GENESYS™

API Reference

Notification Service REST API

4/28/2025

Notification Service REST API

Contents

Description

The Notification Service REST API is enables you to keep users of your website engaged and informed. You can reach your entire user base quickly and effectively with notifications that are delivered to your web pages. The Notification Service is responsible for delivering the push messages (events) to the Browser Tier. On the Browser Tier, each message is handled by the Notification Agent (a module of the Tracker Application). The Notification Agent provides a list of predefined messages that can be used in engagement scenarios. Each message uses the "gpe." prefix in the channel name.

The Notification Service REST API has two methods:

- Notify by Visit ID — delivers notifications from the GWE Server to the browser tier based on the visit ID (through CometD).

- Send Disposition Code for Invitation from Browser to GWE Server — delivers notification disposition (result) from the browser tier to the GWE Server (through HTTP GET).

Notify by Visit ID

Description

This method performs an engagement notification through the Web Engagement Server by visit ID. If the request is performed correctly, then the Web Engagement Server creates and sends a notification message to the CometD /notification/{visitID} channel.

Request

Method	POST
Body	JSON array of Notification Messages
URL	http://<gwe_server_host:gwe_server_port>/server/data/notification/notify/{visitID} The HTTPS schema is also allowed, if configured.
Parameters	

Name	Value	Mandatory	Description
visitID	string	yes	The visit ID for the engagement notification.

Response

Response	Standard HTTP Responses
Response Type	none

Example

Request

```
curl -X POST \
  -H "Content-Type: application/json" \
  -d '[{"page": "654E0122924F42EA82B5C581A394555D", "channel":
"gpe.appendContent", "data": { "url": "http://www.genesys.com/invite.html",
"method": "GET", "container": "body" }}]' \
  http://server:port/server/data/notification/notify/654E0122924F42EA82B5C581A394555D
```

Notification Messages

The notification request body is represented by an array of JSON objects, one object for each notification message:

```
[
    {
        // message 1
    },
    {
        // message 2
    },
    ...
]
```

Message Structure

Notification messages should follow this format:

```
{
    "page":"yyy",
    "channel":"any-string",
    "data":{
        "someKey":"someValue"
        ...
    }
}
```

Option	Type	Mandatory	Default Value	Description
page	string	yes	undefined	The page attribute can have the following values:

Option	Type	Mandatory	Default Value	Description
				• **<pageID>** — The notification message is only received by the page with the unique page ID. For example, **654E0122924F42EA82B5C5** • **all** — The notification message is received by all pages. • **active** — The notification message is received only by the page with focus in the browser.
channel	string	yes	undefined	The channel name. See Standard Messages for details.
data	object	no	undefined	Channel-specific data. See Standard Messages for details.

Standard Messages

The following messages are standard in Genesys Web Engagement.

gpe.appendContent

With this message, you can append content retrieved by **url** to the **container** element.

Options

Option	Type	Mandatory	Default Value	Description
url	string	yes	undefined	The path of the resource to load. This option can have either an absolute

Option	Type	Mandatory	Default Value	Description
				(http://www.genesys.com/invite.html) or relative value (/invite.html). Note: The relative value is relative to the Web Engagement Server.
method	string	no	"JSONP"	Specifies the type of request to make: • GET — You should only use the GET method if the resource is located on the same server, in order to avoid issues with cross-origin resource sharing. • JSONP — For the JSONP method, the server you point to in the url option must support the JSONP protocol.
container	string	no	"body"	The jQuery selector for the loaded content container. The loaded content is appended to the current element.

Example

The following message uses the AJAX GET request to load data from the server.

```
{
    "page": "654E0122924F42EA82B5C581A394555D",
    "channel": "gpe.appendContent",
    "data": {
        "url": "http://www.genesys.com/invite.html",
        "method": "GET",
        "container": "body"
    }
```

```
}
```

gpe.setVariable

You can use this message to set the variable on the page in the window context.

Options

Option	Type	Mandatory	Default Value	Description
variable	string	yes	undefined	The global variable name. The variable name could contain a namespace, for example: `com.service.param` — this object will be available by `window.com.service.param`.
value	string	no	undefined	The variable value.

Example

This example sets the variable `window.serviceData` to the object `{ 'name': 'Pat', 'message': 'Hello world!' }`. If the variable is already defined, then it is extended.

```
{
    "page": "654E0122924F42EA82B5C581A394555D",
    "channel": "gpe.setVariable",
    "data": {
        "variable": "serviceData",
        "value": {
            "name": "Pat",
            "message": "Hello world!"
        }
    }
}
```

gpe.callFunction

This message calls the corresponding function on the page.

Options

Option	Type	Mandatory	Default Value	Description
function	string	yes	undefined	The name of the function. The name can contain the namespace of the function. For example, `com.service.alert` — this means that

Option	Type	Mandatory	Default Value	Description
				`window.com.service.alert()` will be called.
arguments	array	no	undefined	The arguments to pass to the function. This should always be an array.

Example

This example calls the `window.reactToServerMessage('Hello world!', 1)` function.

```
{
  "page": "654E0122924F42EA82B5C581A394555D",
  "channel": "gpe.callFunction",
  "data": {
    "function": "reactToServerMessage",
    "arguments": ["Hello world!", 1]
  }
}
```

Using the API to Customize Widgets

Deprecation notice

- **Starting with the 8.5.000.38 release of Genesys Web Engagement, Genesys is deprecating the Native Widgets—and the associated APIs (the Common Component Library)—in preparation for discontinuing them.**

  This functionality is now available through a single set of consumer-facing digital channel APIs that are part of Genesys Mobile Services (GMS), and through Genesys Widgets, a set of productized widgets that are optimized for use with desktop and mobile web clients, and which are based on the GMS APIs.

  Genesys Widgets provide for an easy integration with Web Engagement, allowing you to proactively serve these widgets to your web-based customers.

  ### Important

  Although the deprecated APIs and widgets will be supported for the life of the 8.5 release of Web Engagement, Genesys recommends that you move as soon as you can to the new APIs and to Genesys Widgets to ensure that your functionality is not affected when you migrate to the 9.0 release.

  - Note that this support for the Native Widgets and the associated APIs will not include the addition of new features and that bug fixes will be limited to those that affect critical functionality.

You can also use the Notification Service REST API to customize the out-of-the-box Genesys Web Engagement invitation widgets. See the following sections for details:

- Chat Invitation Message
- Ads Message

Chat Invitation Message

The chat invitation message is a JSON object that is sent by Orchestration Server to the browser. The chat invitation is represented by two serial messages: `gpe.setVariable` and `gpe.appendContent`.

For example:

```
[
    {
        "page":"654E0122924F42EA82B5C581A394555D",
        "channel":"gpe.setVariable",
        "data":{
            "variable":"com.genesyslab.gpe.invite.data",
            "value": {
                "type":"chat",
                "engagementID":"21def142-5ba9-4d64-8cd4-83d3c9d78ba9",
                "subject":"Chat",
                "message":"Good evening! Would you like some help with the selection? Our
technical experts are available to answer questions.",
                "acceptBtnCaption":"Chat",
                "cancelBtnCaption":"No Thanks",
                "inviteTimeout":30,
                "modal":false,
                "chatOptions": {
                    "serverUrl": "http://demosrv.genesyslab.com:9081/server/cometd",
                    "widgetUrl": "/server/api/resources/v1/chatWidget.html",
                    "registration":true,
                    "embedded": false,
                    "userData": {
                        "visitID":"3ea3efab-ca33-4383-acbd-90720db72288"
                    }
                }
            }
        }
    },
    {
        "page":"654E0122924F42EA82B5C581A394555D",
        "channel": "gpe.appendContent",
        "data": {
            "url": "/server/api/resources/v1/invite.html",
        }
    }
]
```

You can use the Notification Service REST API to modify the chat invitation, which uses `gpe.setVariable` to set the `com.genesyslab.gpe.invite.data` variable to a list of options. You can update these options to customize the chat invitation. See the table below for details.

Field	Type	Mandatory	Default Value	Description
type	string	yes	undefined	The type of engagement (typically, "chat").
engagementID	string	no	undefined	The unique identifier of the proactive engagement offer (part of the notification message).
subject	string	no	"Chat", if the type is set to "chat".	The invitation widget title.
message	string	no	"Hello! Would you like some help with the selection? Our technical experts are available to answer questions."	The main message in the invitation widget.
acceptBtnCaption	string	no	"Chat", if the type is set to "chat".	The title of the accept invitation button.
cancelBtnCaption	string	no	"No Thanks"	The title of the cancel invitation button.
inviteTimeout	Number	no	30	The timeout (in seconds) for the invitation widget. After this time interval, the invitation widget is automatically closed.
modal	Boolean	no	false	Specifies whether to show the invitation widget as a modal dialog.
chatOptions	object	yes	undefined	This object is passed to the chat application in the startChat method. See Chat JS Application API for details.

Ads Message

You can use the gpe.appendContent message to load the **ads.html** file.

```
[
    {
```

```
    "page": "654E0122924F42EA82B5C581A394555D",
    "channel": "gpe.appendContent",
    "data": {
        "url": "/server/api/resources/v1/ads.html"
    }
  }
]
```

Send Disposition Code for Invitation from Browser to GWE Server

Description

The disposition code lets Web Engagement know the status of the invitation: whether it was accepted, rejected, or ignored. Based on this status, GWE decides what it should do with the invitation. The browser side (invitation widget) sends the disposition code to the Web Engagement Server with a GET request.

Request

Method	GET
URL	http://<gwe_server.host:gwe_server.port>/server/data/invites/?result=<result code>ŋagementID=<ID of proactive engagement offer>&pageID=<ID of page where invite appeared>&visitID=<current visit ID>[&media=<selected media>] The HTTPS schema is also allowed.

Parameters			
Name	**Value**	**Mandatory**	**Description**
result	string	yes	A string that describes the result type. The following values are allowed: • accept • timeout • pageExit • cancel
engagementID	string	no	The unique identifier of the proactive engagement offer (part of the notification message). If this parameter is not specified, Web Engagement will only be able to process one engagement request for

Method	GET		
			the current page instance.
pageID	string	yes	The ID of the page where the invitation widget appeared.
visitID	string	yes	The current visit ID.
media	string	no	The media for which the invitation was applied.

Response

Response	Standard HTTP Responses 200 - OK 400 - FAULT
Response Type	none

Example

The code below shows an example of how to send the disposition code from the browser to GWE.

```
_gt.push(['getIDs', function(IDs) {
    $.ajax({
      type: "GET",
      url: "http://server:port/server/data/invites",
      data: {   result: "accept",
                engagementID: "21def142-5ba9-4d64-8cd4-83d3c9d78ba9",
                media: "chat",
                visitID: IDs.visitID,
                pageID: IDs.pageID
          }
    })
      .done(function( msg ) {
        // Data has been saved
      });
}]);
```

> ### Tip
> You can use the Monitoring JS API to access the visitID and pageID parameters.