have the Notification displayed on top of everything.



Methods

`addButton(caption As String, hasFocus As Boolean, optional closeOnClick As Boolean = True)`

Adds a button to the Notification.

Only one button can have `hasFocus = True`.

When `CloseOnClick` is `True` (default) the button will raise the `ButtonClicked` event then the Notification will close.

`setIcon(icon As String, iconType As Integer, optional iconStyle As String)`

`icon`

Is the name of the icon, can be a pre-defined icon, a FontAwesome icon, a Google Material Icon

`iconType`

Values are:

0 - pre-defined icon (or constant `gNotify.gNotifyIconTypePreDefined`)

1 - Font Awesome icon (or constant `gNotify.gNotifyIconTypeFontAwesome`)

0 - Material Icon icon (or constant `gNotify.gNotifyIconTypeMaterialIcon`)

`iconStyle`

For Font Awesome icons only (ignored for other types).

Values are:

« fas » for Solid (or constant `gNotify.gNotifyFontAwesomeStyleSolid`)

« far » for Regular (or constant `gNotify.gNotifyFontAwesomeStyleRegular`)



Support

Support and bug corrections will be provided exclusively by mail, on best effort. Please use the following address for a better support: support@gate61.com

Support is limited to gNotify components.

Updates

When you purchase a license you are enlisted to receive futures updates for free. You will get an email when a new version is published.

Limitations

gNotify has been written using Xojo 2017r2.1 and tested from 2016r2.1 upwards.