



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

API Reference

Notification Service REST API

Notification Service REST API

Contents

- **1 Notification Service REST API**
 - **1.1 Description**
 - **1.2 Notify by Visit ID**
 - **1.3 Notification Messages**
 - **1.4 Standard Messages**
 - **1.5 Using the API to Customize Widgets**
 - **1.6 Send Disposition Code for Invitation from Browser to GWE Server**

Description

The Notification Service REST API 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	<p>http://<gwe_server_host:gwe_server_port>/server/data/notification/notify/{visitID}</p> <p>The HTTPS schema is also allowed, if configured.</p>		
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":
"ge.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
				<ul style="list-style-type: none">• <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"	<p>Specifies the type of request to make:</p> <ul style="list-style-type: none">• 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: <code>com.service.param</code> — this object will be available by <code>window.com.service.param</code> .
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, <code>com.service.alert</code> — this means that

Option	Type	Mandatory	Default Value	Description
				<code>window.com.service.alert()</code> 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 Chat and Callback Widgets—and the associated APIs (the Common Component Library)—in preparation for discontinuing them in the Genesys Engagement Manager 9.0 release.**

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 (which will become Genesys Engagement Manager in the 9.0 release), 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 Chat and Callback 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.
- As mentioned above, all support for the deprecated widgets and APIs will be dropped as of the 9.0 release of Genesys Engagement Manager.

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](#)
- [Callback 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/resources/chatWidget.html",
          "registration": true,
          "embedded": false,
          "userData": {
            "visitID": "3ea3efab-ca33-4383-acbd-90720db72288"
          }
        }
      }
    }
  },
  {
    "page": "654E0122924F42EA82B5C581A394555D",
    "channel": "gpe.appendContent",
    "data": {
      "url": "/server/resources/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 ("chat" or "callback").
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 <code>startChat</code> method. See Chat JS Application API for details.

Callback Invitation Message

The callback invitation message is a JSON object that is sent by Orchestration Server to the browser. The callback 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": "callback",
        "engagementID": "21def142-5ba9-4d64-8cd4-83d3c9d78ba9",
        "subject": "Voice",
        "message": "Good evening! Would you like some help with the selection? Our technical experts are available to answer questions.",
        "acceptBtnCaption": "Call Me",
        "cancelBtnCaption": "No Thanks",
        "inviteTimeout": 30,
        "modal": false,
        "callbackOptions": {
          "callbackUrl": "http://demosrv.genesyslab.com:9081/server",
          "widgetUrl": "/server/resources/callback.html",
        }
      }
    }
  },
  {
    "page": "654E0122924F42EA82B5C581A394555D",
    "channel": "gpe.appendContent",
    "data": {
      "url": "/server/resources/invite.html",
    }
  }
]
```

You can use the Notification Service REST API to modify the callback 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 callback invitation. See the table below for details.

Option	Type	Mandatory	Default Value	Description
type	string	yes	undefined	The type of engagement ("chat" or "callback").
engagementID	string	no	undefined	The unique identifier of the proactive engagement offer (part of the notification message).
subject	string	no	"Callback", if the	The invitation

Option	Type	Mandatory	Default Value	Description
			type is set to "callback".	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	"Callback", if the type is set to "callback".	The title of the accept invitation button.
cancelBtnCaption	string	no	"No Thanks"	The title of the cancel invitation button.
inviteTimeout	Integer	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.
callbackOptions	object	yes	undefined	The configuration options for the callback widget.
callbackUrl	string	yes	undefined	The URL of the callback server. For instance, 198.51.100.12:9081/server
widgetUrl	string	yes	undefined	The URL of the callback server. For instance, 198.51.100.12:8081/resources/callback.html

Ads Message

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

```
[
  {
    "page": "654E0122924F42EA82B5C581A394555D",
    "channel": "gpe.appendContent",
    "data": {
```

```
    "url": "/server/resources/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	<p>http://<gwe_server.host:gwe_server.port>/server/data/invites/?result=<result code>&agementID=<ID of proactive engagement offer>&pageID=<ID of page where invite appeared>&visitID=<current visit ID>&alias=<alias of the GWE Server>[&media=<selected media>]</p> <p>The HTTPS schema is also allowed.</p> <div><p>Important</p><p>The alias field is deprecated in 8.5 and will be removed in 9.0</p></div>		
Parameters			
Name	Value	Mandatory	Description
result	string	yes	<p>A string that describes the result type. The following values are allowed:</p> <ul style="list-style-type: none">• accept• timeout• pageExit• cancel• acceptChat (deprecated)• acceptCallback (deprecated)
engagementID	string	no	The unique identifier of the proactive engagement offer (part of the notification)

Method	GET		
			message). If this parameter is not specified, Web Engagement will only be able to process one engagement request for the current page instance.
pageID	string	yes	The ID of the page where the invitation widget appeared.
visitID	string	yes	The current visit ID.
alias	string	no	Alias of the GWE Server. Used for Load Balancing. Tracker script holds this parameter in the <code>com.genesyslab.wme.tracker.server</code> cookie. <div>Important The alias field is deprecated in 8.5 and will be removed in 9.0</div>
media	string	no	The media for which invitation was applied. For example, this parameter is helpful for combined invitations (chat/web callback).

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,  
        alias: IDs.alias  
    }  
  })  
  .done(function( msg ) {  
    // Data has been saved  
  });  
});
```

Tip

You can use the [Monitoring JS API](#) to access the visitID and pageID parameters.