Service Management UI Help

Genesys Mobile Engagement 8.5.2

5/7/2025

Table of Contents

Service Management Help

Genesys Mobile Services (GMS) includes a Service Management User Interface, where you can manage services and also access dedicated interfaces for Callback and Context Services.

Before you can access this UI, first install GMS. See the deployment instructions here and configure the new panels in the features section of your GMS configuration.

Note that, to work properly, this UI requires access to the following URLs:

```
http://<GMS Local Host>:8080/genesys/1/admin/*
```

Make sure to enable this access through your firewall and security if needed.

Monitor
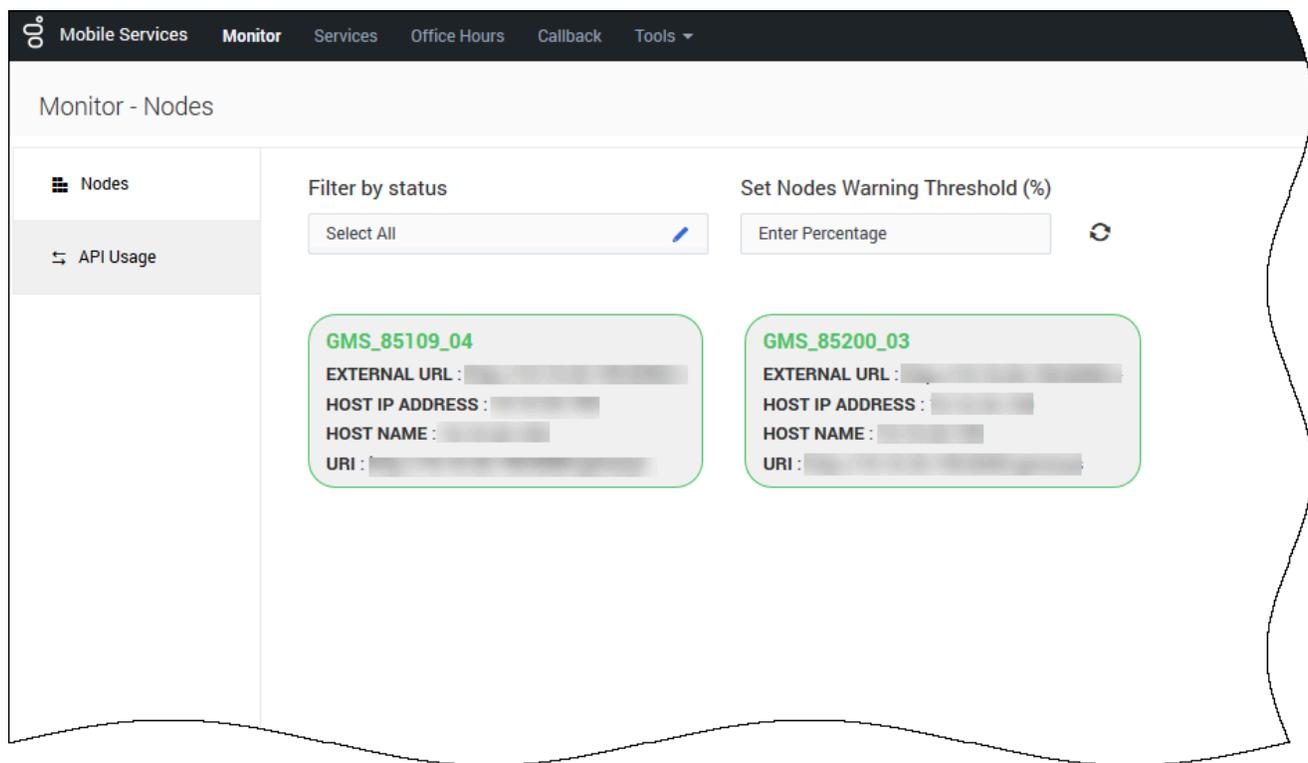
The **Monitor** tab (former Admin UI) and the **Tools** menu give you access to an administrator console, where you can:

- Monitor your GMS nodes
- Load and manage service templates
- Create resource groups and patterns
- Run reports
- Access samples
- Download DFM files
- Retrieve snippets for your `jetty-http.xml` file
- Manage the exception list for Callback services

Services
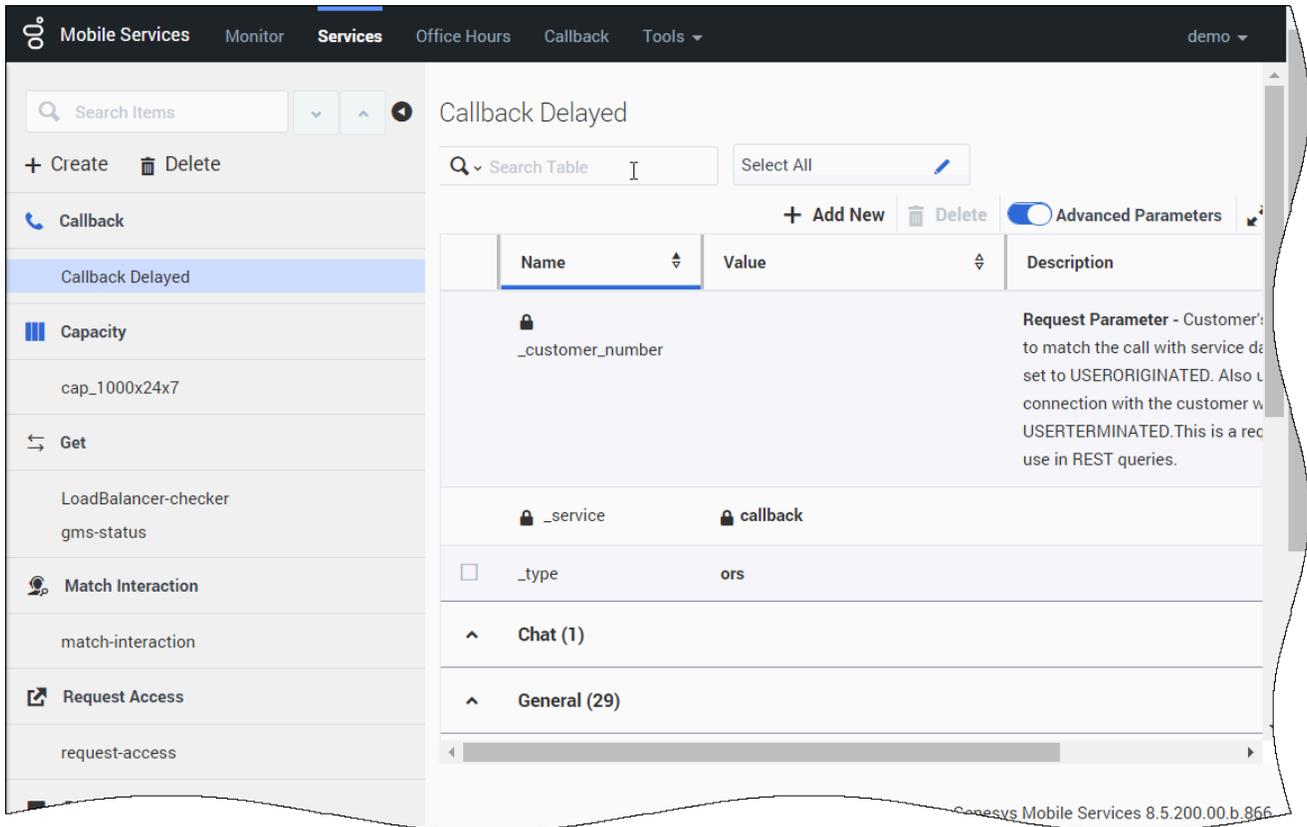
The **Services** tab is the user-friendly, web-based interface that is intended to be used by administrators and supervisors to:

- Add, delete, and modify a GMS service Agent Capacity

Office Hours
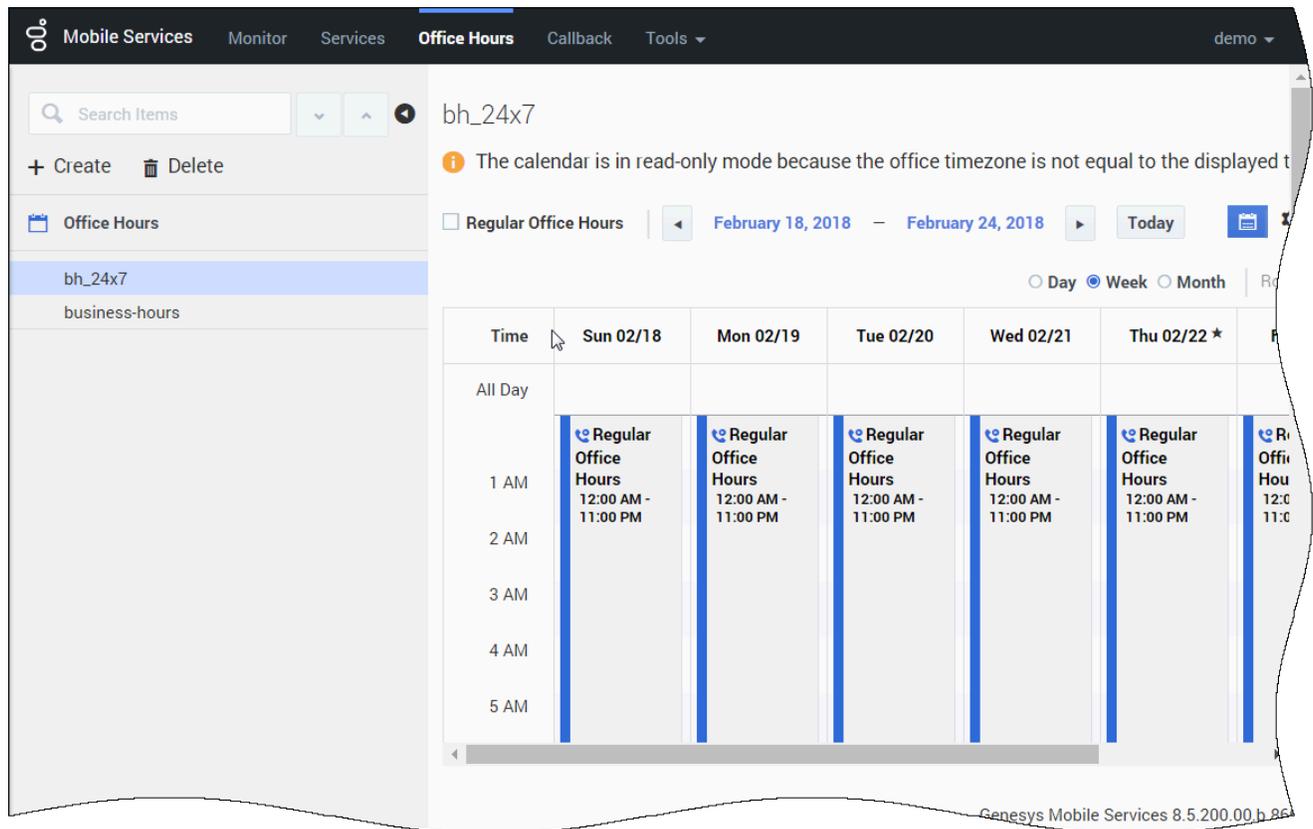
The **Office Hours** tab is the user-friendly, web-based interface that is intended to be used by administrators and supervisors to:

- Manage Call Center Office Hours and holiday schedules.

Callback
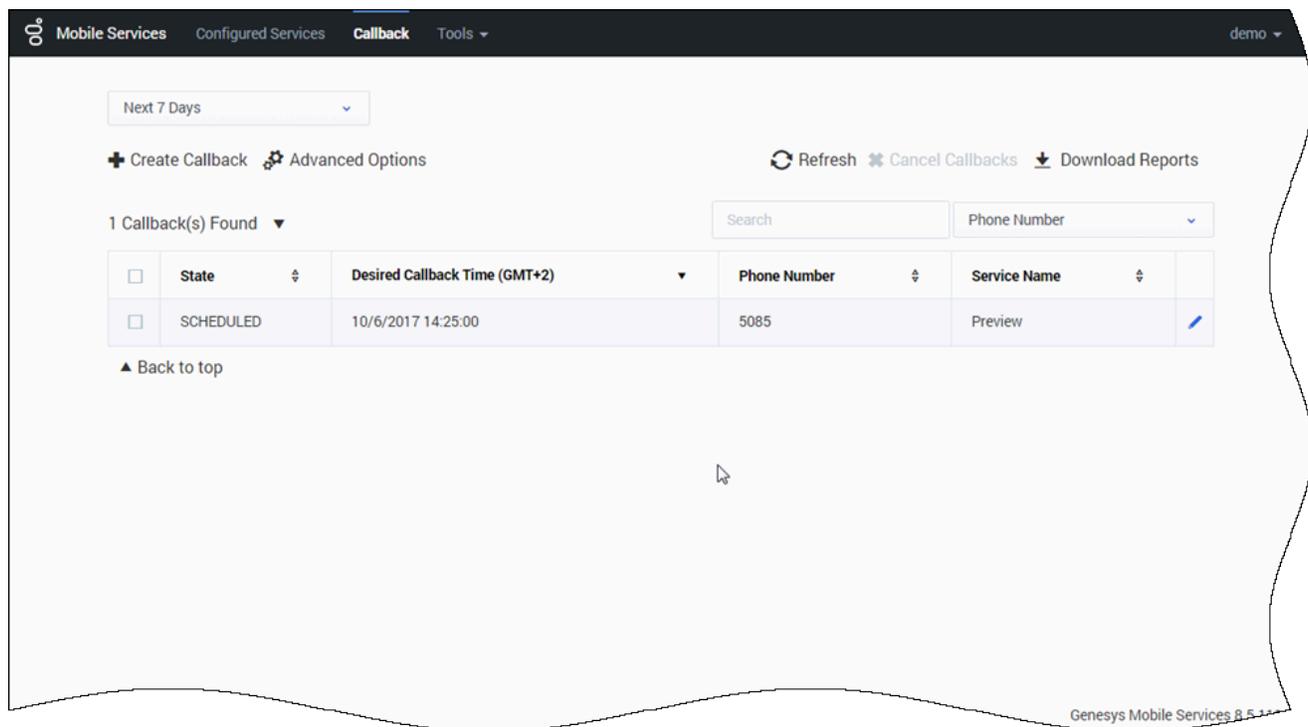
If you installed and configured Callback, this interface enables you to:

- Create a Callback record
- Manage your callbacks

Learn About Scenarios

GMS provides service templates and their scenarios that you can load in the Mobile Engagement UI, and then access through REST queries. All callback related scenarios are detailed in the Callback Solution Guide.

Enable Logging in the UI

The Service Management UI can provide detailed logs by configuring the following options in the GMS Configuration:

- enable-logger = `true` to activate DEBUG and INFO traces in the console.
- enable-logger-error = `true` to activate ERROR traces in the console.

Help

Introduced in 8.5.202
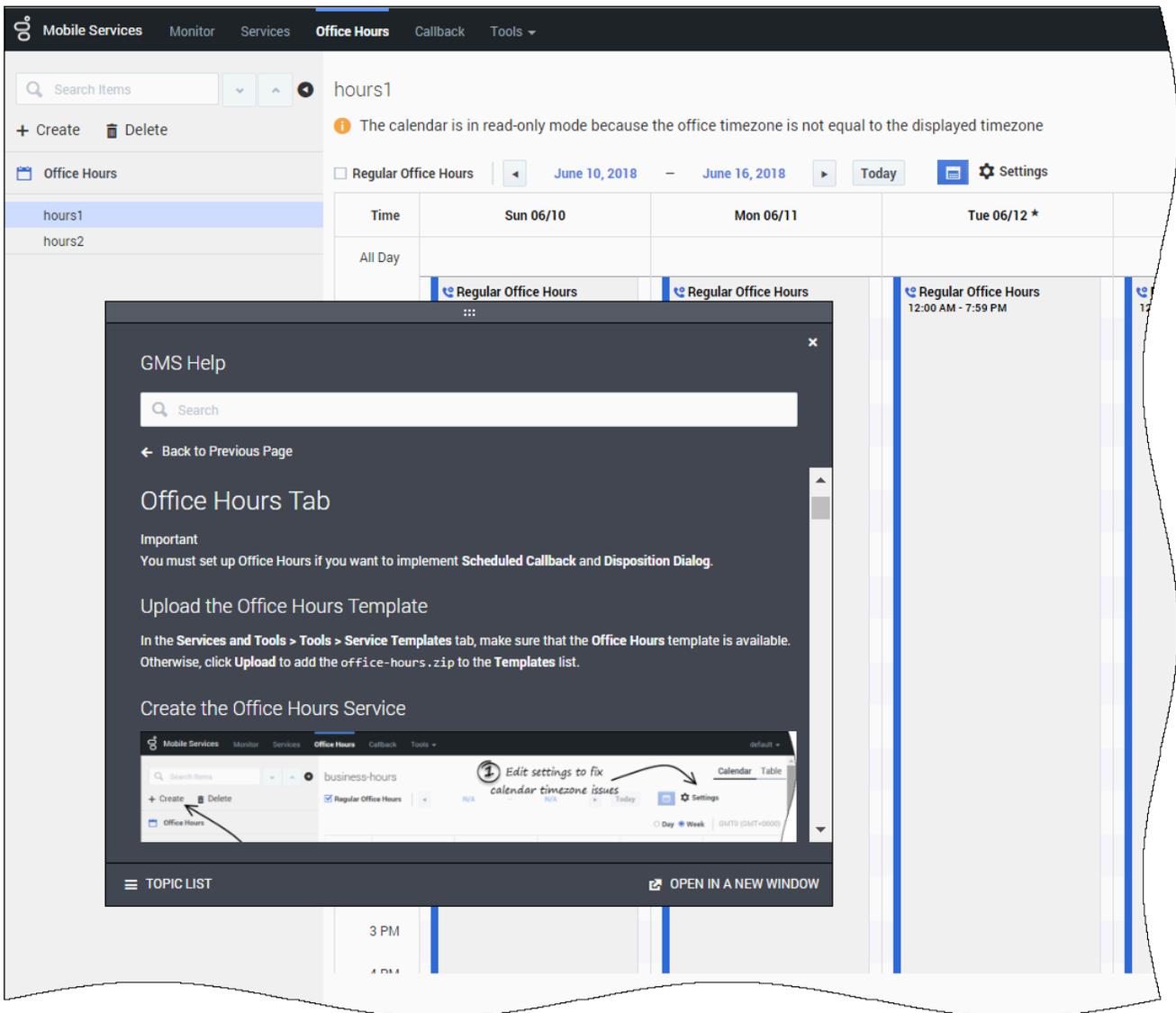
If you configure enable-contextual-help= true, the ![icon] icon of the Service Management UI opens a Contextual Help panel. This panel displays help pages, including videos and images, related to the active panel in the UI. It also includes search features and related topics.

Login, Roles, and Permissions

Set permissions for your Administrator

Updated in 8.5.207
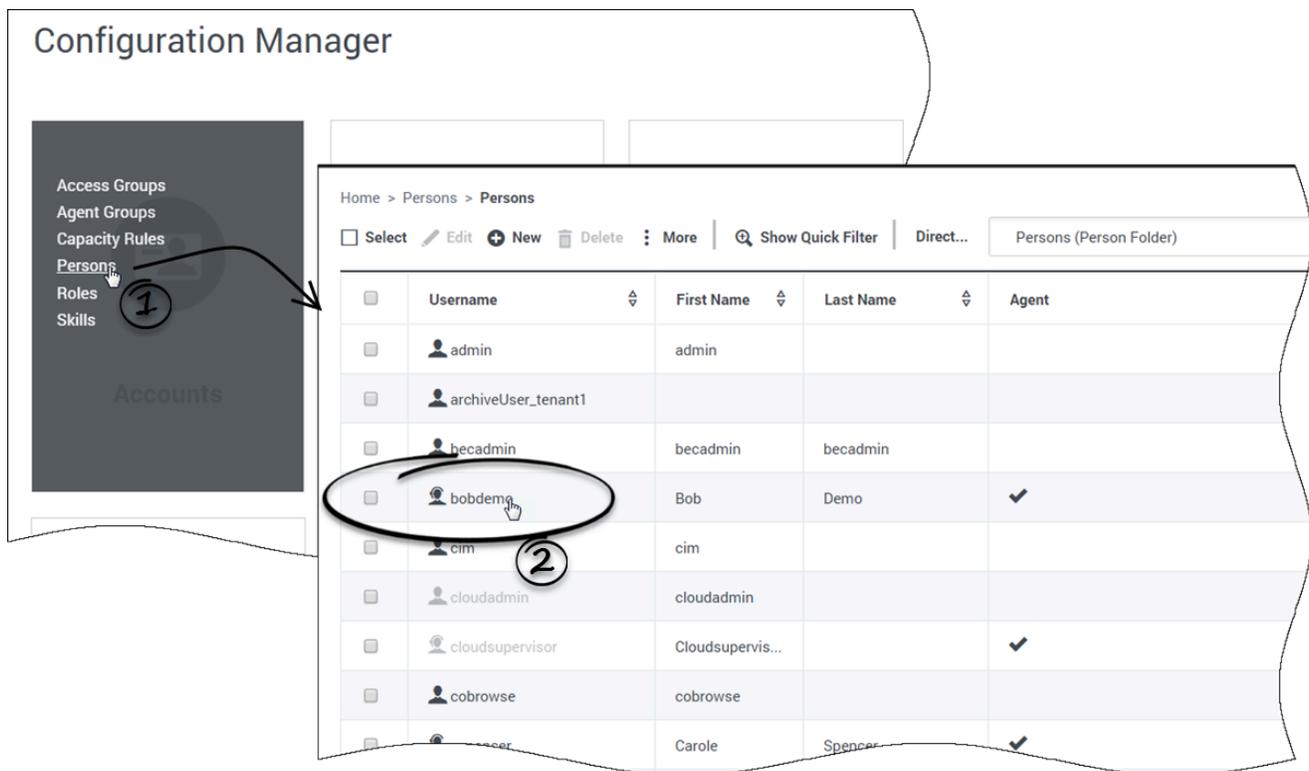
Open GAX and select the **Configuration Manager** section. In the **Accounts** menu, select **Persons** to get the list of configured users.

Edit the person who will be logging into the Service Management UI. You are going to give this user the permissions to read/write data into Callback related configuration objects (for example, GMS Application, Business Attributes, Transaction Lists for Resources/Patterns, and so on).
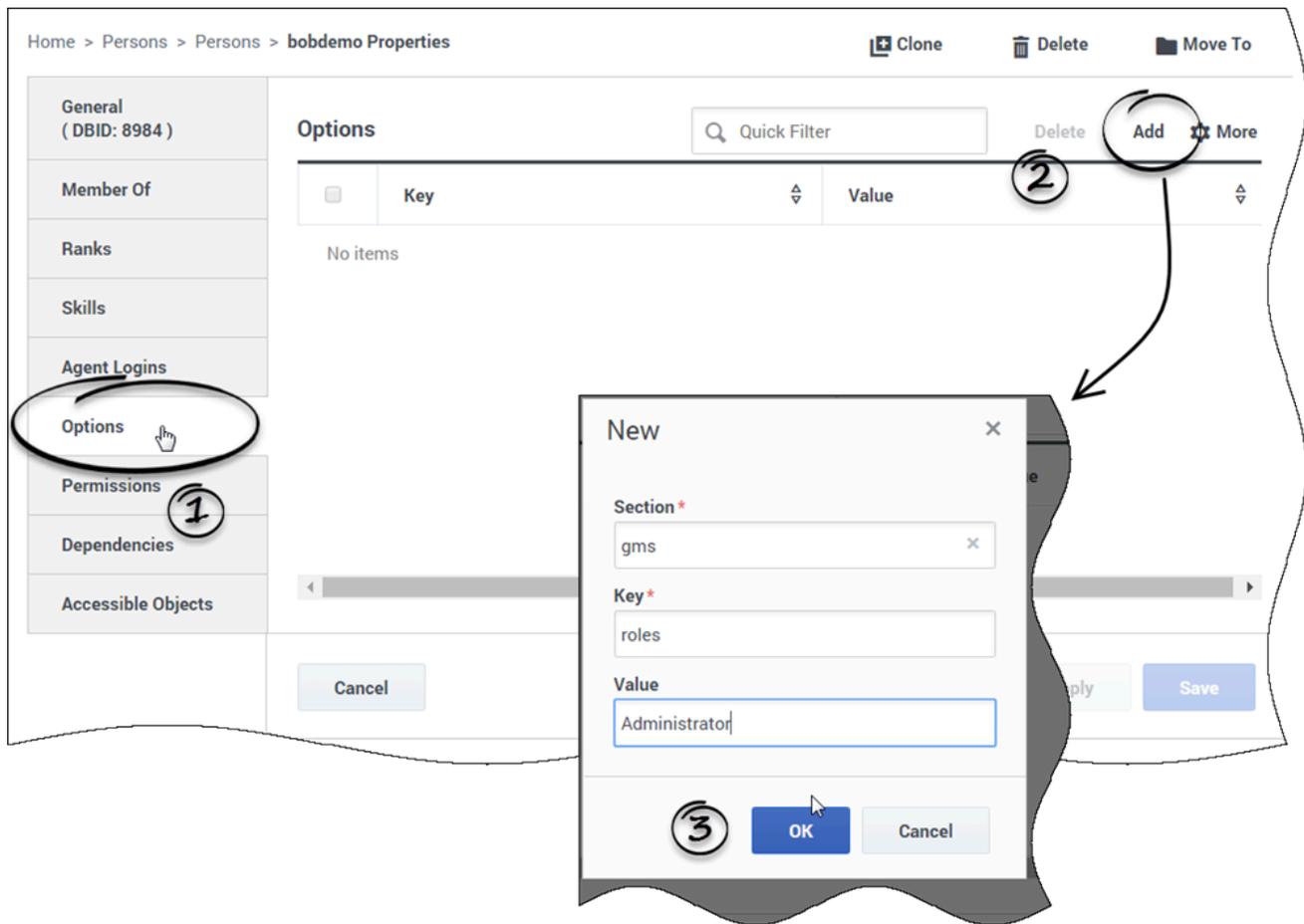
In the **Options** tab, click **Add** to set the following options:

1. Add the `gms/roles` option:
   - Enter gms for **Section**,
   - Enter `roles` for **Key**,
   - Enter one of the following values:
     - `Supervisor`: Role used to monitor and configure Callbacks only.
     - `Administrator`: Role used to administer the creation of Callback Services. This role provides access to all panels and includes the **Supervisor** role.
     - `CallbackReadOnly`: Role used to provide read-only access. The user can see the Callback panel without `Create`, `Update`, or `Delete` capabilities. This role was introduced in 8.5.226.03.

2. (Optional) Add the `gms/services` option:
   - Enter gms for **Section**,
   - Enter `services` for **Key**,
   - For the value, you can enter:

- **Exact matching names**–Enter the comma-separated list of services that the administrator is allowed to monitor. For example, "callback-support, callback-sales, callback-blackfriday-support".

- **A virtual service group name**–Enter a virtual service group name. For example, 'support'. To add services to a given virtual group, add the **_service_groups** option to your services and enter a comma-separated list of group names.

  For example, if you add the **_service_groups** = 'support' to the `callback-support` and `callback-blackfriday-support` services, these services will be displayed to the agent.

  Note that if you defined several virtual groups for your service, you need to add only one virtual groupe name to your user permissions.

Login

You can log into the UI hub at this URL: `http://<GMS Local Host>:8080/genesys`

Browser support is listed in the Genesys Supported Operating Environment Reference Guide.
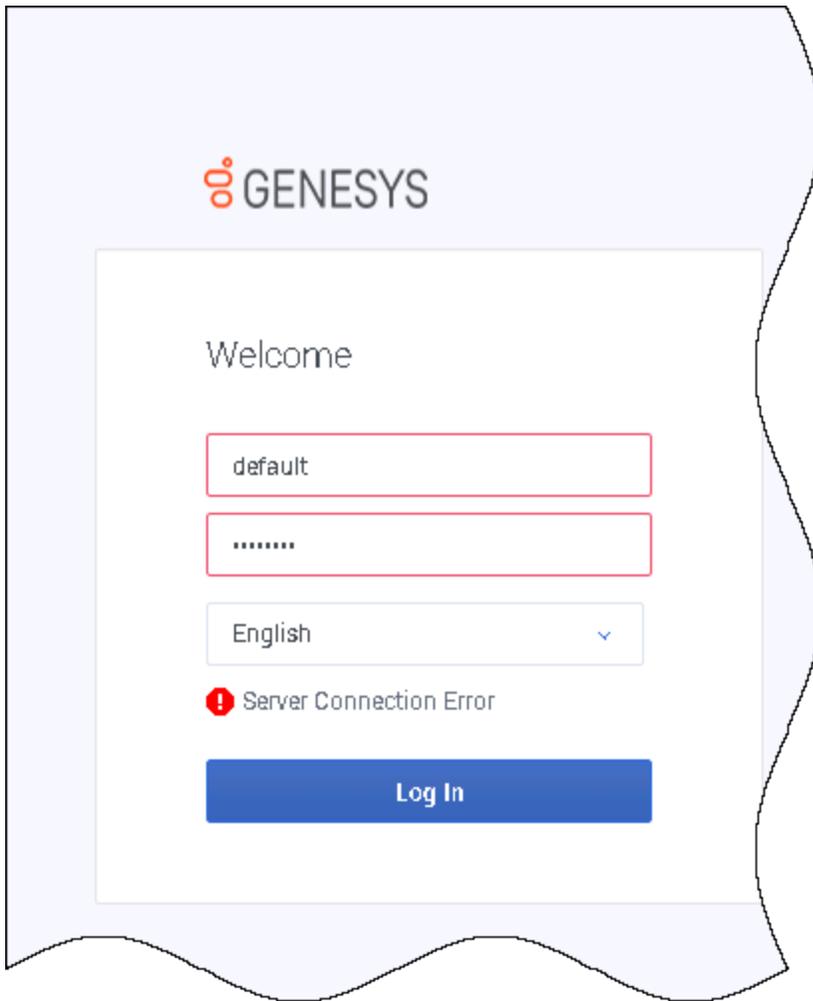
If no GMS is running, you get a **Server Connection Error**.

Services and Tools UI
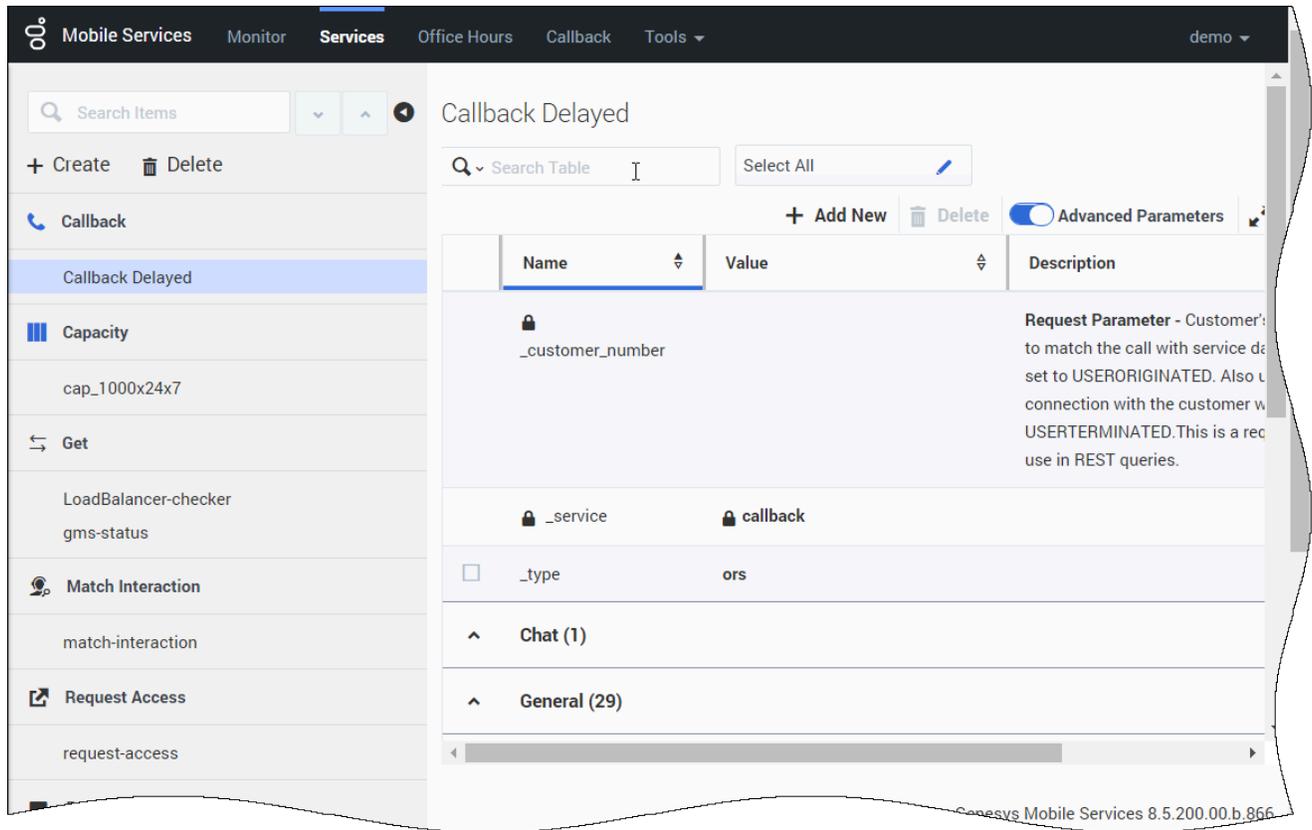
The **Services and Tools** Interface is a user-friendly, web-based interface for administrators and supervisors. Use this interface to manage and customize your Services, Office Hours, and Service templates. It provides functionality formerly available in the **Admin UI > Configured Services** tab.

To access this interface, you must log in as a user who owns the Administrator or Supervisor role. Then, select the **Services and Tools** icon and switch to the **Services** or **Tools** tab.

How to Implement a Service

1. Check the list of templates available in the Tools > Service Templates tab. If your template is not available, upload it.

2. Create your service in the **Services** tab.

3. Configure your service.

You can now start to send requests to this service.

Monitor Tab

Access Monitor Tab
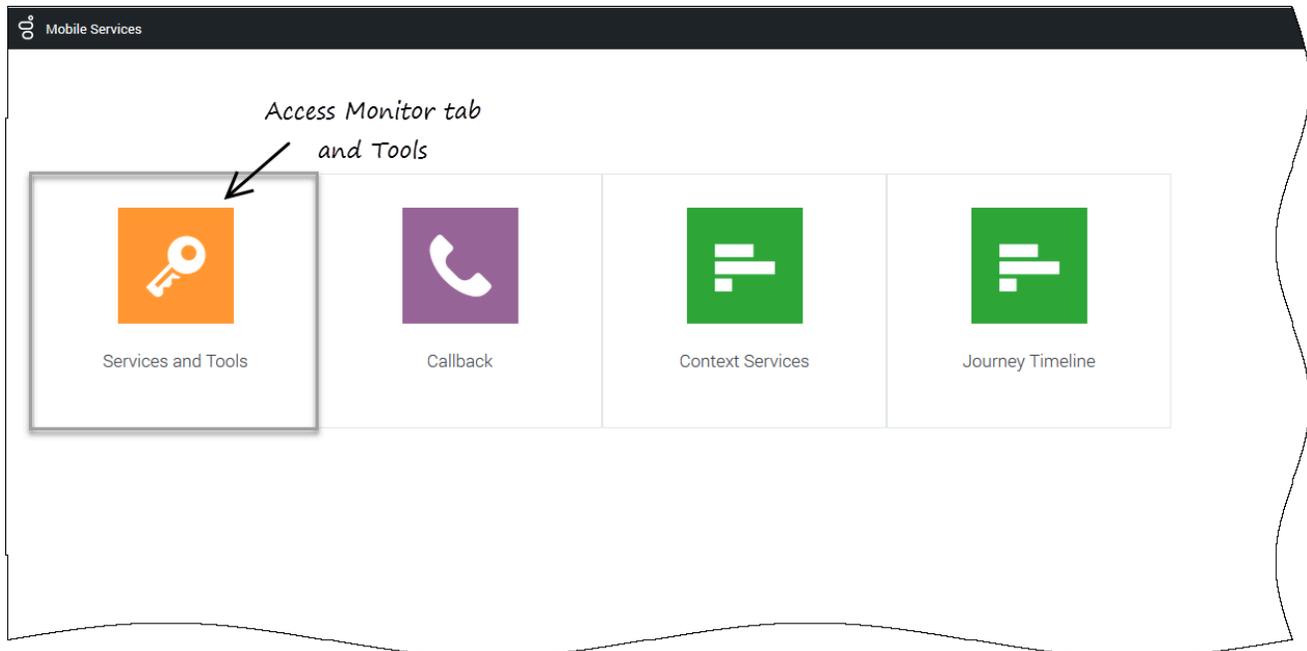
To access this interface, you must log in as a user who owns the Administrator or Supervisor role. Then, you can select the Admin UI icon.

> ### Important
>
> To make sure that the UI displays the right data of the GMS nodes, you need to consider some use cases and configuration options in your GMS application. See the options reference for details.

Monitor tab
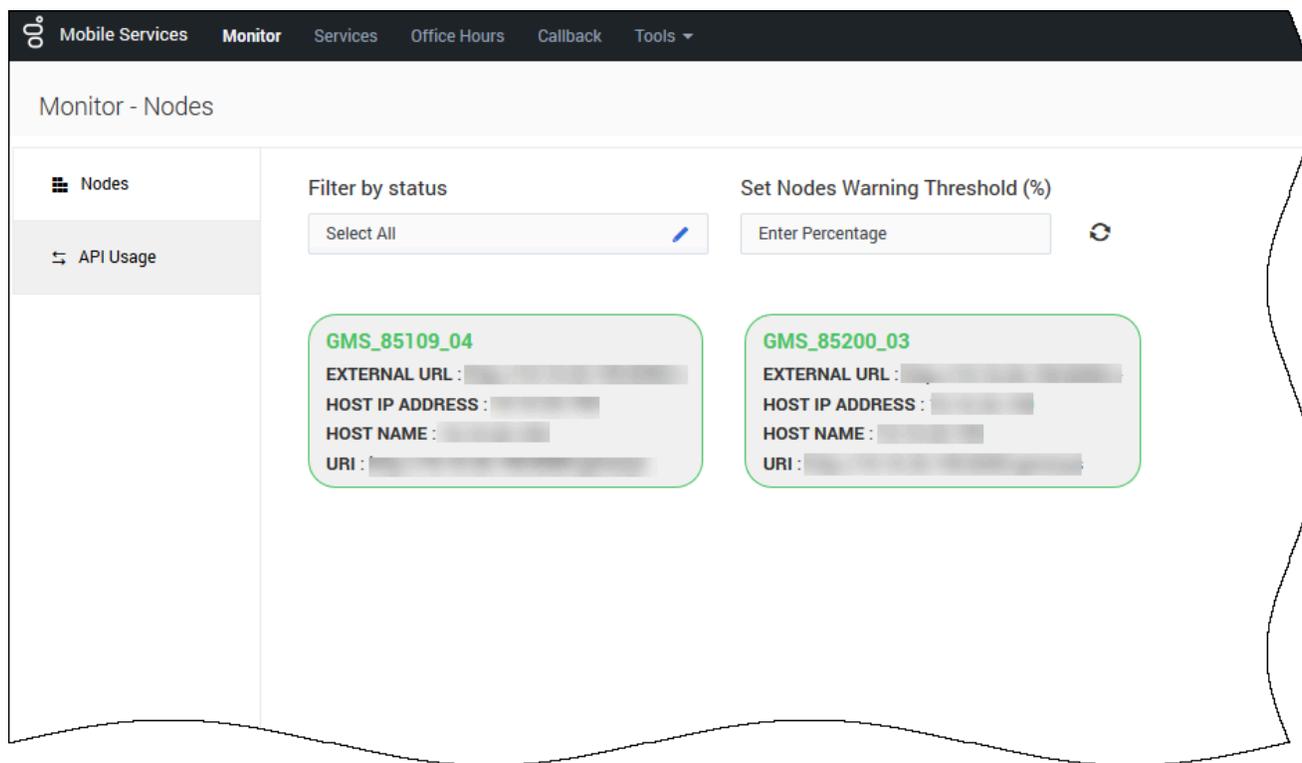
The **Monitor** tab displays the current health of the GMS nodes.

In this example (left), you can see several running GMS nodes. A green bar shows the system status for that node is up. A red bar means the system status for that node is down.

The *Monitor* screen refreshes every three minutes.

Services Tab

Create Services
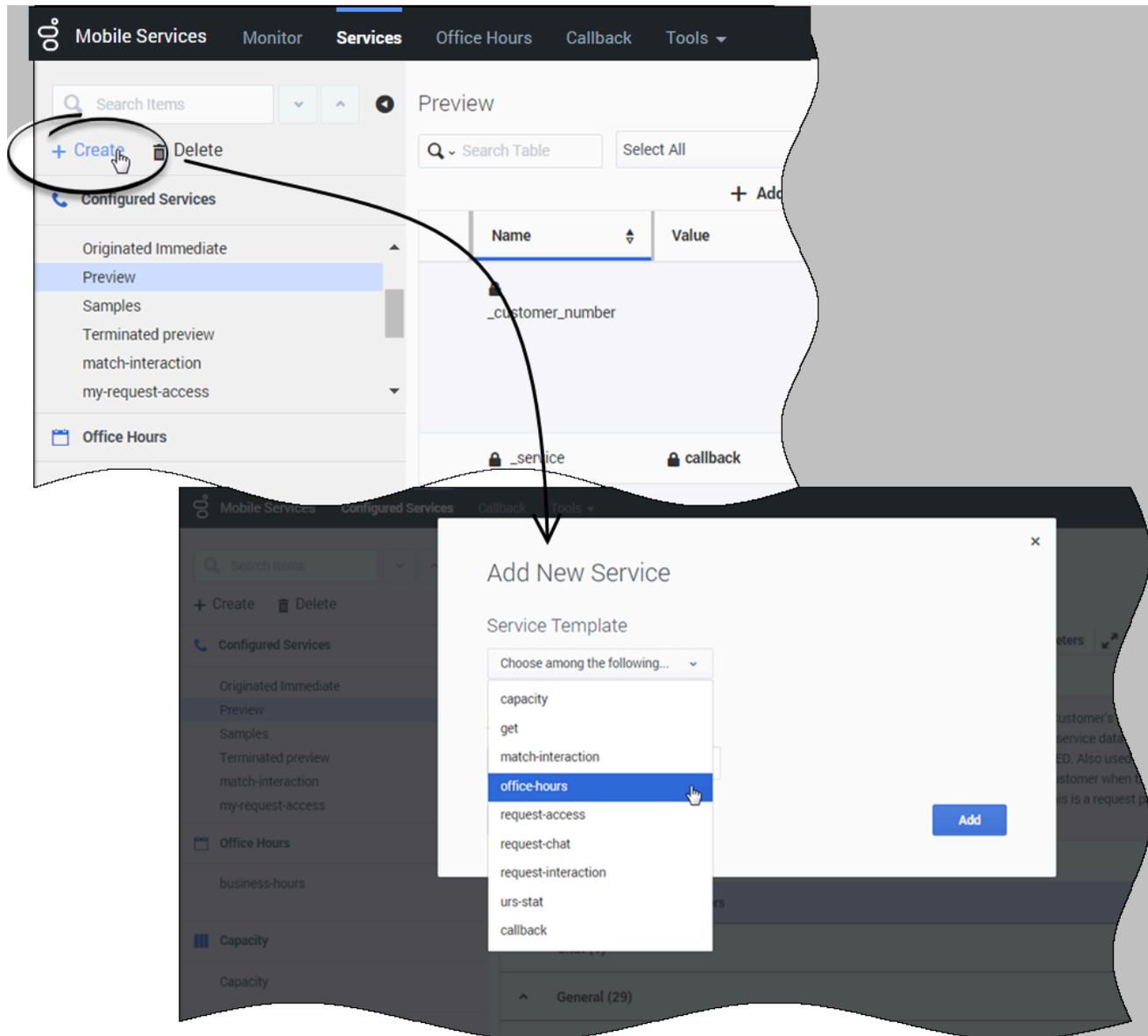
After you load the template, you can create a new service. The loaded service templates are available in the drop-down list on the creation form.

Select a template and fill in the form. Based on this information, the interface pre-populates the service parameters that match a scenario configuration and grant usage of GMS APIs.

> ### Important
>
> Learn which template to use for given scenarios and APIs here. Read more about `builtin` and `ors` services here.
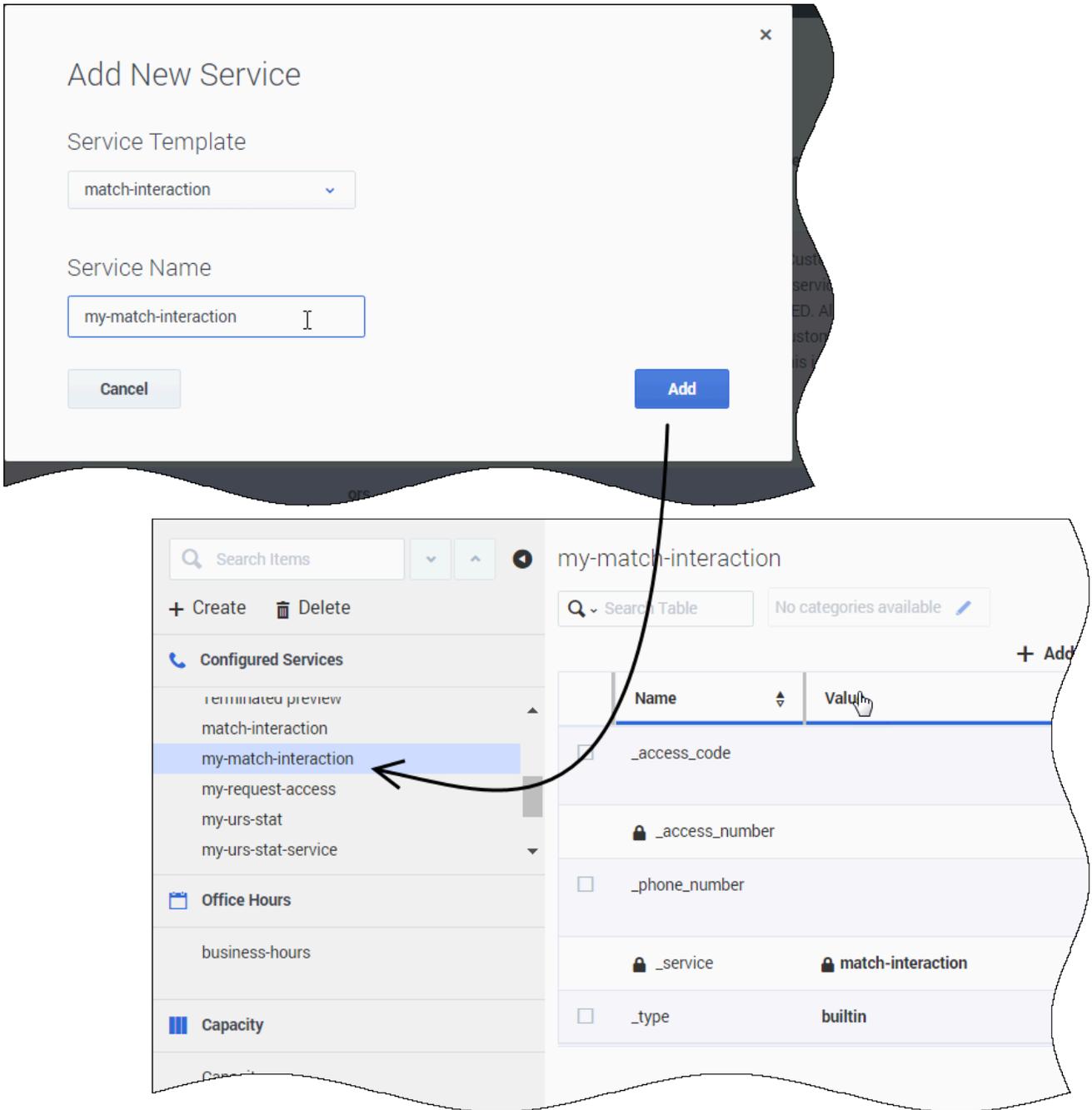
Click the **Create** button and fill in your service information. Select your template, then click **Save**.
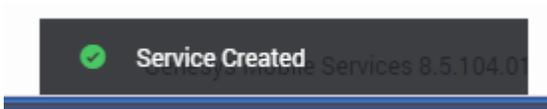
An information message confirms the service creation.

The new service appears in the list of Configured Services. You can now configure your service.

The service is also created in the `service.{service-execution-name}` section of your GMS configuration. The URLs used by the Service API are dependent on the name of the service that you have just created. Services are available at the following URL:

`http://host:port/genesys/1/service/{service-execution-name}`

For instance, if you create a service named `match-interaction`, then `{service-execution-name}` is `match-interaction` and the service is available at:

`http://host:port/genesys/1/service/match-interaction`

Important

To use a service, start by allocating resources to this service with a create service request. Note that for some `builtin` services, this may not be necessary.

Configure your Service

Key parameters for the service are automatically populated with the appropriate default values.

Important

For cluster configurations, all changes made in a service will be replicated into the entire GMS cluster.

The parameters have the following characteristics:

Mandatory Parameters
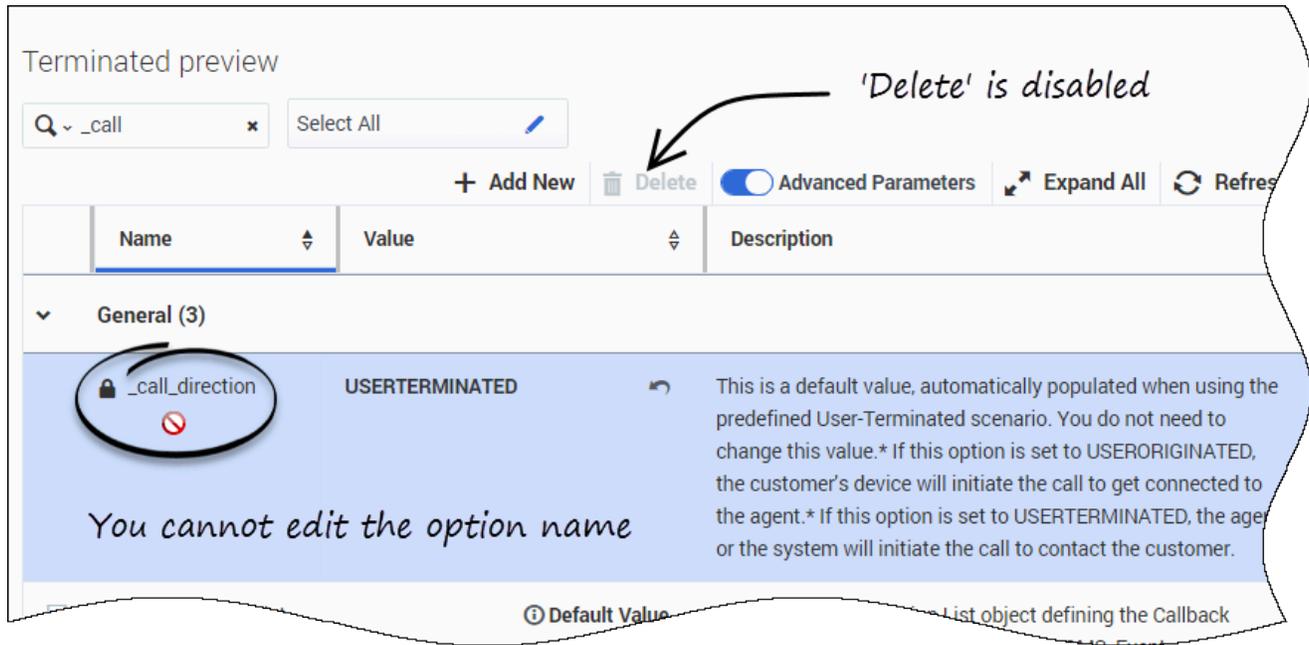
Mandatory parameters are identified with a 🔒 icon; you cannot rename or remove them.

Optional Parameters
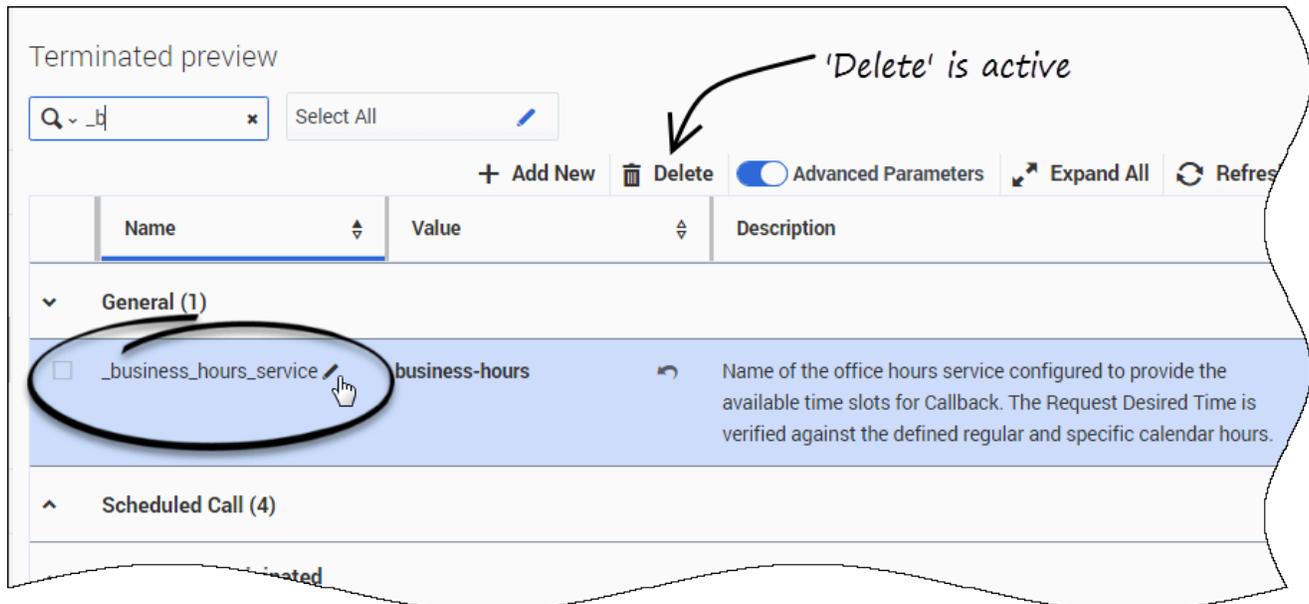
Optional parameters are identified with a ✎ icon when hovering; you can rename, edit, and remove

them.

Advanced Parameters
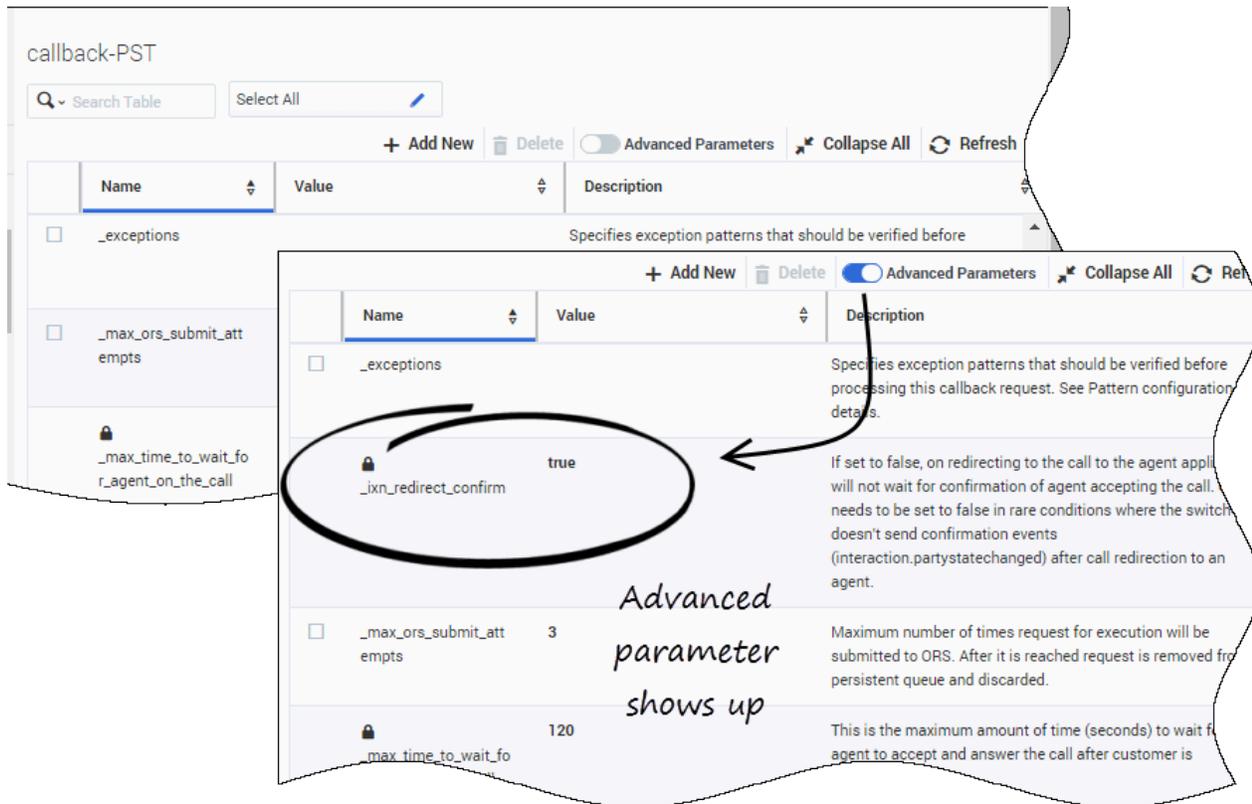
Advanced parameters are mandatory parameters used for advanced customization purposes. By default, they are hidden, but you can display them by enabling the **Advanced Parameters** selector.

Request Parameters
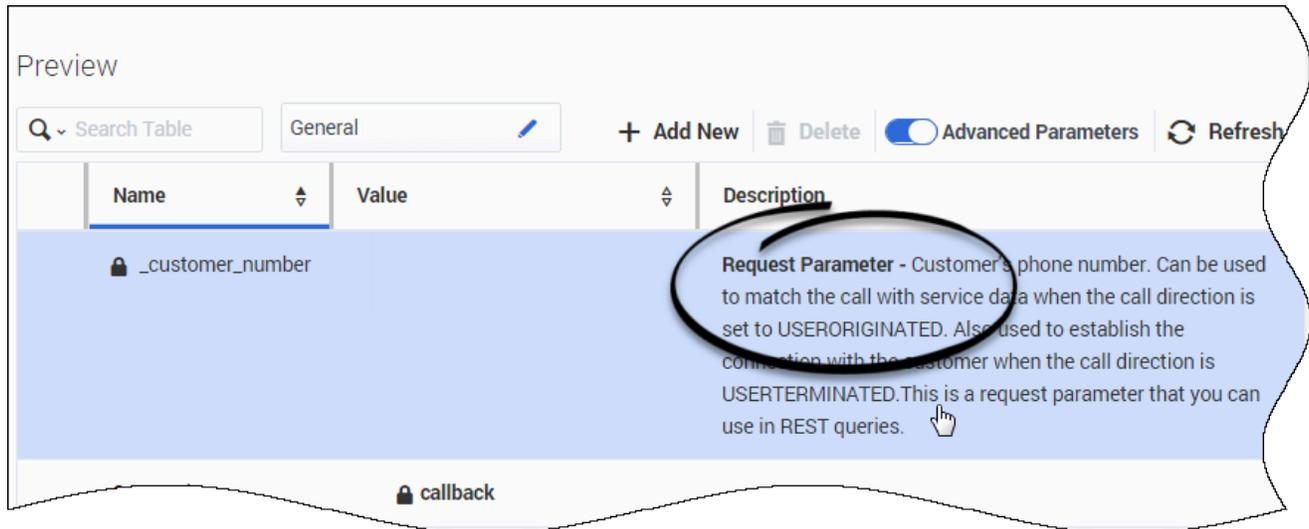
Request parameters are identified in the Description. You can use these parameters in your queries to the Callback Services API.

Edit Values
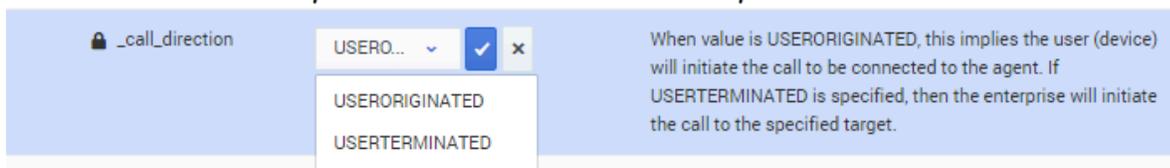
Editable values are identified with a ✎ icon when hovering over them. Just click to edit the field.

The interface will display pre-defined values if they exist, and you can read the **Description** field for more information about the option.

Add New Parameters
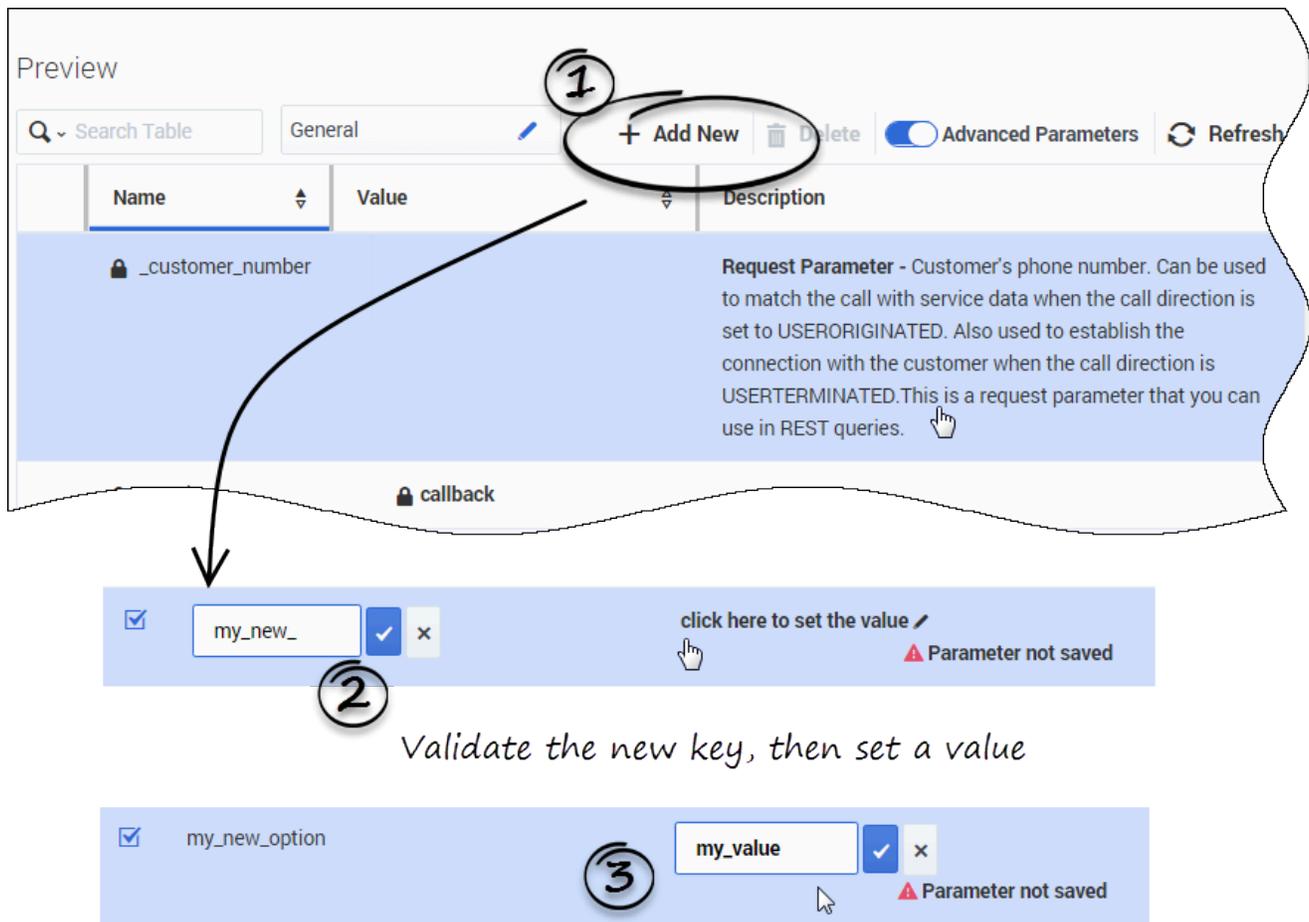
In the service panel, click **Add New** to add a new parameter, fill in the form, and save the parameter. A popup message displays the operation result.

> The parameter's name must be a valid ECMAScript variable name. This means that variable semantics that include elements like "." (for example, foo.foo) and "-" (for example, foo-foo) are not allowed.

Search for Services and Options
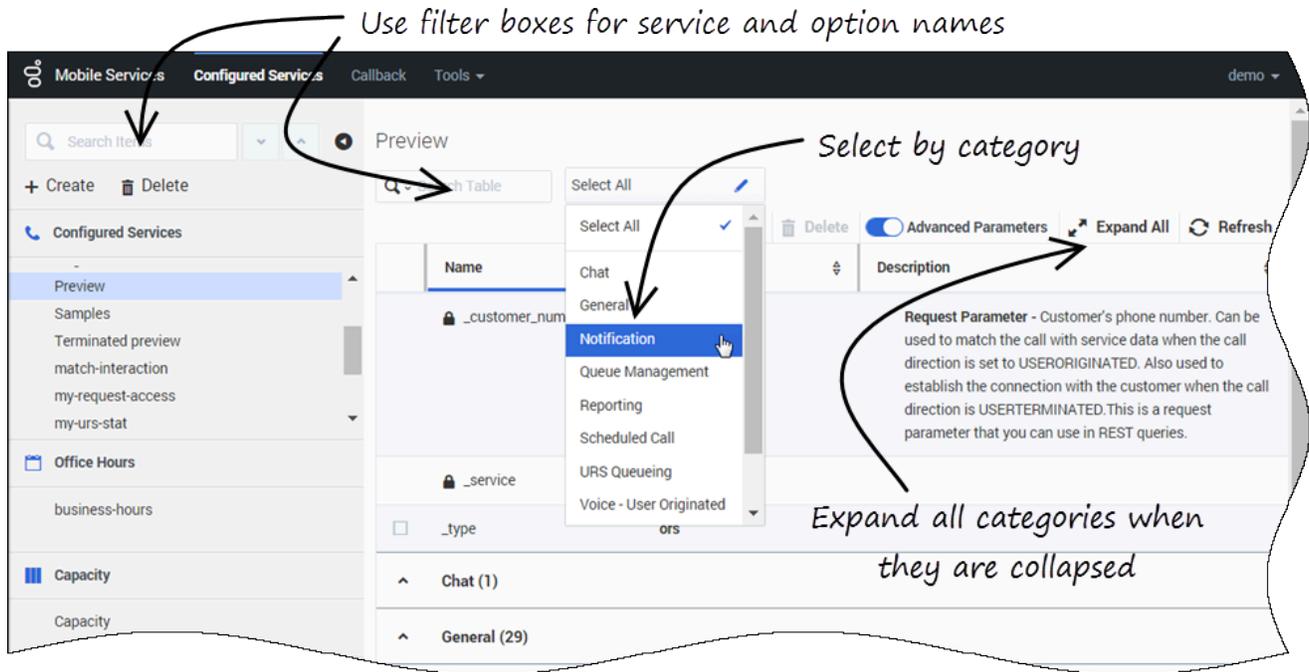
The services can be filtered in the 🔍 **Search Items** box. You can also filter the parameters displayed for the selected service or display them by category.

Callback Service

> **Important**
>
> The Callback tab provides the Callback UI. You cannot use it to customize a Callback Service. To create and configure your Callback Service, use the **Services** tab.

After Callback is configured, you must create a Callback Service for each Callback scenario that you want to implement. Then, you can manage Callback interactions in the Callback UI.

- To create a Callback service, you need Administrator permissions.

- You must set up Office Hours if you want to implement **Scheduled Callback**.

Load the Callback Service Template

Open the Service Management UI, and navigate to **Services and Tools > Tools > Service Templates**. By default, **callback** should be part of the **Service Templates** list.

If not, you must load the Callback service template before you can create a Callback service. The `callback.zip` template is located in the *<GMS installation directory>/service_templates* directory.

Once the callback template has been loaded, it is available in the filter drop-down list of the **Services** tab.

Create a Callback Service

Navigate to the **Services and Tools > Services** tab. There, you can manage your services based on the templates. The services and categories can be collapsed or expanded for easier viewing.

1. Click **Create**.

2. Enter a **Service Name**: This name will be used as the `{callback-execution-name}` parameter in your Callback queries.

3. Select your **Common Default Configuration**. These configuration scenarios are detailed in the Callback Scenarios section of this guide.
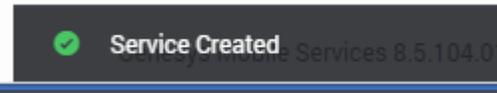
4. Click **Add**. An information message confirms the service creation.

The new Callback service appears in the *Configured Services* list.

> ## Important
> The service is also created in the `service.{callback-execution-name}` section of your GMS configuration.

Configured Callback Services

Display Advanced Parameters

Callback Delayed

Search Table | Select All

+ Add New | Delete | ○ Advanced Parameters | ↗ Expand All | ○ Refresh

Name	Value	Description
☐ _max_ors_submit_attempts	3	Maximum number of times request for execution will be submitted to ORS. After it is reached request is removed from persistent queue and discarded.
🔒 _max_time_to_wait_for_agent_on_the_call	3600	This is the maximum amount of time (seconds) to wait for the agent to accept and answer the call after customer is connected
🔒 _max_transfer_to_agent_attempts		This is the max number of times to retry transfer of the call to the agent
🔒 _media_type	voice	Media type of the interaction the service is expected to handle. This is required for URS to select agent with proper media capabilities.
☐ _orig_service_id		Request Parameter - Id of the service that originated callback service request. Example: IVR service, web session service, etc
🔒 _snooze_duration	300	After agent availability notification is received, user can snooze the notification for the specified duration.
🔒		

Optional

Mandatory

Request parameter for API queries

Configured Services
- Callback Delayed
- match-interaction
- request-access
- request-interaction

Office Hours
- Business-hours

Search Items

+ Create | 🗑 Delete

Key parameters for the service are automatically populated with the appropriate default values. For cluster configurations, all changes made in a service will be replicated into the entire GMS cluster.

Important

To configure your Callback service, refer to the configuration options described in the associated scenario page. See the Callback scenarios for further details.

The parameters have the following characteristics:

- Mandatory parameters - are identified with a lock icon; you cannot rename them or remove them.

- Optional parameters - you can rename them by hovering your cursor over the value field (you will see a pencil icon), click, and then enter the new value. You can delete optional parameters by clicking the *Delete* button.

- Advanced parameters - can be displayed by selecting the **Advanced** button in the upper right.

- Request parameters - are identified with an ***Request Parameter*** label in the Description.

Important

Request parameters are usually provided within the request itself. Genesys recommends that you do not configure the Request parameters through this UI, otherwise, the parameter within the request will be overridden. If a value was entered through this UI and you wish to remove it at some later date, you can click the garbage can icon and the value becomes *not specified.*

Adding your Service to Virtual Service Groups

Introduced in 8.5.207

To add services to a given virtual group, add the **_service_groups** option to your service and enter a comma-separated list of group names.

For example, let's consider adding the 'support' service group by adding **_service_groups** = 'support' to the `callback-support` and `callback-blackfriday-support` services.

Then, you can use this virtual group to filter the services displayed to an agent. See Setting Permissions for further details.

Accessing Callback Services and Records with the Callback API

The URLs used by the Callback API are dependent on the name of the Callback service that you have just created. Callback services are available at the following URL:

`http://<host>:<port>/{base-web-application}/service/callback/{callback-execution-name}`

For instance, if you create a callback service named `callback-for-mobile`, then {callback-execution-name} is `callback-for-mobile` and the callback service is available at:

`http://<host>:<port>/{base-web-application}/service/callback/callback-for-mobile`

For further details, see Callback API.

Capacity Service
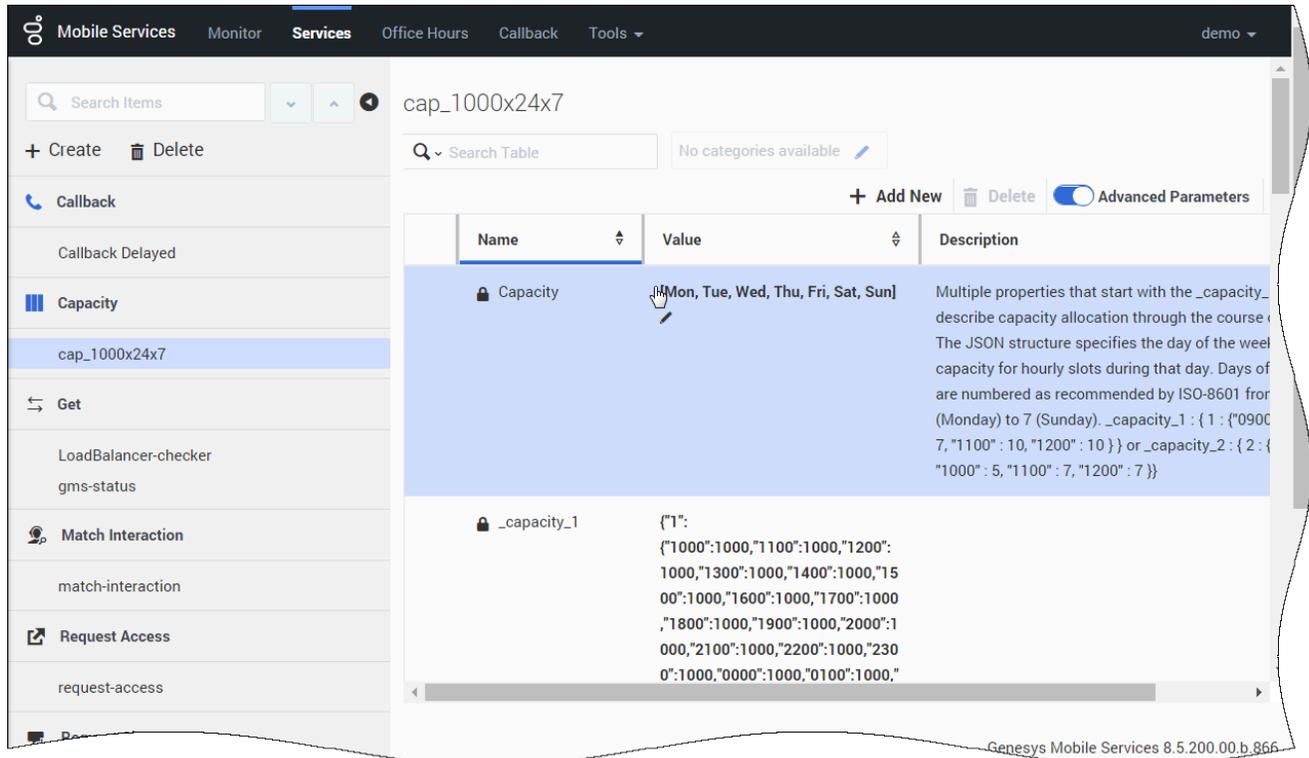
The Capacity Service enables you to define the number of scheduled callbacks that are allowed for Callback for a given time slot in the week. Then, your Callback service refers to your Capacity service and to your Office Hours service to adjust the agent availability and the number of scheduled callbacks.

You can define exceptions for dates when fewer or more scheduled callbacks are available, and you can define as many Capacity services that you need to match your Callback services.

Make sure to update the existing calendar configuration to set the correct timezone for your Capacity service. For instance, if you configured "EST", or "PST" timezones with the configuration, your parameters must use the timezones defined for Java such as "America/Toronto", or "Europe/Paris". See Wikipedia to get the list of correct timezones.

Create a Capacity Service

Create a new service and select **capacity** in the list of templates.

Once this service is created, you can use the Capacity API.

> **Important**
>
> You also need to create an Office Hours service.

Link your Capacity to your Callback Service
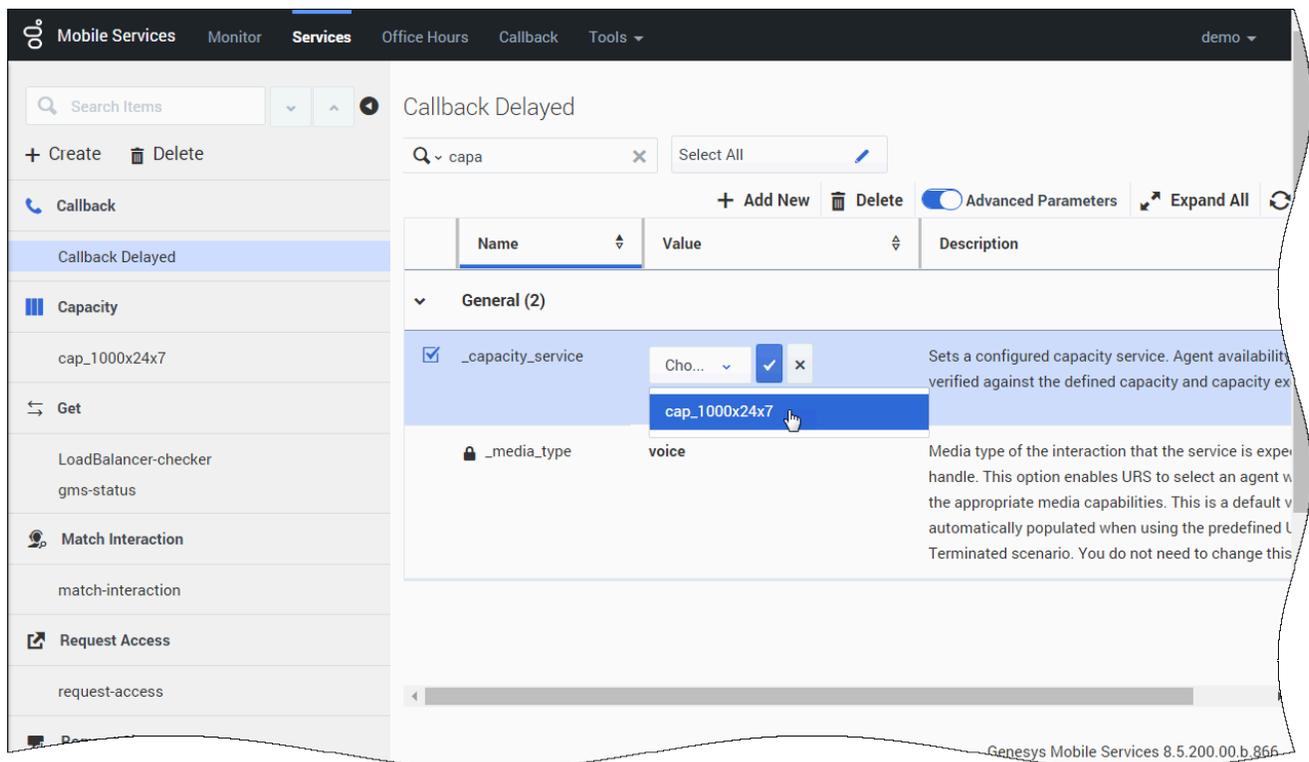
If your Callback service needs variable capacity levels, you must map its _capacity_service parameter value with the name of the Capacity service that you have created.

> **Important**
>
> Callback services that need fixed capacity levels can continue to use the _max_request_by_time_bucket option. But, if your Callback service includes both _capacity_service and _max_request_by_time_bucket options, then _max_request_by_time_bucket is ignored.

Add Capacity

Capacity

Q ⌄ Search Table   No categories available ✎

+ Add New   🗑 Delete   ⬤ Advanced Parameters   ↻ Ref

	Name ⬍	Value ⬍	Description
	🔒 Capacity	[] ✎	2 Dimentional Matrix specifying the number of agents available per time slot.
☐	Capacity Exception	{}	1 Dimentional Matrix for specific days. Overrides _capacity.
	🔒 _service		
	🔒 Service Timezone		
☐	_type		

Capacity   ✕

Timestamp	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Set Slot
00:00	0	0	0	0	0	0	0	___
01:00	0	0	0	0	0	0	0	___
02:00	0	0	0	0	0	0	0	___
03:00	0	0	0	0	0	0	0	___
04:00	0	0	0	0	0	0	0	___
05:00	0	0	0	0	0	0	0	___
06:00	0	0	0	0	0	0	0	___
07:00	0	0	0	0	0	0	0	___
08:00	0	0	0	0	0	0	0	___
09:00	5	10	10	10	0	0	0	___
10:00	5	10	10	10	0	0	0	___
11:00	5	10	10	|	0	0	0	___
12:00	5	10	10	0	0	0	0	___
13:00	5	10	10	0	0	0	0	___
14:00	5	10	10	0	0	0	0	___
15:00	5	10	10	0	0	0	0	___
16:00	5	10	10	0	0	0	0	___
17:00	5	10	10	0	0	0	0	___
18:00	5	10	10	0	0	0	0	___
19:00	0	0	0	0	0	0	0	___
20:00	0	0	0	0	0	0	0	___
21:00	0	0	0	0	0	0	0	___
22:00	0	0	0	0	0	0	0	___
23:00	0	0	0	0	0	0	0	___
Set Day	___	___	___	___	___	___	___	

Cancel   Reset   Save

Edit the **Capacity** value to open the Capacity grid widget. Enter your capacity per Day or Hours, then **Save**.

> ### Important
> Values above 999 are accepted but may not display properly in the UI.

You can see for which days of the week the Capacity service is defined.

Add Exceptions

For more flexibility, you can set exceptions for the Capacity service. You can enter dates with different capacities for federal holidays, vacations, and so on.

Edit the **Capacity Exception** value to open the interface.

You can add as many exceptions as you need. You can also even update later to modify your capacity.

Click **Save** to update your Capacity data. A popup message displays the operation result.

Display Sub-capacities and Sub-exceptions

To display the list of sub-capacities and sub-exceptions, enable **Advanced Parameters**.

Office Hours Tab

> **Important**
>
> - You must set up Office Hours if you want to implement **Scheduled Callback** and **Disposition Dialog**.
>
> - Office Hours are used when requesting Callback API to create an immediate or a scheduled callback. See Start or Schedule a Callback for more information.

Upload the Office Hours Template

In the **Services and Tools > Tools > Service Templates** tab, make sure that the **Office Hours** template is available. Otherwise, click **Upload** to add the office-hours.zip to the **Templates** list.

Create the Office Hours Service
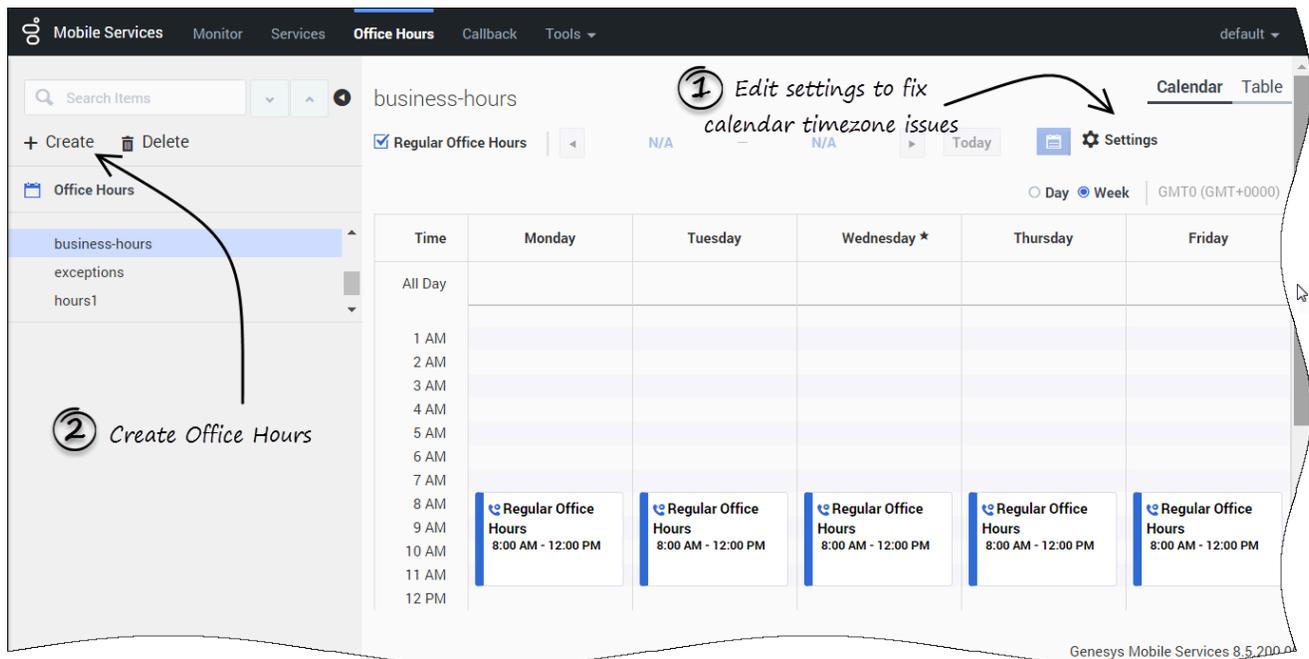
Click **Create** in the **Office Hours** tab, select office-hours in the filter and configure the business-

hours service.

Set your Calendar Timezone
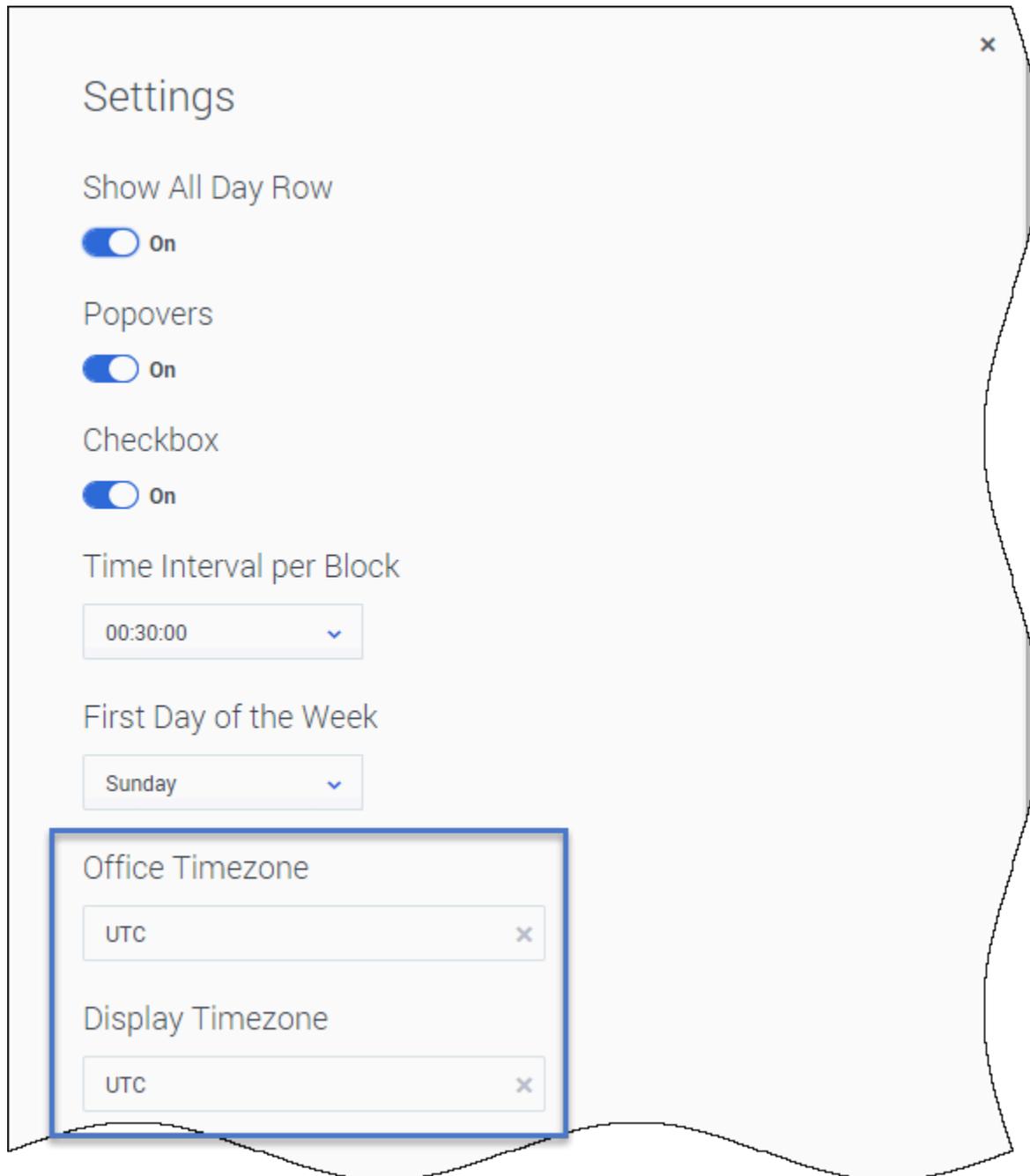
The **Display Timezone** is the timezone of your **Calendar** view and is set to the Browser timezone by default. The **Office Timezone** is bound to the `_timezone` parameter of the Office Hours service.

The **Calendar** view is in Read-Only mode if your Display Timezone and Office Timezone are different. In that case, the interface shows a warning message and does not allow you to edit the **Calendar** view.

[+] Show me the message

> ℹ The calendar is in read-only mode because the office timezone is not equal to the displayed timezone

To fix this issue, click **Settings** and fix your **Office Timezone** and **Display Timezone** by setting identical timezones.

Add Office Hours to your Callback Service

Select your Callback Service and expand the **General** category. Find the `_business_hours_service` parameter and select your Office Hours instance in the drop down list.

Create Regular Office Hours
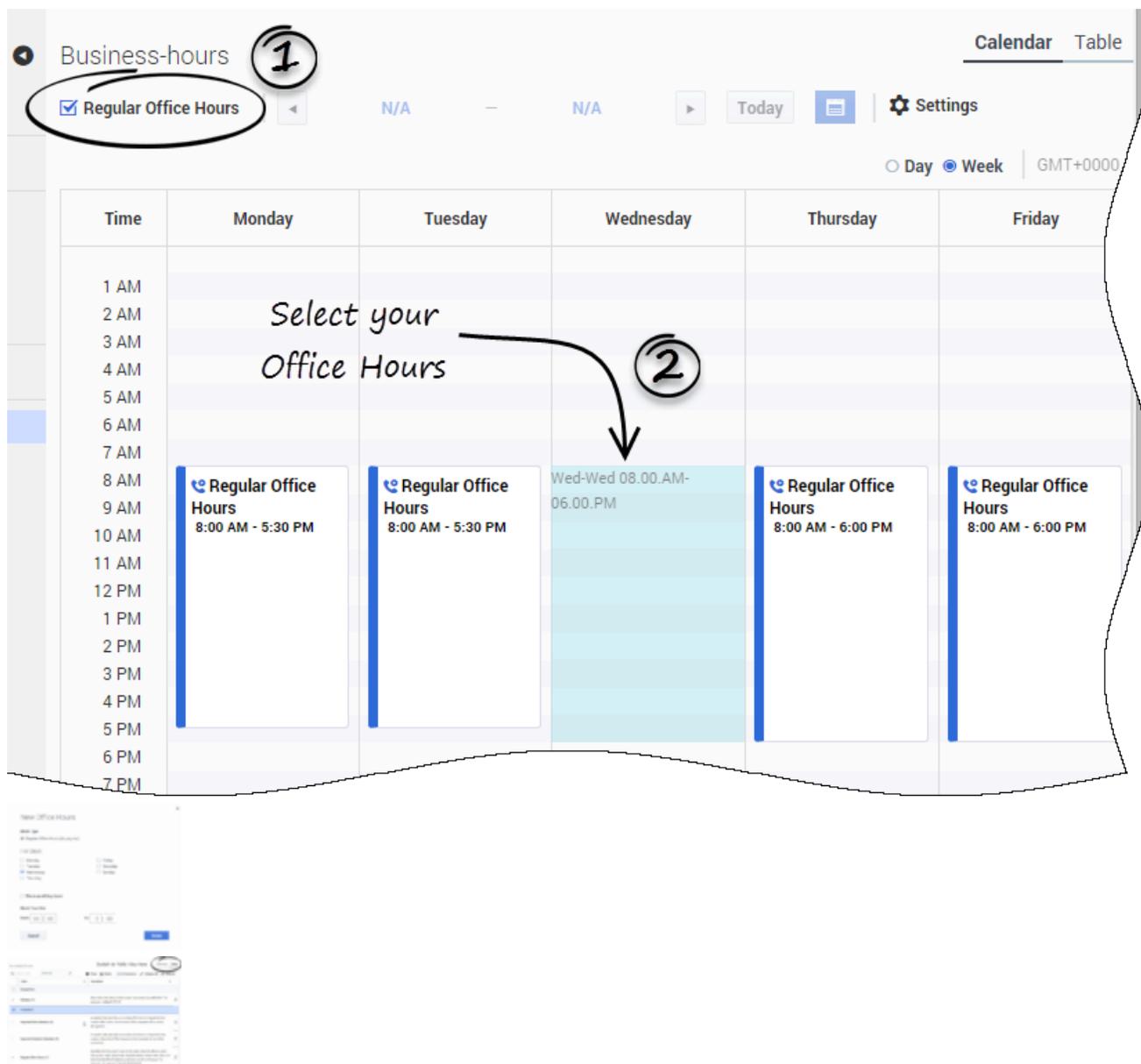
Select your Office Hours service in the **Configured Service** tab.

In the **Calendar** view, make sure that **Regular Office Hours** is checked.

Select your Office Hours service in **Configured Services**. You can add Office Hours by selecting a timezone in your **Calendar** view; the Office Hours interface shows up. Or, you can switch to the **Table** view and edit options there.

Switch Views to Check Office Hours

If you add Office Hours to one or more days in the week, the results apply to all months and weeks. Switch to the **Month** view to see the actual Office Hours of the month.

Add Holidays and Exceptions
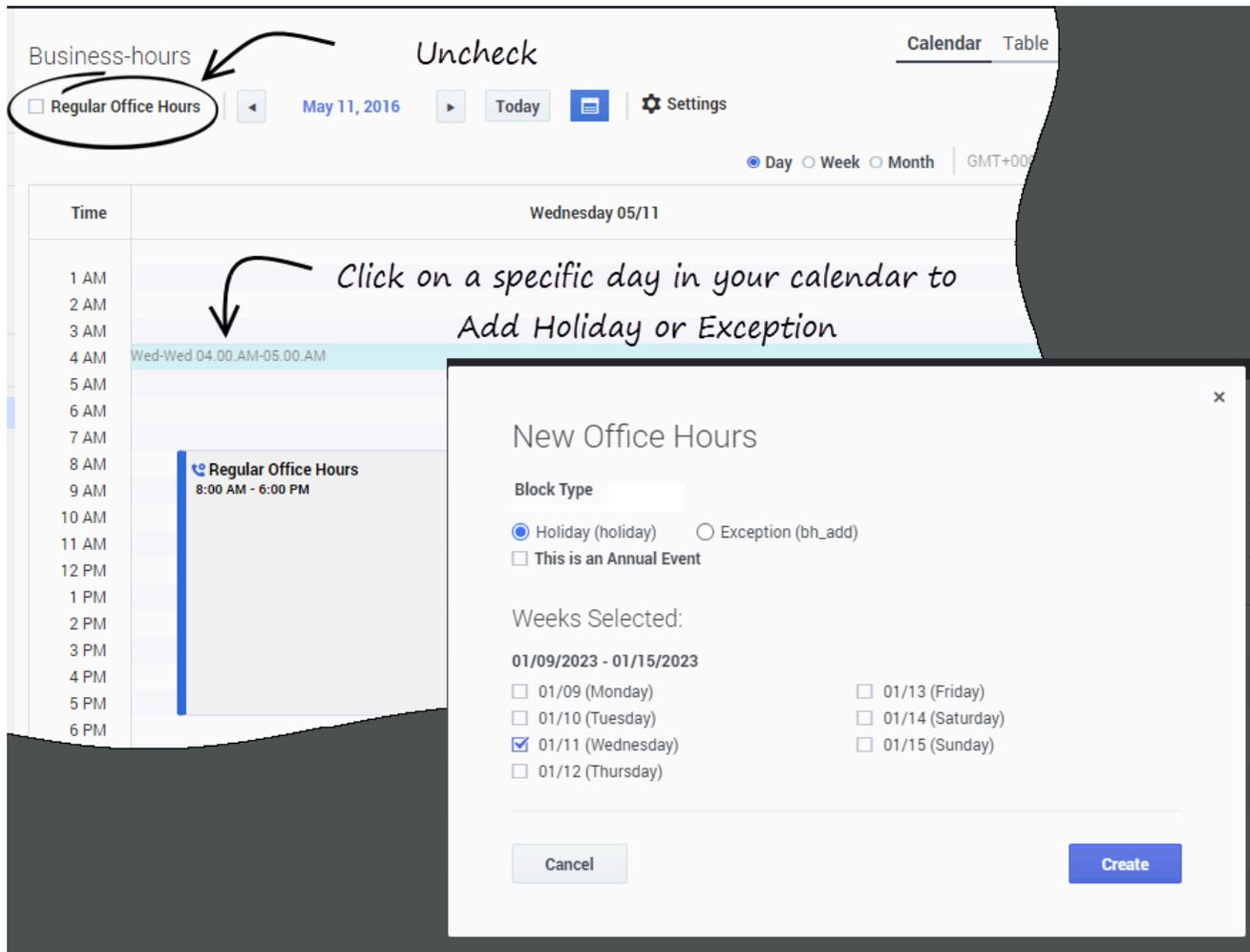
You can set certain dates as Holidays so they will not be used for Callback and certain dates as Exceptions (or Special Office Hours) for days that require additional hours.

To add these special events, uncheck the **Regular Office Hours** parameter, then click in the **Calendar** view on the appropriate date to open the dialog box. By default, the special event is added to the current year.

When you create the holiday or the exception, if you select the **This is an Annual Event** option, the holiday or exception is planned annually, not only for the current year.

Exceptions Always Apply

If you set a Holiday and an Exception on the same day, then the Exception applies. The office is opened during the hours configured in the Exception.

If Regular Hours, Holiday, and Exception are set on the same day, then Holiday blocks Regular Hours, but the Exception still applies and the office is opened during the Exception's hours.

Add Holidays and Exceptions from the Table view

You can also add Holidays and Exceptions by editing the associated options in the **Table** view. If you do not check the Annual Event option, the holiday or exception is added to the current year.

Add Holidays and Exceptions from the Table view

You can also add Holidays and Exceptions by editing the associated options in the **Table** view.

Import Rules in Office Hours

The **Import Rules** feature enables you to set up Regular Hours, Holidays, and Exceptions (Special Office Hours) that you want to apply to several Office Hours services. The imported content (options and setup) becomes part of the Office Hours service.

- Further changes to the imported services (Regular Hours, Holidays, Exceptions) instantly apply to all the Office Hours services that imported the rules.

- The **Import Rules** feature ignores the timezones defined in the imported service.

Use Case: Import Holiday Rules

Let's consider that you have several offices in the United States, sharing the same federal holidays and vacations. In that scenario, you can create a dedicated Office Hours called `Federal USA Holidays`, and then import it in your U.S. offices, instead of manually adding the same holidays to your **Calendar** views for each service (`Office New York` and `Office SFO` in the example below).

Use Case: Import Exception Rules

Let's consider that, every two weeks during spring, all of your offices are opened on Wednesday evenings from 6-9 p.m. for a special event called *Special Deal*. In that scenario, you can create an Office Hours service called `Special Deal Events` where you create the needed extra hours. Then, you can import this rule in each concerned office (`Office New York`, `Office SFO`, and `Office Amsterdam` in the example below).

Note that `Special Deal Events` will apply even if Regular Hours and Holiday are set for a given date, as explained above.

Set up Rules in a Dedicated Office Hours Service
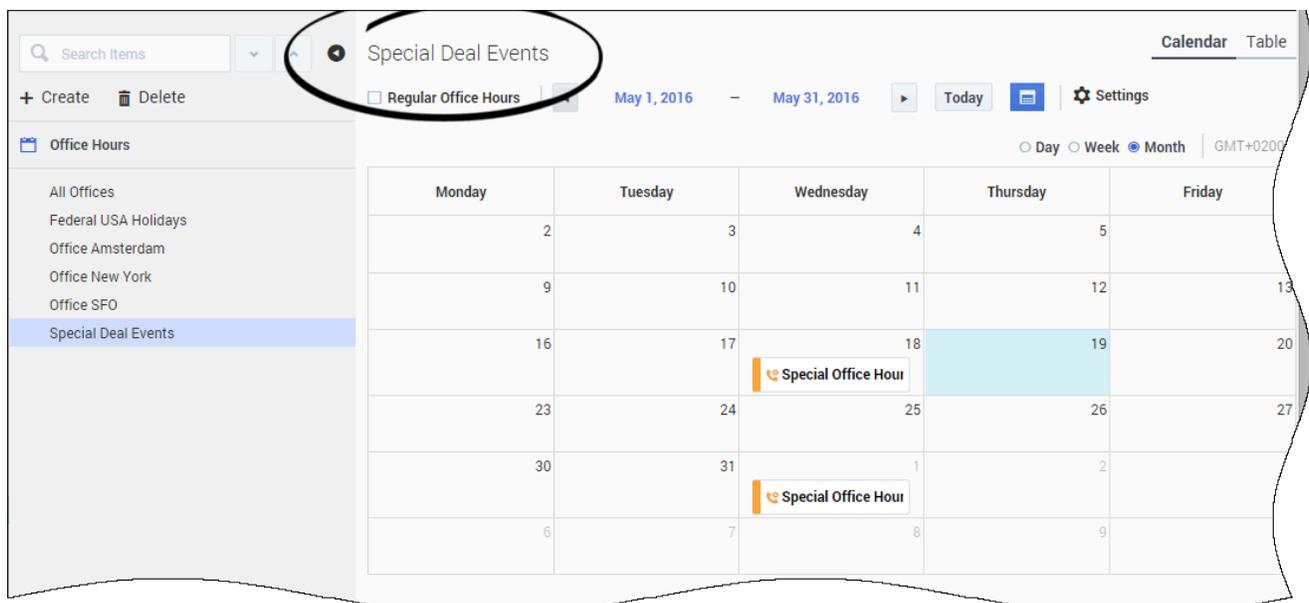
To set up these rules, create a new Office Hours service. In this example, two services are created: `Special Deal Events` and `Federal USA Holidays`.

Import Rules in Office Hours Service

Switch to **Table** view and expand **Imported Rule Calendars**. Click to add a new rule line, then select the appropriate Office Hours service.

View Imported Rules
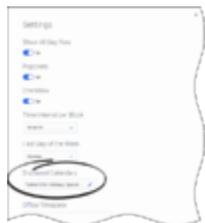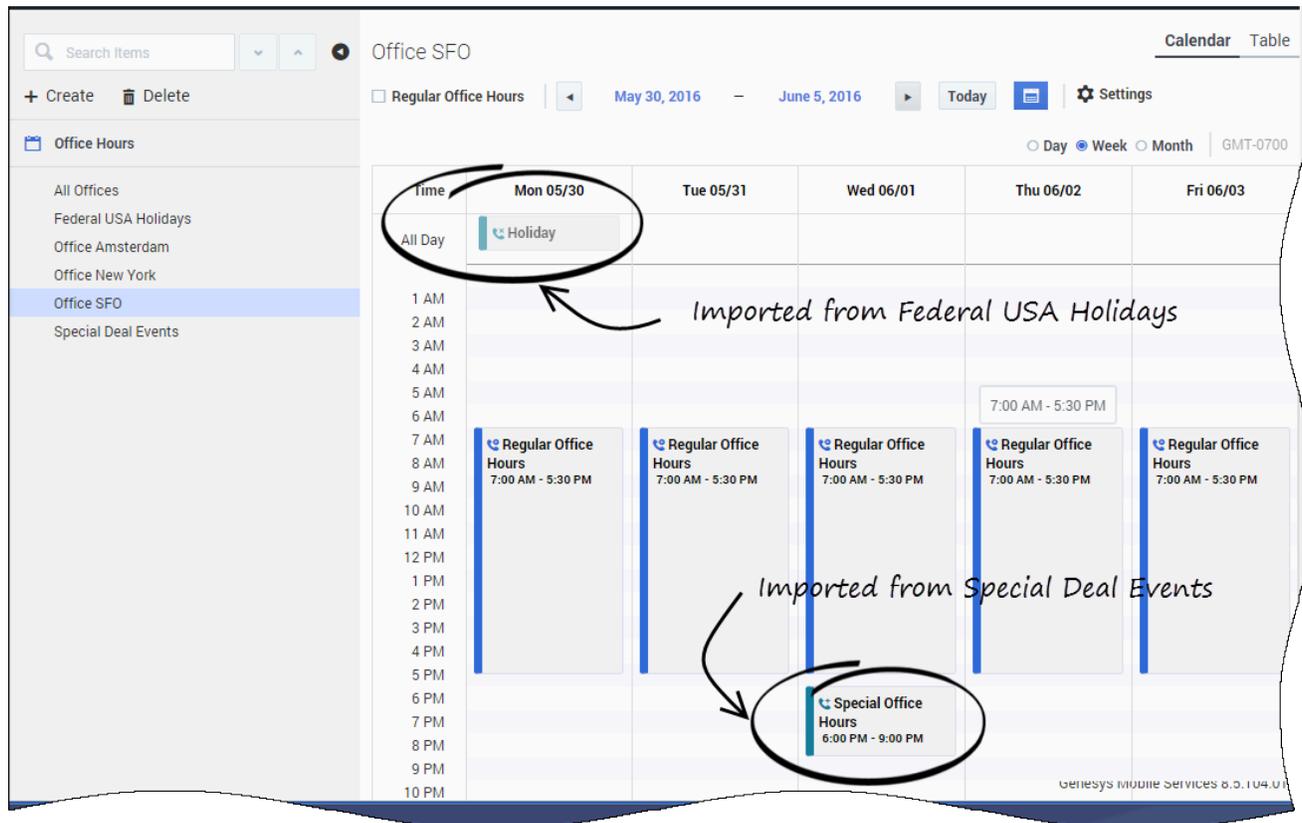
Switch to the **Calendar** view to display the results of the imports. In this example, you can see that the `Special Deal Events` and `Federal USA Holidays` are imported in the `Office SFO` service.

If you update an Office Hour service (for instance, `Special Deal Events` or `Federal USA Holidays`), you will see these updates replicated to the Office Hours services importing the modified calendars (here, `Office SFO`).

You can also click **Settings** and see the imported Offices Hours in the list of **Displayed Calendars**.

Import Schedules in Office Hours

You can import **Schedules** from other Office Hours Services to combine working hours of multiple teams and shifts in the same Office Hours service. In this scenario, you can create a main Office Hours service (`All Offices` in the example below) into which you can import all the Office Hours schedules (Regular Hours, Holidays, Exceptions, and timezones). This results in the union of the imported Office Hours schedules within the timezone of the main Office Hours.

> ### Tip
> The **Import Schedules** feature helps you to manage teams dispatched in different timezones.

Import Schedules in Office Hours Service
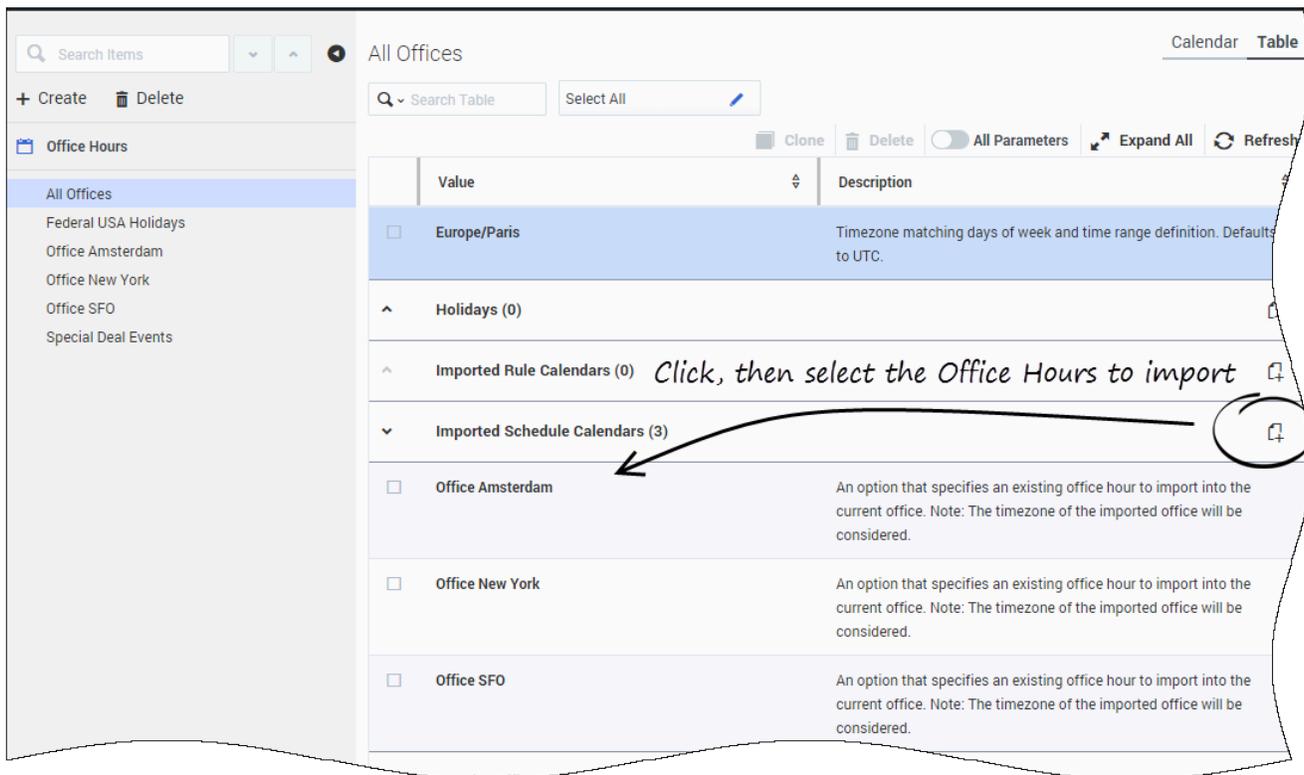
Switch to **Table** view and expand **Imported Schedule Calendars**. Click to add a new rule line, then select the appropriate Office Hours service.

Display Imported Schedules
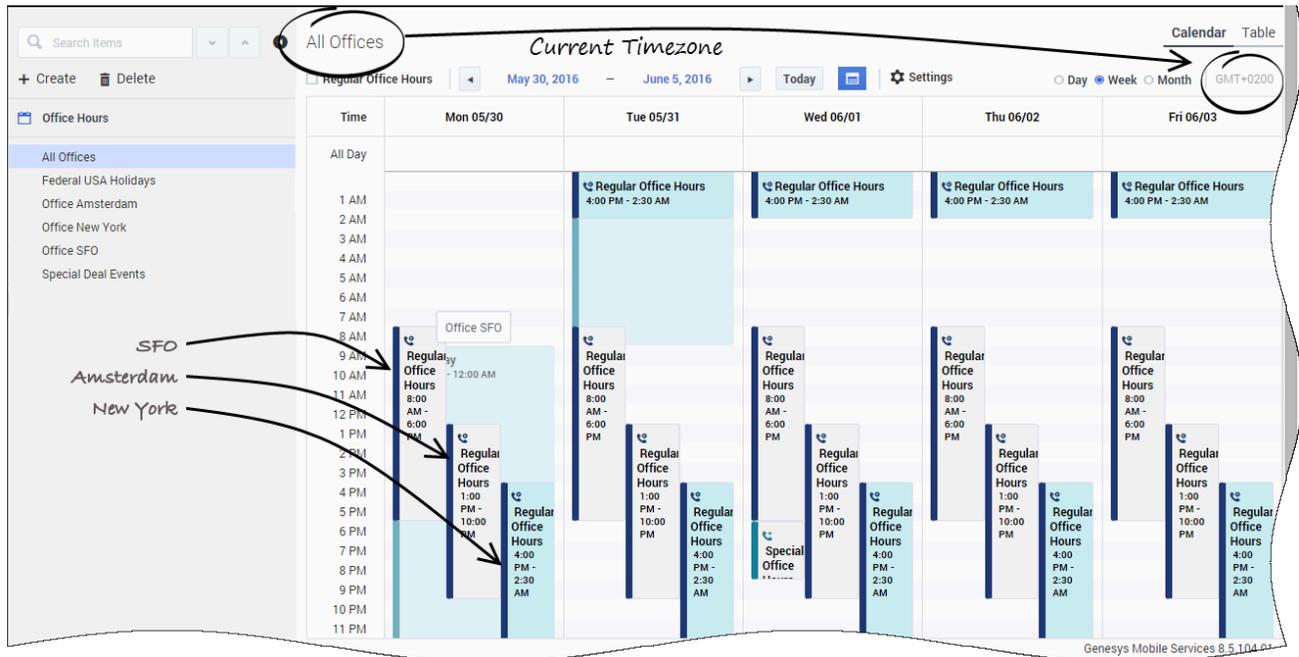
Switch to the **Calendar** view to see the results of the imports.

In this example, the `All Offices` service combines the schedules of the `Office SFO`, `Office New York`, and `Office Amsterdam` services.

The timezones, Regular Hours, Holidays, and Exceptions are aggregated in the **Calendar** view that shows the callback coverage for the three offices.

You can also click **Settings** and see the imported Offices Hours in the list of **Displayed Calendars**.

Office Hours Options Reference

For a list of Office Hours options, see the Office Hours section of the Genesys Mobile Engagement Options Reference Guide.

Tools tab
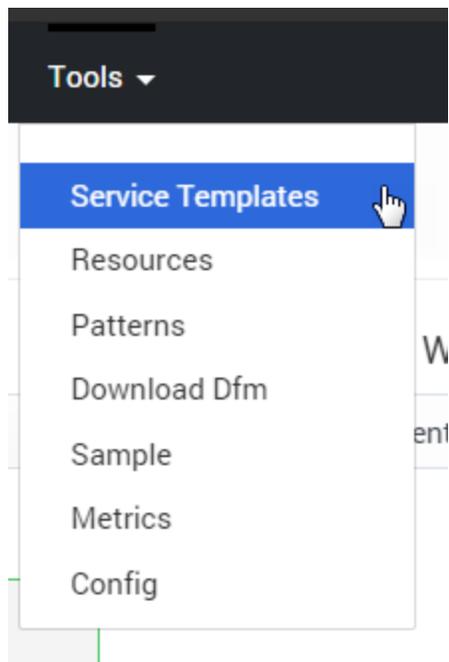
The content of the **Services and Tools > Tools** menu depends on the options configured in the features section.

Service Templates

The **Tools > Service Templates** panel illustrates how to import service templates. See Service Templates for details.

Resources

To enable resources in the Service Management UI, configure enable-resources = true in your GMS configuration.
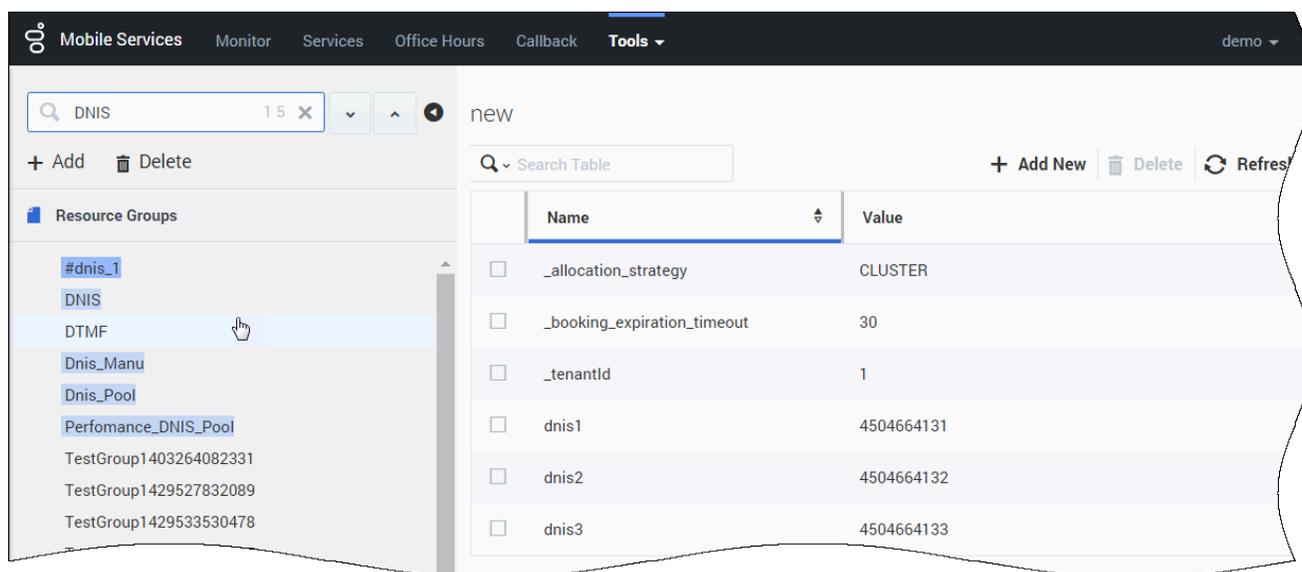
The *Resources* tab enables you to create new resource groups and add resources. It provides the same functionality as *Resources* (Configuration Database Objects) in Configuration Manager, as well as displaying available and total resources.

Patterns

To enable patterns in the Service Management UI, configure enable-patterns = true in your GMS configuration.
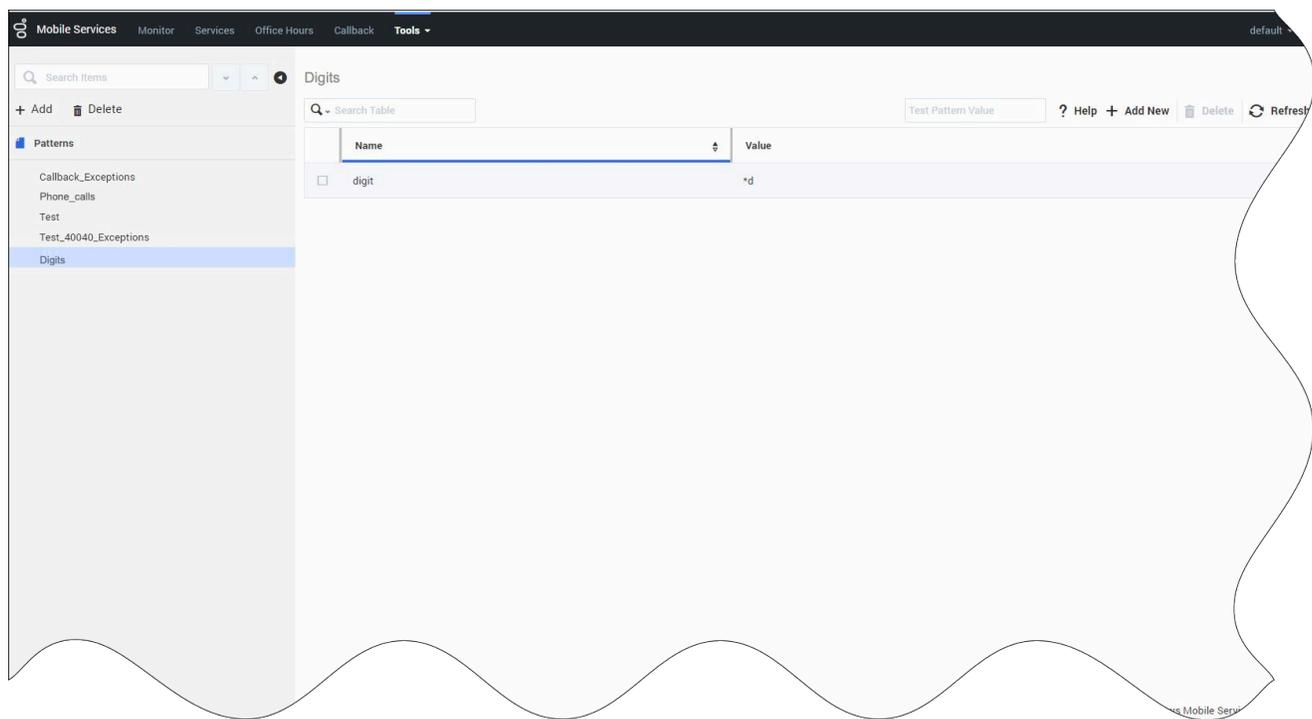
The *Patterns* tab enables you to configure groups of exceptions; for example, phone numbers. The *Help* button displays the format for the expressions. Once you have defined some patterns, use the Pattern Matcher API queries to check the validity of your parameters. In the patterns group, you can test the value of a pattern against an entire group. Type a value in the input field, and if a match is found, the corresponding table row will be highlighted.

Download DFMs

To enable DFMs in the Service Management UI, configure enable-downloaddfm = `true` in your GMS configuration.
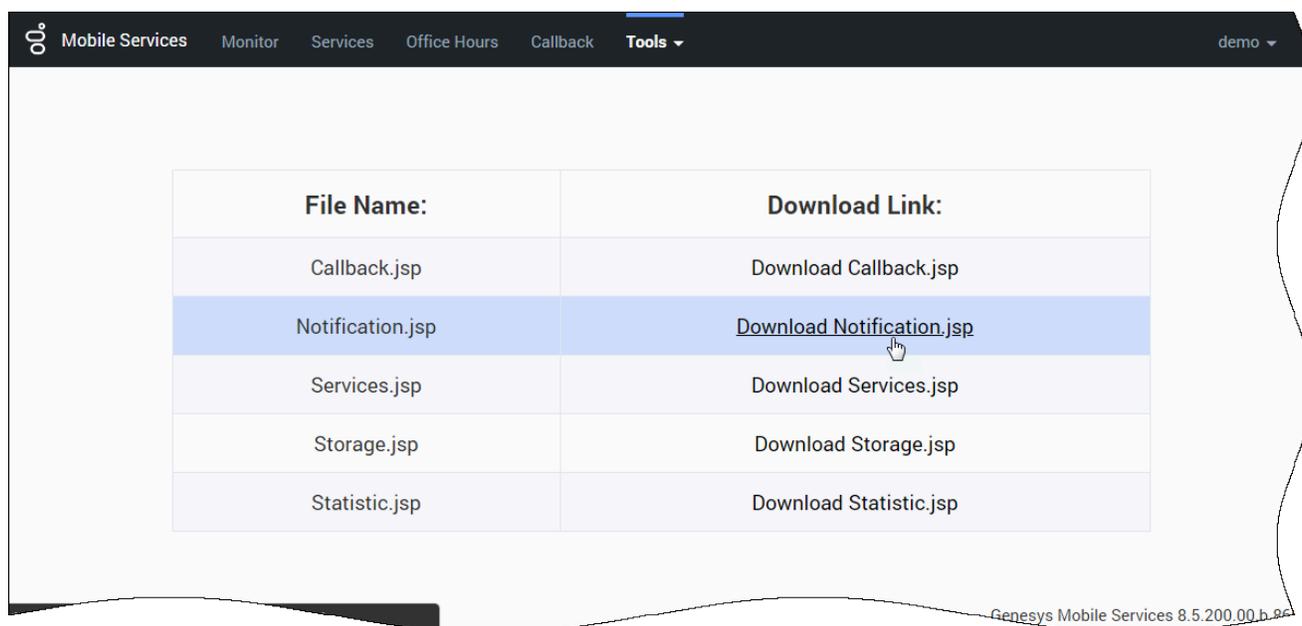
The **Tools > Download Dfm** panel enables you to download special configuration files, called DFM.

- DFM files define Genesys Mobile Services-specific SCXML that are required for Orchestration Server-based services.

- You must deploy these files in your Orchestration Server application if you need to execute SCXML applications used within Orchestration Server-based Services.

Click one of the DFM file links and the download will start.

Important

You must update the DFM files deployed locally with the latest version provided in the GMS Service Management UI.

Sample

The **Tools > Sample** panel illustrates how to implement a Desktop/Mobile browser web application that communicates with GMS and performs supported contact scenarios. It is primarily meant to be used by developers as a reference to build a Javascript-based web application with GMS. See Sample for details.

Metrics

To enable Metrics in the Service Management UI (known as Reporting in the former UI), configure
enable-metrics = `true` in your GMS configuration.

The **Tools > Metrics** panel is a dashboard that displays current resource and service utilization. You
can select pre-defined date and time ranges, and export the reports into CSV, PDF, or Excel file
formats. You also have the ability to zoom into a selected area of the graph to view additional details.

Four types of reports are available:

- Resources - Displays the usage history of the selected Resource group. **Note:** These are the Resource
  groups that were defined in the ***Tools*** menu.
- Services - Displays the aggregated total a service was accessed for a given time period.
- Callback - Displays Callback services data, if the Callback services are active.
  - Display Advanced Metrics - When this option is selected, you can choose between a metric type of
    `Holdtime` or `Watermark`.
    - Holdtime - You can set parameters for displaying the HoldTime statistics (media type, rollup-
      range, and time range).
    - Watermark - For Callbacks in the QUEUED state, you can display high, low, and average
      watermarks (AVG, MIN, MAX) over time periods of 15 minutes, 30 minutes, an hour, or a day.
      Note that the Watermark is computed every minute, so for example, running 20 Callbacks
      simultaneously within a one-minute period, then none for the rest of the same 15-minute period,
      will result in Min=0, Max=20, and Avg=1.
- Operational metrics - DNIS pool usage and availability.
  - DNA availability by pool – Determines, in real time, the number of outstanding service requests (or
    available resources) by resource pool.

- Average Handle Time (AHT) by resource pool – Reports historical AHT, in seconds, by resource pool summarized by 15 minutes, 30 minutes, an hour, a day, or over a range of periods such as today, yesterday, last week, last month, or a range of dates. Timeouts do not impact the calculation.

- Service Exiting Reporting — Reports on historical service terminations (match or timeout) by resource pool summarized by 15 minutes, 30 minutes, an hour, a day, or over a range of periods such as today, yesterday, last week, last month, or a range of dates.

Config: Jetty XML Snippet

To enable **Config** in the Service Management UI, configure enable-config = `true`
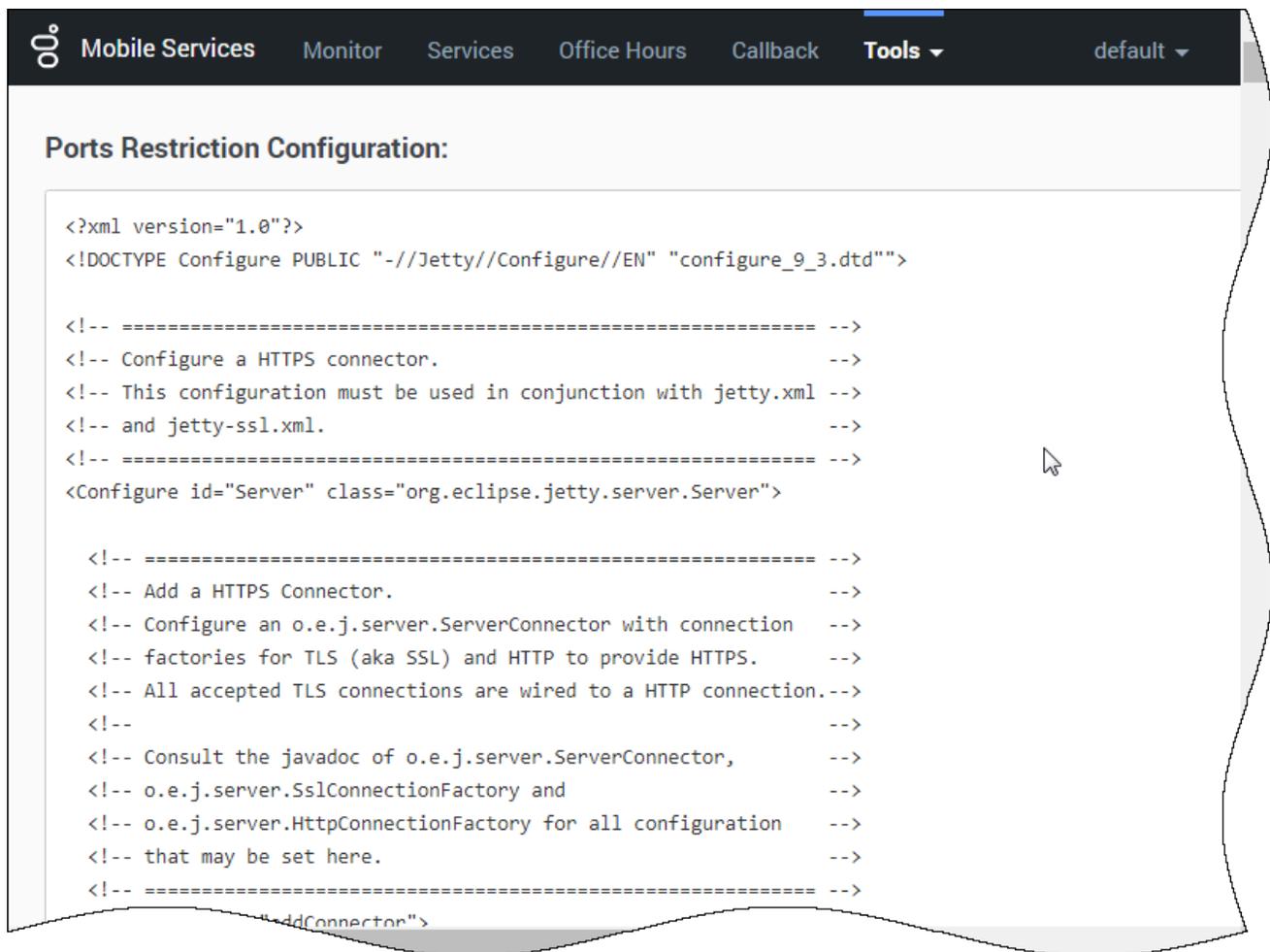
The *Config* tab provides you with a code snippet that you must add to the `jetty-http.xml` file for port number control. The code snippet is available when port restrictions have been enabled in your GMS Configuration. See Restricting Ports for more information about using this feature.

By default, restrictions are disabled and the **Config** Tab displays a default message.

[+] See default message
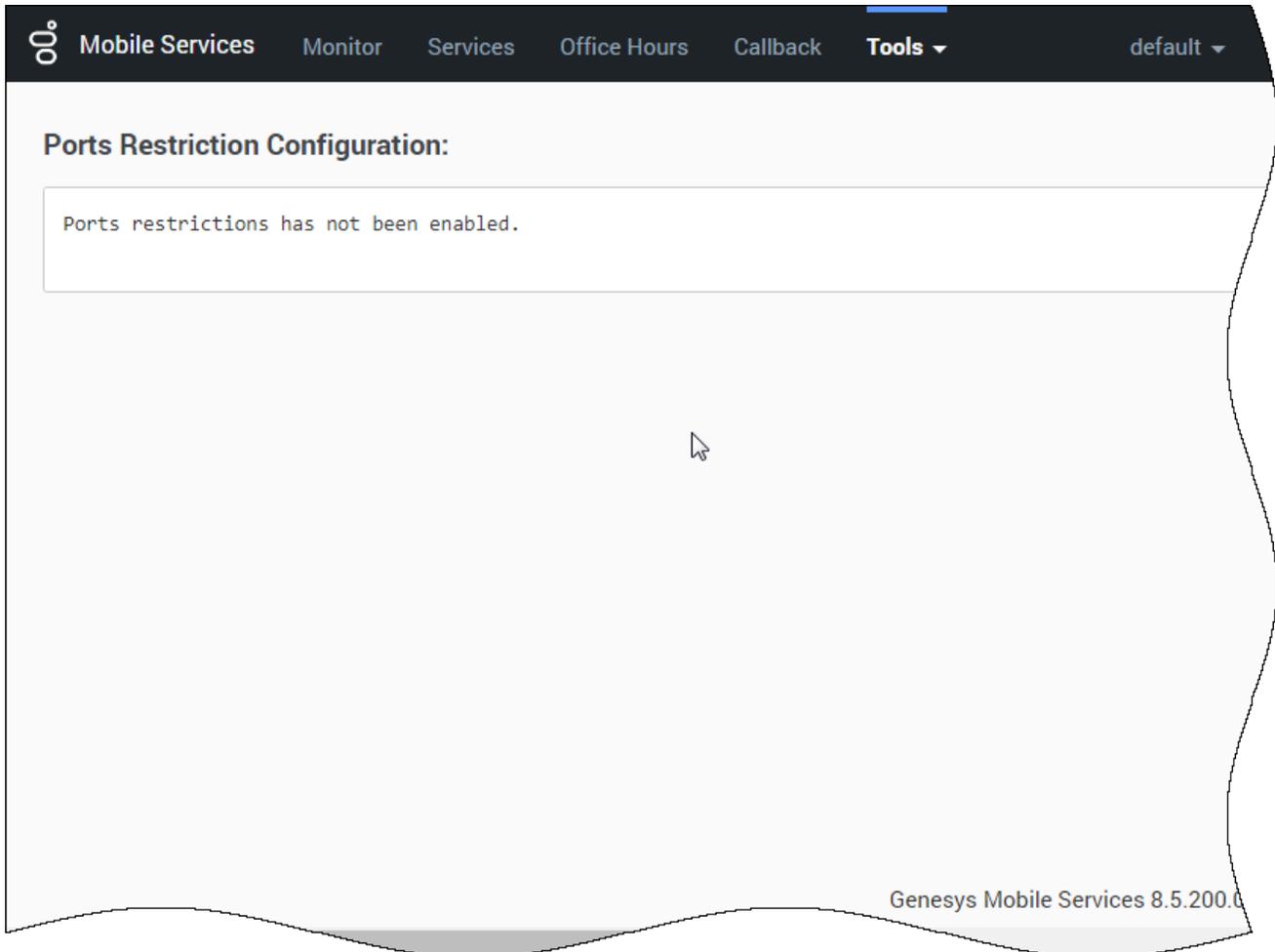
Service Templates

Upload Service Templates
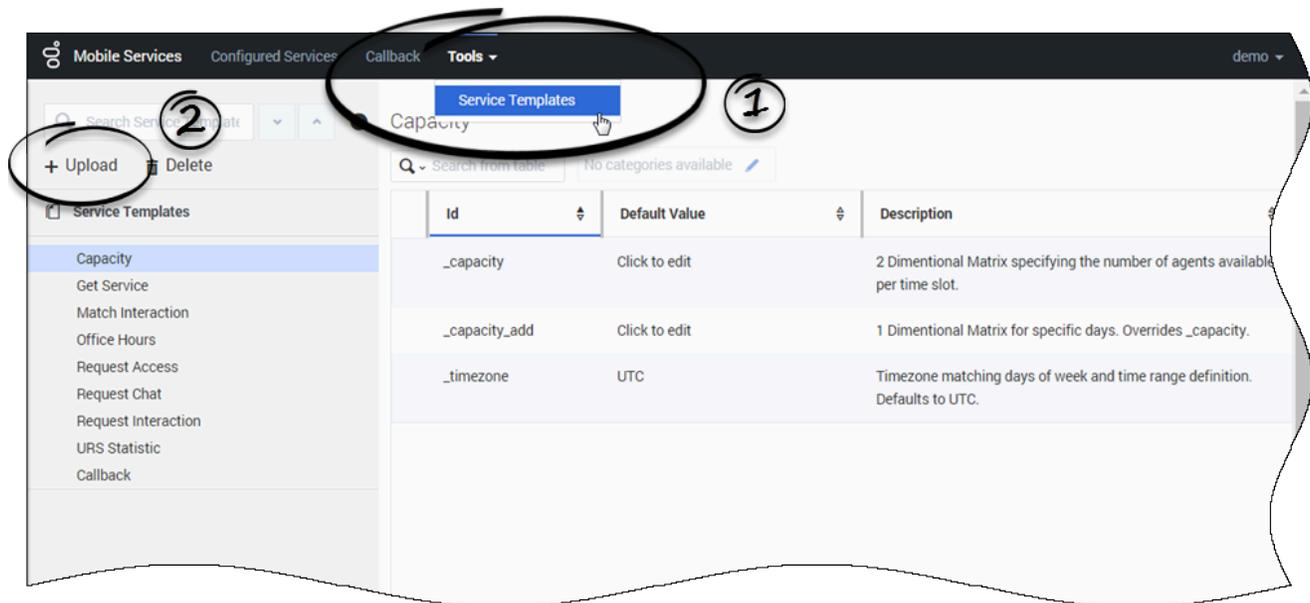
Before you can start creating services, you must load the template associated with these services.

- The template lists options related to your services.
- Each template includes one or more scenarios detailed in this Help, except for the Callback template scenarios, which are detailed in the Callback Solution Guide.

To add a template, select **Service Templates** in the **Tools** menu bar and click **Upload** to browse your template. The template is added to the list.

Manage Service Templates
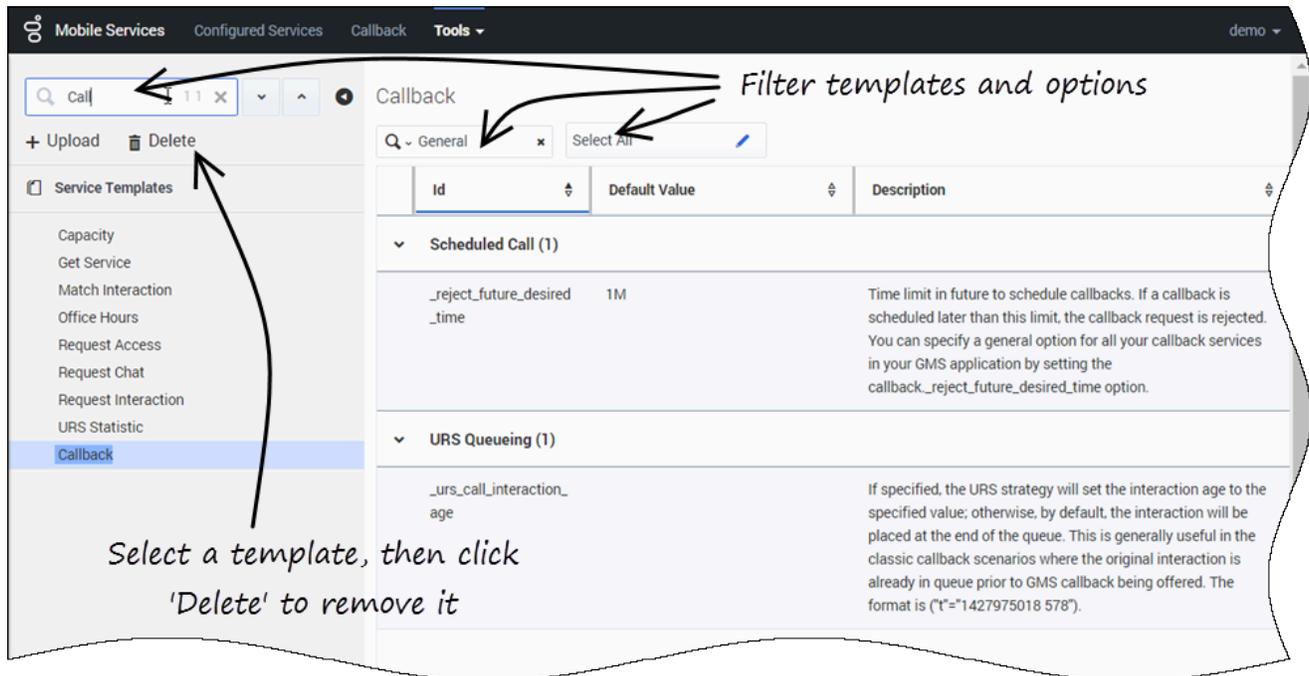
The Service Templates tab enables you to upload and delete templates. Added features include:

- A Filter box to focus on specific services.
- A Search box to filter the displayed template's options based on an entered keyword.
- A Selector to display the options of a selected section.

Learn about Templates, Scenarios, and APIs

To understand the relashionship between services, scenarios, templates, and APIs, you should read the chapter about Built-in and ORS services.

Available templates, scenarios, and APIs

Template Name	Detailed Built-in	Related API(s)
Get Service (get.zip)	Get and Basic Get Services	Node API to check GMS nodes health and manage your nodes: start, suspend, stop.
Match Interaction (match-interaction.zip)	Match interaction	• Service API to check that a voice call with an existing GMS service is associated

Template Name	Detailed Built-in	Related API(s)
		with the access number. • Storage API to allow users to temporarily store arbitrary data. Data may consist of key/value pairs of strings or binary objects.
Office Hours	Office-hours	Calendar Service API to create and manage office hours, special events, and more.
Request Access	Request-access	Service API to request resources.
Request Chat		Create a chat session in the Chat Server using the Chat API v1.
Request Interaction	Simple Voice Inbound–Immediate Call	See the scenario page.
URS Statistic (urs-stat.zip)		Stat Service API to query URS Stat.
Callback (callback.zip)	User Originated Immediate	Query to create an inbound immediate service.
	User Originated Delayed	Query to create an inbound delay service.
	Chat Immediate	Chat APIs
	Chat Delayed	Chat APIs
	User Terminated Immediate	Callback Services API
	User Terminated Delayed	Callback Services API
	User Terminated Scheduled	Callback Services API.
	User Terminated Delayed Agent Preview	Callback Services API
Capacity (capacity.zip)	Capacity	Capacity API to manage Agent availability.

Callback Interface

Link to video

The **Callback** UI displays the list of Callbacks that are still alive. After the Callback due date (which is the _desired_time parameter), the service record will appear in the UI for the amount of time configured in the _ttl parameter (in seconds).

Callback records can have the following states:

- SCHEDULED - Request is handled by Callback Management service (there are no sessions started in ORS). While in this state, the request will be handled by Management when the specified `desired_time` is upcoming.
- QUEUED - Callbacks actively waiting for an agent in ORS/URS; the agent not assigned yet.
- ROUTING - Agent is reserved but the call is not yet routed to the agent.
- PROCESSING - Callback being handled by assigned agents.
- COMPLETED - Callback was completed with `_callback_reason`, for example, timed-out, cancelled, and so on.
- PAUSED - Callback was paused. See Pausing Callback for details.

Important
You must have the appropriate Supervisor role in order to use the Callback UI.

Callback UI Overview
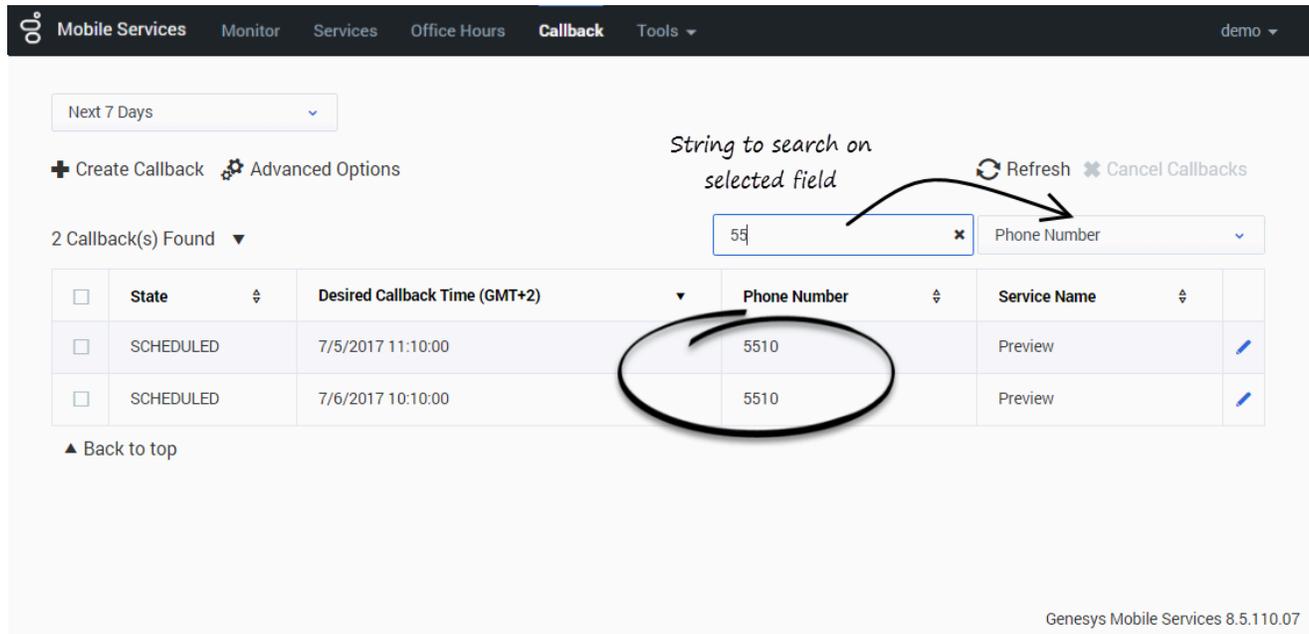
The following features are available:

- The drop-down at the top left gives you the option to display Callbacks using a pre-defined range for the past day, week, or month; or for the next day, week, or month.

- Refresh – You can force the interface to refresh the list of Callbacks.

- Search Box – You can select a column field in the drop-down at the right of the Search Box. Then, the interface filters the results dynamically as you type.

Limitations of the Interface

- The total number of callbacks that the table can display is 100000.

- Callbacks are displayed in pages of 100 items for Internet Explorer and 250 for other browsers.

- There is a dropdown which you can select to get pre-defined ranges or set a custom range.

Advanced Options - Opens a new window where you can customize how the Callback Management table displays:

- **Add Column Field** - You can add your own custom column field to display in the table. This option can be useful when you have created your own properties that you want to display in the table. You can add more than one field, which will display in the same column. In this case, to enhance readability in the single column, you can use the Alias option, and create a short name to display.
  To enable new Column Fields, configure disable-additional-columns-from-callbacks = true in your GMS application.

- **Filter Table by States and Service** - You can include one or more states to display in the table, and/ or you can select a service to display in the table. Note: You can only select one service at a time to display.

- Max # of Callbacks per Service - Default is 500.

Filtering Callbacks by User

Introduced in 8.5.207

If you configure a list of services in the **services** option, in the **gms** section of the Agent's configuration options, the Callback UI filters the callbacks that the administrator can see based on this configured list. See also Adding your Service to Virtual Service Groups.

Create or Schedule a Callback

You can submit or schedule a callback request by clicking **Create Callback**. A new dialog opens and you can add a new Callback to be displayed. You can select the **Callback Type** and **Service Name**, add your own properties to the Callback, and enter the following Callback Parameters:

- _customer_number - Your Callback number.

- _urs_virtual_queue - Queue to use for this Callback if several virtual queues are used for Callback with identical configuration. If you are defining the _urs_virtual_queue here, you must then remove this option from the Callback Service (through Genesys Administrator), because it cannot be defined in both places.

- _request_queue_time_stat - Queue statistics. For example, "ExpectedWaitTime;Queue;8999@SIP_Server;Environment".

> ### Important
>
> Make sure to set _wait_for_user_confirm to `false` in the selected service.

Edit a Callback

In the Callback table, for Callbacks that can be edited, a blue pencil displays in the last column. Clicking this pencil displays the edit options for that Callback.

- You can choose to reschedule the Callback (only for SCHEDULED Callbacks).

- You can choose to cancel the Callback.

Reschedule a Completed Callback

Introduced in 8.5.200

In the Callback table, you can reschedule completed Callbacks that show a blue repeat icon in the last column. Clicking this icon displays the Reschedule options for that Callback.

- The properties and user data of the completed callback are merged with the parameters of the newly scheduled callback.

- You can choose to cancel the operation.

Get Additional Details about the Queued State

You can expand the QUEUED state to get ORS-level diagnostics about the queued Callback.

The following information can be displayed:

- **Estimated wait time**: The estimated time that the customer will wait for the callback. For example: 46.5 seconds
- **Position in queue**: The callback's current position in the queue. For example: 3
- **Agents Logged in**: The number of agents that have logged in. For example: 3
- **Callback type**: The type of callback. For example: WAIT_FOR_AGENT
- **Callback version**: The version of the ORS Callback strategy. For example: v2.39

- **Channel**: The callback channel. For example: WEB

- **Dial attempt number**: The number of dials that the agent has attempted. For example: 2

- **Dial Result**: The result of the callback dial. For example: PUSH_DELIVERY_NOT_CONFIRMED

- **EWT at first outbound call**: The estimated wait time when the first outbound call happened. For example: 22.5

- **EWT at callback offer**: The estimated wait time when the callback is offered. For example: 0

- **Is snoozed**: Shows whether the callback is snoozed or not. For example: false

- **ORS session ID**: ORS session ID of the callback. For example: 00NEEH6C74C6NAC01G015B5AES000005

- **Position at first outbound call**: The callback's position in the queue when the first outbound call happened. For example: 3

- **Position at callback offer**: The callback's position in the queue when the callback is offered. For example: 3

- **Current priority**: The callback's priority. For example: 0

- **Routing target**: The callback's target or skill expression. For example: GMSCallbackAgents@stat.GA

- **Time of first outbound call**: The time when the first outbound call happened. For example: 1/3/2017 11:26:21

- **Time callback was accepted**: The time when the callback is accepted. For example: 1/3/2017 11:26:02

- **Time of next outbound call**: The time when the next outbound call happened. For example: 1/3/2017 11:27:24

Bulk Cancel and Export of Callback Records

Added in: 8.5.110

Important

First enable this feature in your GMS configuration by creating a features section in your GMS application, then by setting disable-bulk-cancel-and-export-callback to false.

Bulk Cancel

Make sure to select an appropriate Time Range to filter callbacks, then make a callback selection to activate the **Cancel Callbacks** button. When you click this button, the **Cancel Callbacks Confirmation** dialog opens, displaying the selected callback cancellations.
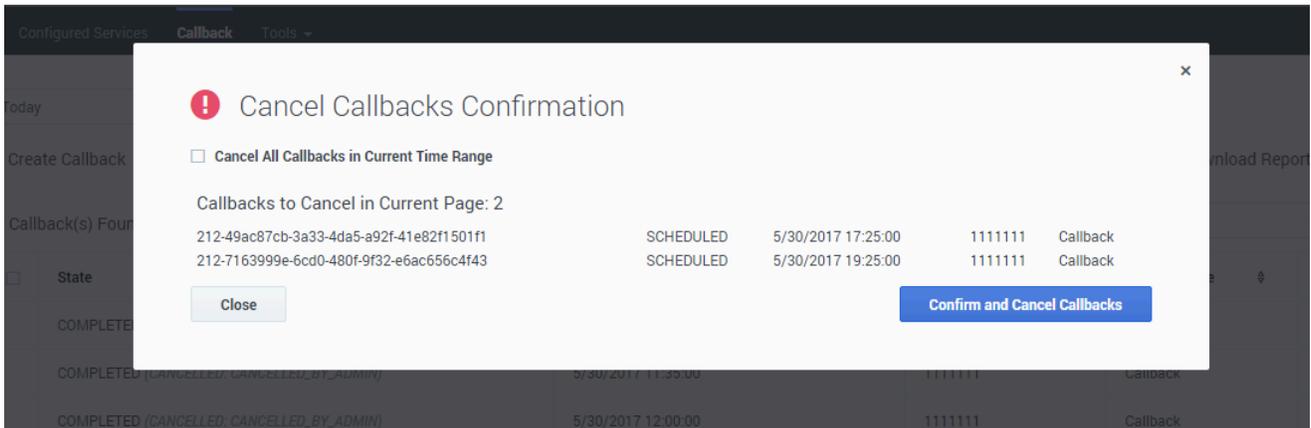
You can either:

- Continue with the selections you have made and click confirm to cancel them.
- Check **Cancel All Callbacks In Current Time Range**, to override your previous callback selections and delete all callbacks in the current time range. In that case, the list of displayed callbacks disappears from the dialog window. Confirm to cancel the records.
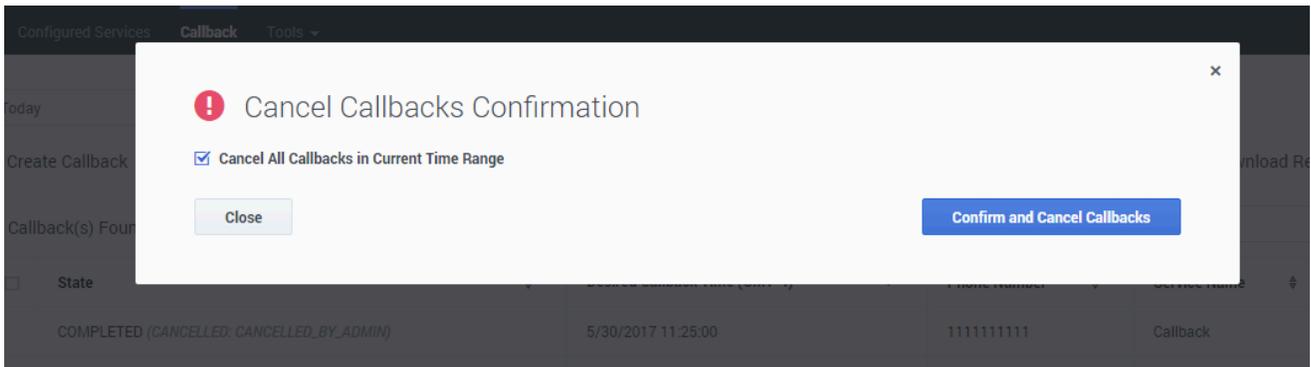
The resulting display shows all the callbacks that you have selected to cancel and that are now in COMPLETED(CANCELLED_BY_ADMIN) state.
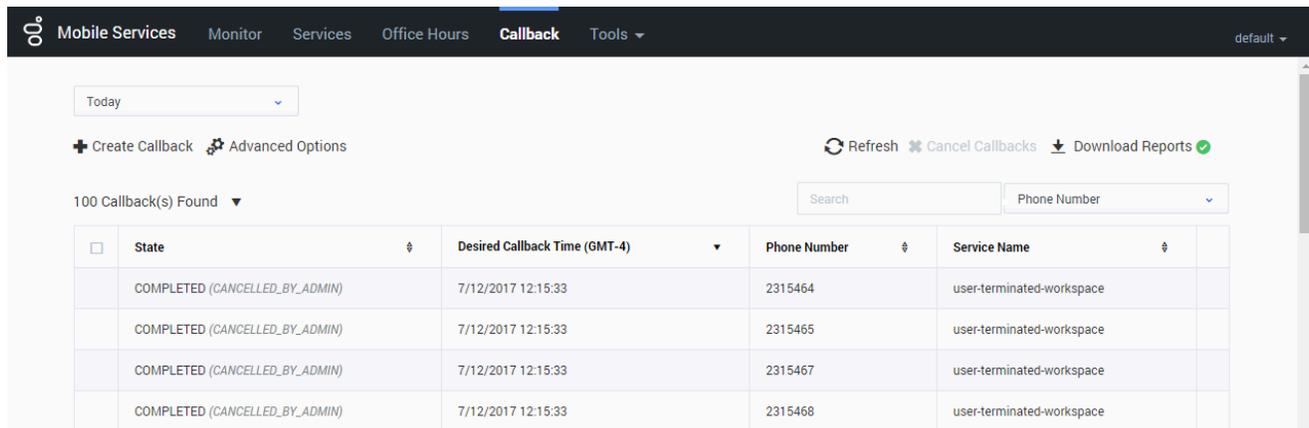
Download the Reports

Then, Download Reports dialog can provide two files:

- **Export Cancelled Callbacks** allows you to export a CSV file that includes all of the recently canceled Callbacks. Click the Refresh button beside the label to get an updated report if the one downloaded seems out of date.
- **Cancellation Summary Report** is available only if you recently canceled some callbacks. This report shows the cancellation status of the recently Cancelled Callbacks.
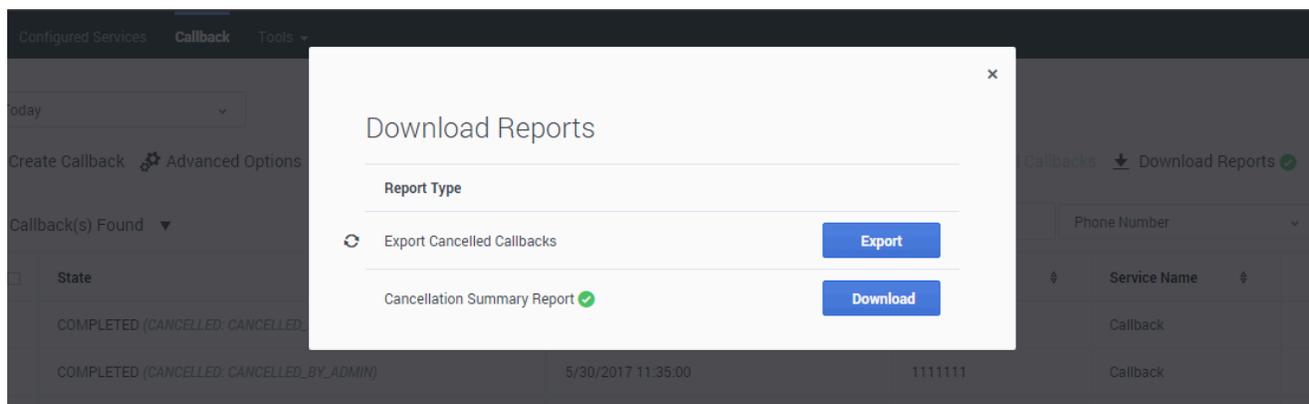
Configure Custom Date Range in Advanced Options

As detailed previously, the **Custom Date Range** selection in the drop-down list enables you to specify a start time and end time for the displayed callbacks. If you are often using the same date filters, consider saving these filters in advanced options.

1. In the **Callback** Panel, click **Advanced Options**, then **Add Custom Date Range**.
2. Add your date filters with appropriate names.
3. Once saved, they will appear at the bottom of the drop-down list.
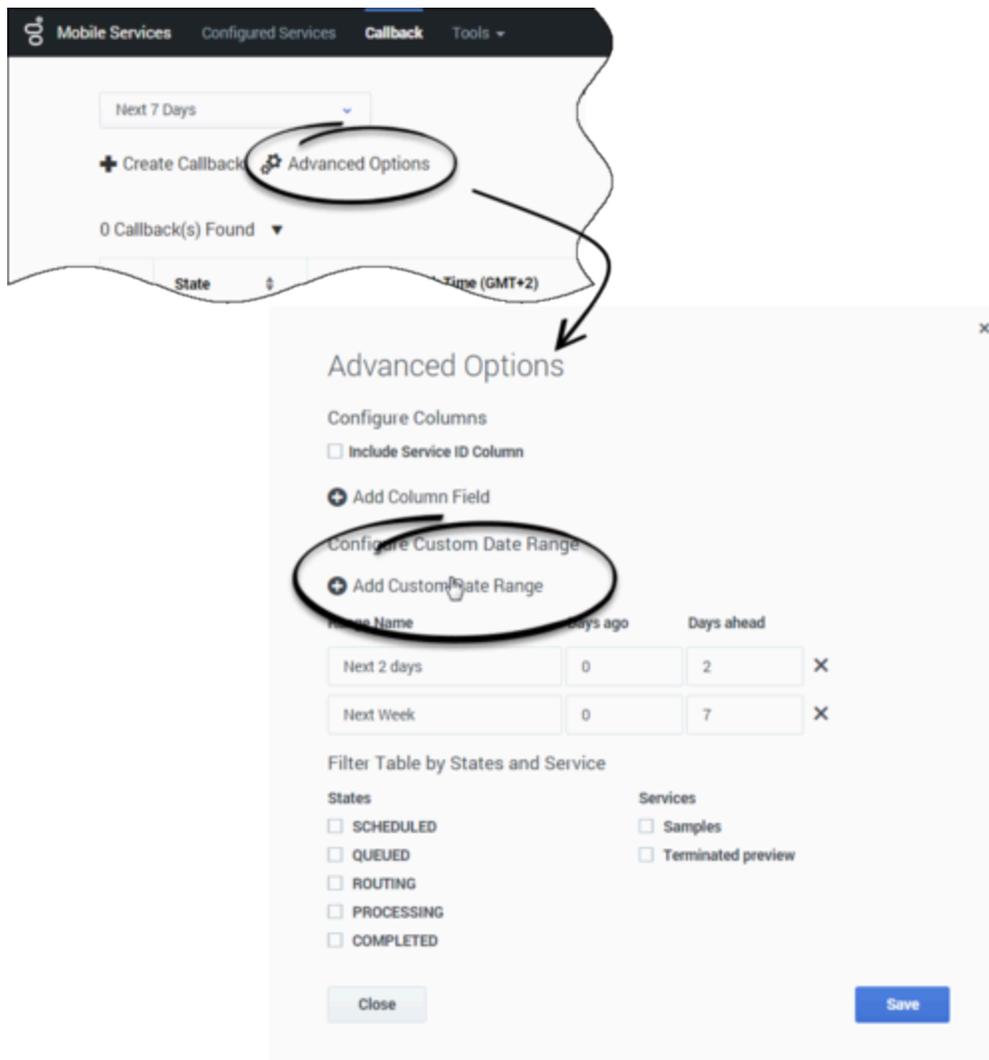
Configure Aliases to Display Custom Fields

Introduced in: 8.5.111

You can configure aliases for additional Column Names that display the custom fields passed in your Callback queries parameters. To do so:

1. Make sure that disable-additional-columns-from-callbacks is false in your GMS configuration.

2. Allow the list of the parameters that you wish to see displayed by using the filter-keys and returned-keys options.

3. Create a list of aliases for the parameters that you wish to see displayed by using the callback_column_alias option.

For example, if you pass the _LOB parameter in your callback queries, enable its alias as follow in the callback section:

```
filter-keys=_callback_state,_callback_reason,_request_queue_time_stat,
_request_ewt_service,_vq,_LOB
```

```
returned-keys=_desired_time,_callback_state,_callback_state,_callback_reason,
_ors_session_id,_LOB
```

```
callback_column_alias = {"_LOB": "Line Of Business"}
```


Then, the Callback UI will display the _LOB values in the **Line Of Business** column.

If you wish to display multiple custom values, the UI will group them in the **Custom Fields** column. For example, if you wish to display the values for the _LOB and _service_type query parameters, configure the alias as follow:

```
filter-keys=_callback_state,_callback_reason,_request_queue_time_stat,
_request_ewt_service,_vq,_LOB,_service_type
```

```
returned-keys=_desired_time,_callback_state,_callback_state,_callback_reason,
_ors_session_id,_LOB,_service_type
```

```
callback_column_alias = {"_LOB": "Line Of Business", "_service_type": "Service Type"}
```

	State	Desired Callback Time (GMT-4)	Phone Number	Service Name	Custom Fields	
☐	SCHEDULED	7/25/2017 14:00:00	10015	user-terminated-workspace	Service Type : Service 1 Line Of Business : Secondary	✏
☐	SCHEDULED	7/25/2017 14:00:00	10016	user-terminated-workspace	Service Type : Service2 Line Of Business : Special Projects	✏
☐	SCHEDULED	7/25/2017 14:00:00	10013	user-terminated-workspace	Line Of Business : Primary	✏
☐	SCHEDULED	7/25/2017 14:00:00	10017	user-terminated-workspace		✏
☐	SCHEDULED	7/25/2017 14:00:00	10011	user-terminated-workspace	Line Of Business : Primary	✏
☐	SCHEDULED	7/25/2017 14:00:00	10020	user-terminated-workspace		✏
☐	SCHEDULED	7/25/2017 14:00:00	10014	user-terminated-workspace	Service Type : Service3	✏
☐	SCHEDULED	7/25/2017 14:00:00	10019	user-terminated-workspace		✏
☐	SCHEDULED	7/25/2017 14:00:00	10012	user-terminated-workspace		✏
☐	SCHEDULED	7/25/2017 14:00:00	10018	user-terminated-workspace		✏

Custom Date Range   7/25/2017   7/26/2017   Submit

➕ Create Callback   ⚙ Advanced Options   🔄 Refresh ✖ Cancel Callbacks ⬇ Download Reports

10 Callback(s) Found ▼   Search   Phone Number

▲ Back to top

Then, the Callback UI will display the _LOB and _service_type values in the **Custom Field** column. The aliases will identify each custom field.

Note that you can also use the aliases to search and filter the list of displayed Callbacks.

> **Tip**
>
> Keys for Custom Fields do not need to start with underscores: you can use _LOB or LOB for the name of your Custom Field.

Context Services Interface
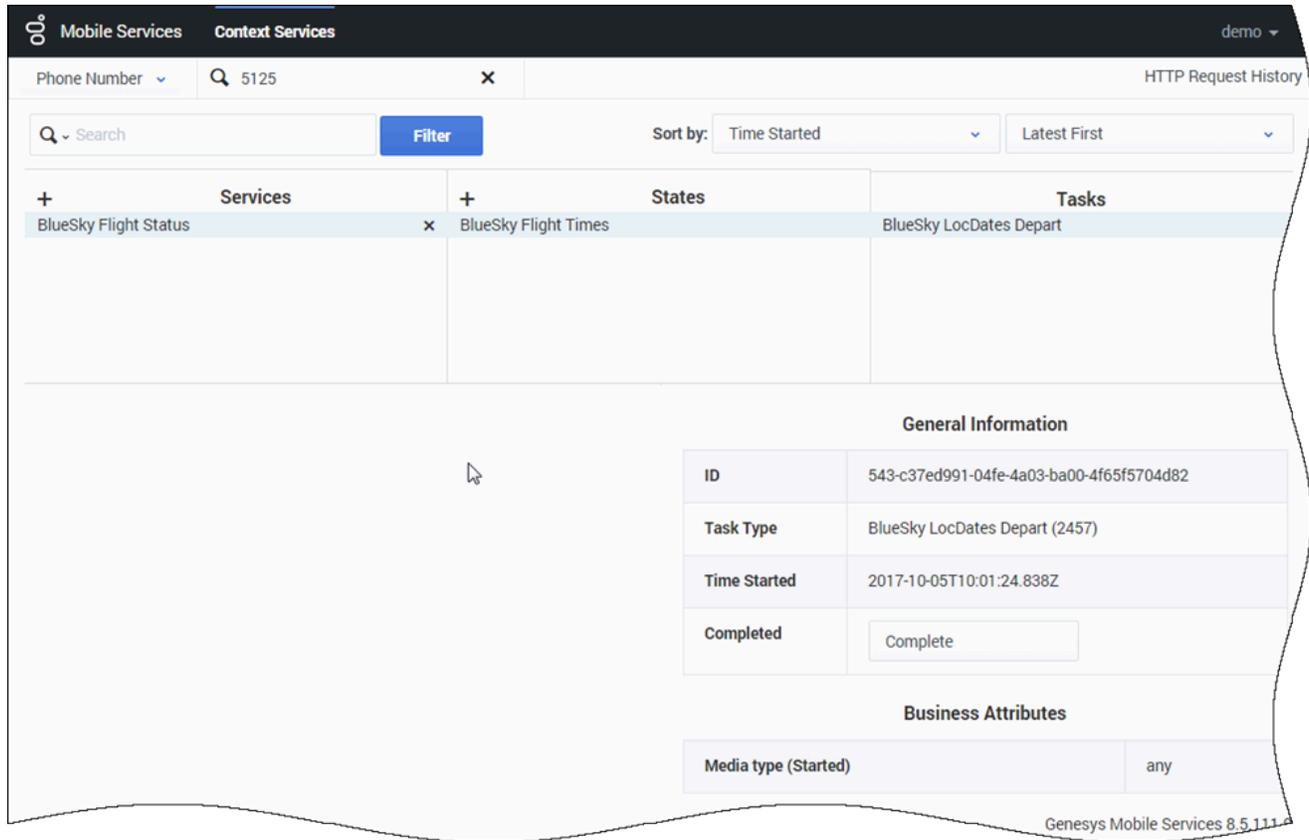
The Context Services Interface is a web-based interface that enables edition of Context Services data. This interface is intended to be used by developers and supervisors looking for detailed information about services because it is built to search for profiles, services, states, and tasks based on ID information or UCS information. It does not include all the search abilities that are available in typical agent interfaces.

This interface also enables you to modify or delete a given service.

> **Important**
>
> 1. The Context Services Interface is available only for single-tenant installations.
>
> 2. If you change business attribute values in your configuration, users will need to refresh the Context Services Interface to see the changes taken into account.

Login Panel
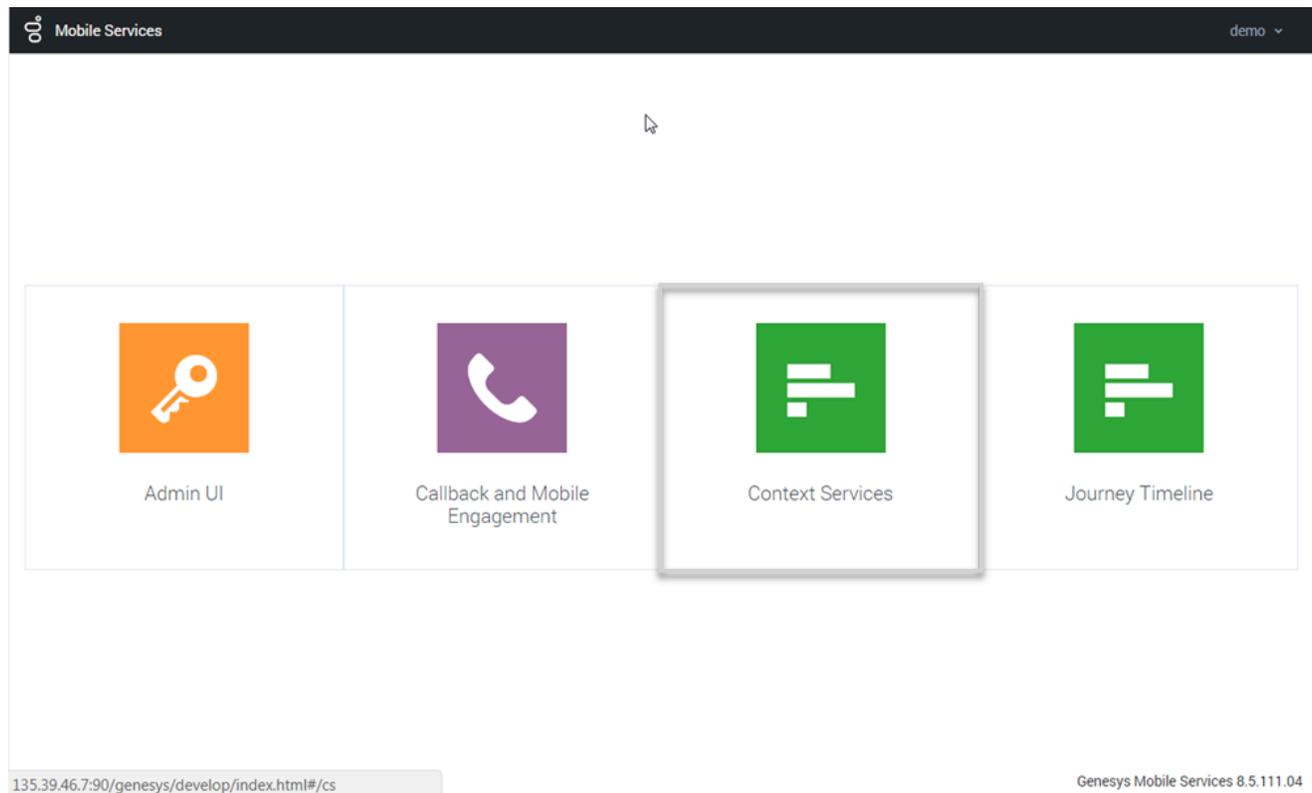
The Context Services Interface is available as part of the GMS Service Management User interface (detailed in the Service Management Help).

- To access this interface, you must login as a user who owns the Administrator or Supervisor role.
- Then, you can select the `Context Services` icon.

Search for Services
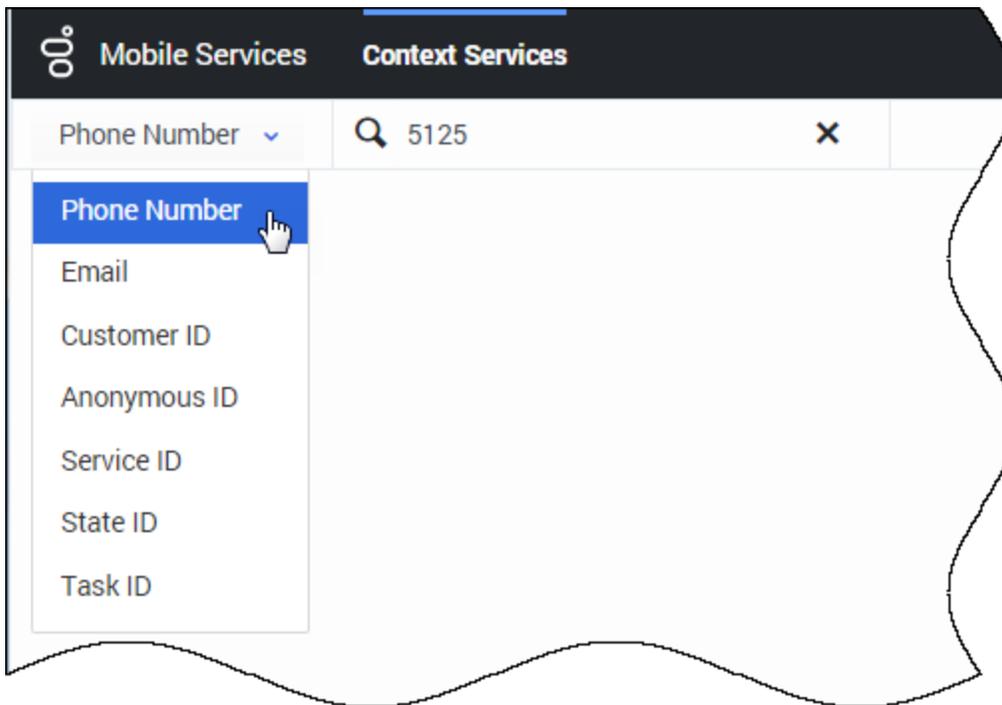
First, you start by searching for services or customers in the Context Services panel. You can search for UCS keys or Service, State, and Task IDs. These fields must be identical to a key in the UCS database to work correctly. There is no automatic completion available.

Select a key in the Search drop-down menu, then enter a value in the Search text box. The value must match a UCS entry to provide a result.
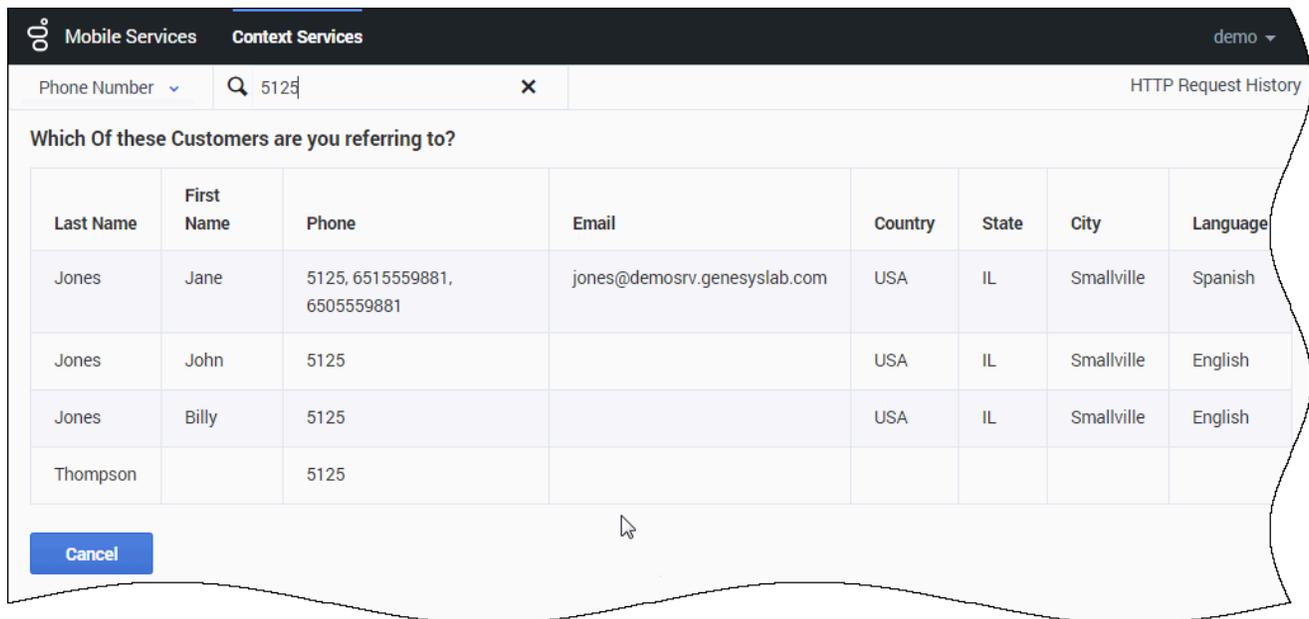
The interface displays a list of results. Select an item in the list.

You can then use the interface to modify the service.

Manage your Services
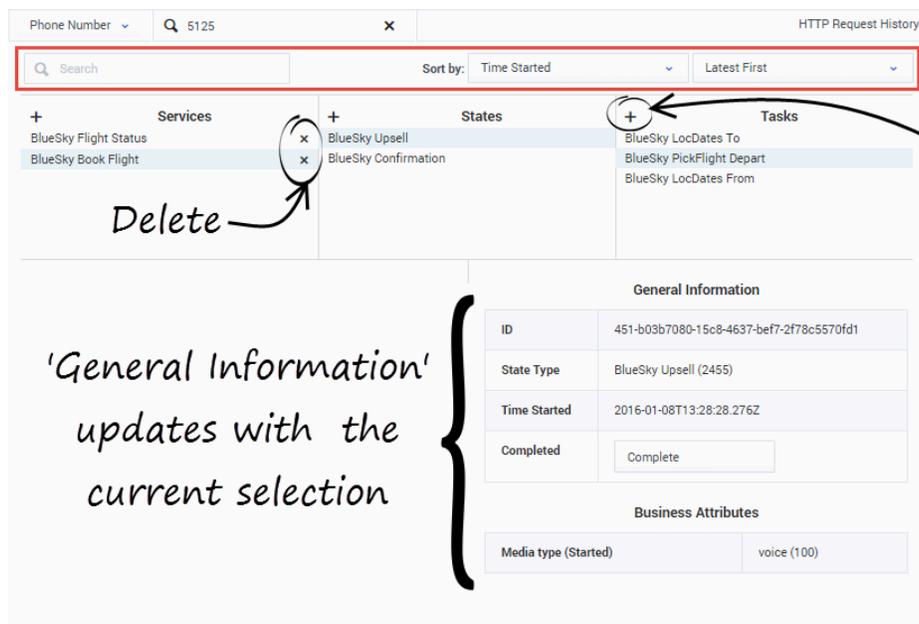
Filter the results

Create new services, states or tasks

Delete

'General Information' updates with the current selection

The interface lets you manage the list of objects that you selected. If you selected a customer instance, you get the complete list of objects associated with the Customer ID.

- You can use the sorting tools to change the list displayed.

- You can select an item in the list, and get more details about the object.

- You can delete an object by clicking the 'x' icons.

- You can use the Action menu to perform more actions, such as creating new services, states, tasks.

- You can add services, states, and tasks by clicking the '+' icons.

- You can get a history view of your queries by clicking **HTTP Request History**.

Create a Service, State, or Task

Select a type

State ID

Finish by starting
the service

Add extensions

You can add services, states, and tasks by clicking the '+' icons. To create states or tasks, first, select a service or a state, then choose a type. When you create a new resource, you can fill it and even add some extension data.

Complete a Service, State, or Task

If you can complete a Service, State, or Task, the interface displays a Complete button, that opens the Completion panel.

Journey Timeline Interface
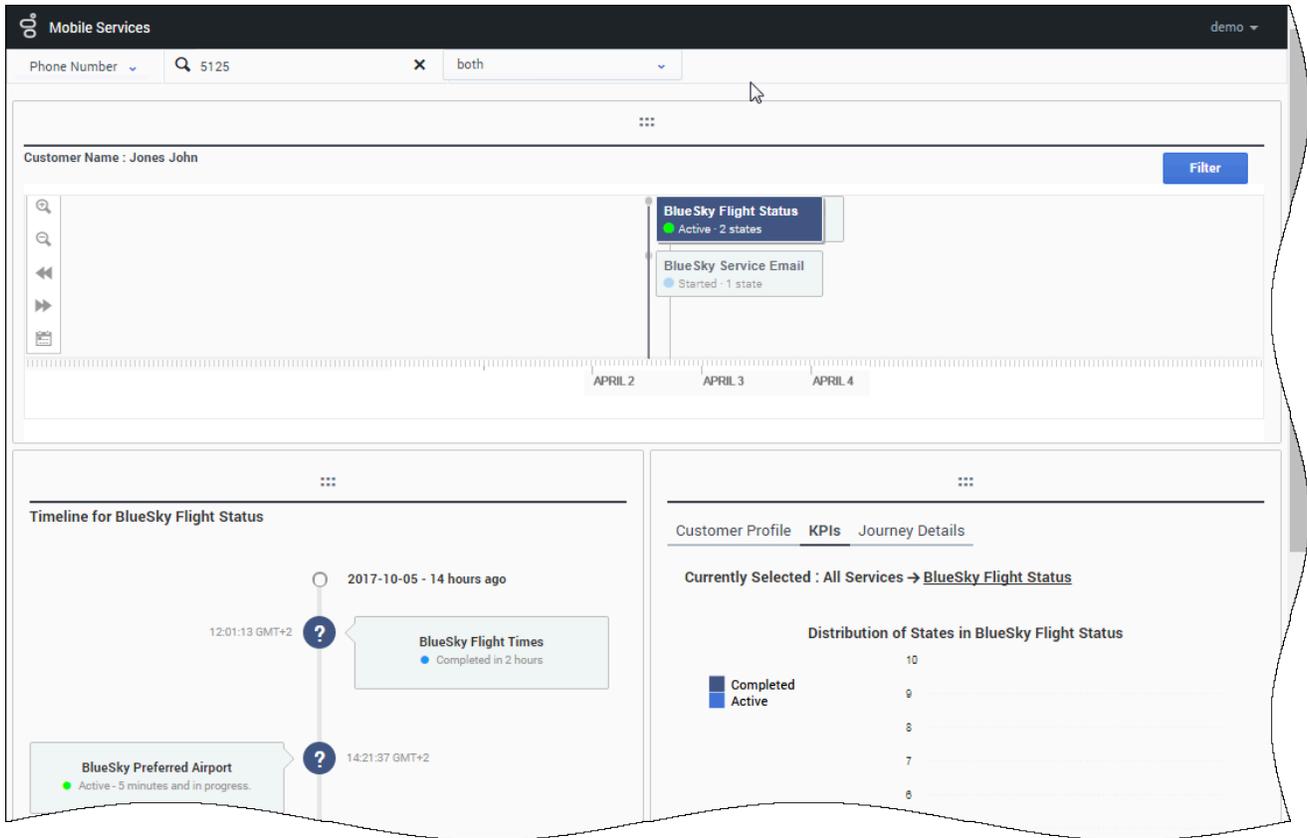
The Journey Timeline is a web-based interface that provides a visualization of Context Services data. This interface is intended to be used by developers and supervisors looking for detailed information about a specific customer.

[+] Tell me why

This interface is built to search for profiles, services, states, and tasks based on ID information or UCS information. It does not include all the search abilities that are available in typical agent interfaces.

Starting in 8.5.103, you can customize this interface.

> **Important**
> The Journey Timeline is available only for single-tenant installations.

Accessing the Journey Timeline

Journey timeline can be accessed from the Context Services interface which is available as a standalone application. The steps to download and install the service is detailed in Installing Standalone Context Service.

> ### Important
>
> If you do not see the Journey Timeline item, it means that you did not enable Context Services properly.

Search a Customer with Customer Journey
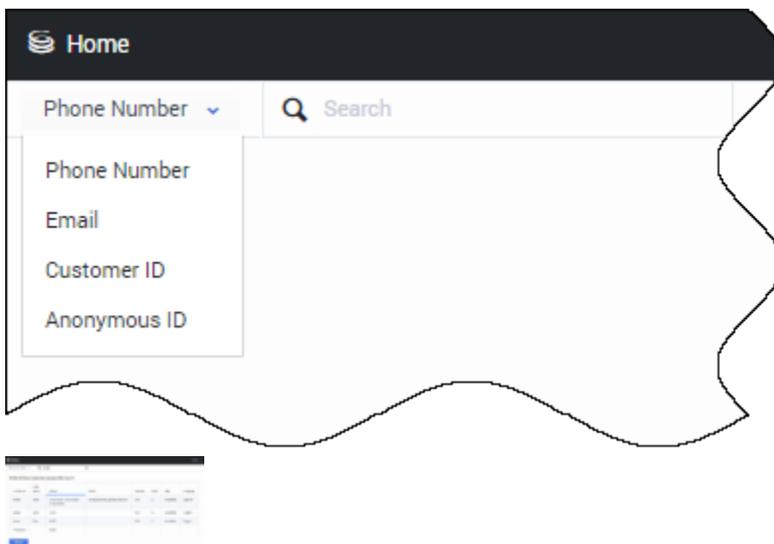
You can query a user based on the email address, phone number, and name fields. These fields must match a value in the UCS database to work correctly. There is no automatic completion available.

1. Select a key in the search drop down menu, then enter a value in the Search textbox. The value must match a UCS entry to get a result.

2. The interface displays a list of results. Select a customer in the list to display the customer's timeline.

> **Important**
>
> You can customize the searched items through JSON configuration. See Customizing Profiles.

Manage the Timeline

Filter the services here
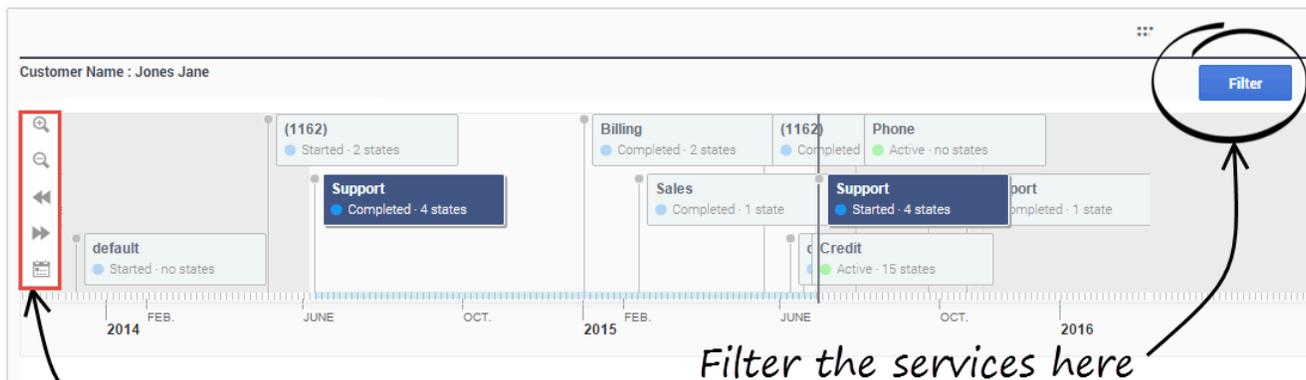
Expand or contract the timeline

The timeline shows all the customer's services and their current status (active, inactive). If you select a service, Customer Journey displays the list of states for the given selection.

- You can manage the timeline (expand or contract) by using the icons in the left menu sidebar.
- You can zoom or navigate to services by using the icons in the left menu sidebar, or you can simply left click in the timeline to move it.
- You can also filter the displayed services.

Display States and Tasks
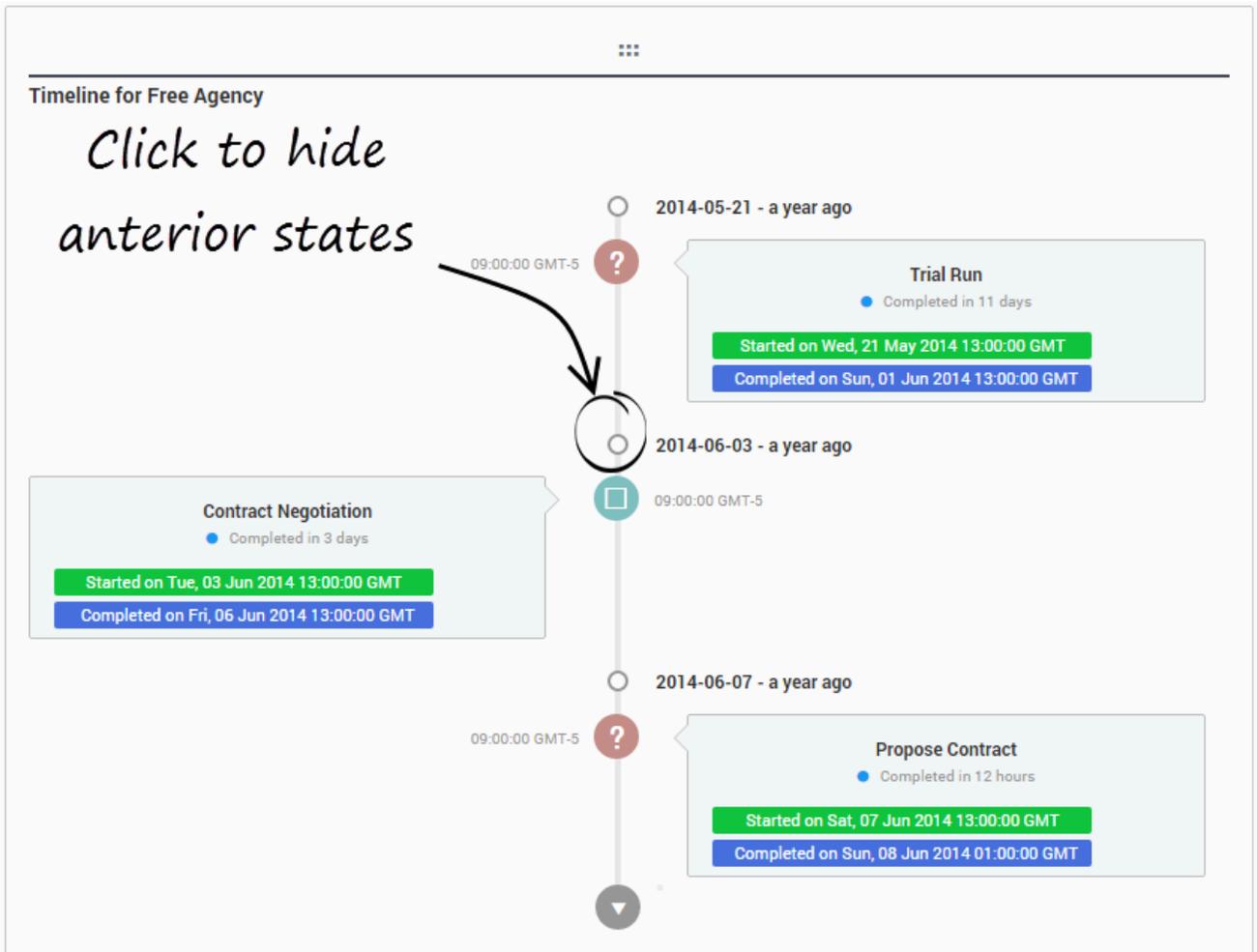
If you select a service in the timeline, for instance, Agency, its nested states and tasks are displayed in the Vertical Timeline. You can then select one of them to get KPIs, customer or journey details.

Important

You can customize the display through templates. See the guidelines in the Developer's Guide.

Display Details related to Service and State Selection

The selection in the horizontal and vertical timelines automatically update the information area.

> ### Important
> You can customize the display through templates. See the guidelines in the
> Developer's Guide.

Integrate with UCS Profiles

If you enabled Customer profiles in UCS, you can integrate Journey Timeline with UCS profiles as follows:

- Install and set up NGINX on your local machine
- Ensure that the NGINX configuration includes the following information:

```
location /genesys/1/cs/profiles { proxy_pass http://<location of UCS>/profiles; }
location /genesys { proxy_pass http://localhost:8080/genesys; }
```

You can read more details here.

More about Built-in and ORS services

Before you start

Genesys Mobile Engagement provides a set of templates that you can use to create your customized workflow. Templates are already loaded at the product installation and show up in the **Templates** panel.
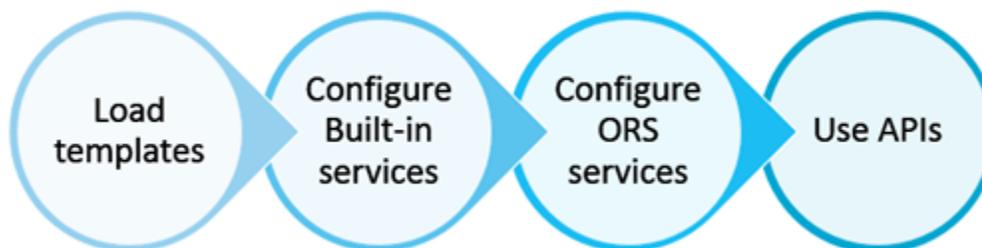
Templates include two types of services:

- Built-in services of type `builtin` that are basic services executed in the Genesys Mobile Services server. They provide fixed functionality that you can tune only through configuration options in the **Configured services** section.
- Orchestration Server-based (ORS) services of type `ors` that implement ORS scenarios. The implemented scenarios depend on the ORS service configuration.

These templates enable you to create services that provide the GMS APIs.

Note that `ors` and `builtin` services may implement some cross-dependencies. You may also be interested in reading the following page, that lists examples of flow diagrams. These diagrams show how you can use `request-access` and `request-interaction` services to implement Access Number Allocation.

> ### Important
>
> The Callback services are executed in the Orchestration Server and managed in the Genesys Mobile Services server. See the Scenarios section of the *Callback User's Guide* for more information.

Relationship between Configured Services and API queries

When you create a new {service-execution-name} service in the **Configured Services** section of the Service Management UI, this service is also created in the service.{service-execution-name} section of your GMS configuration.
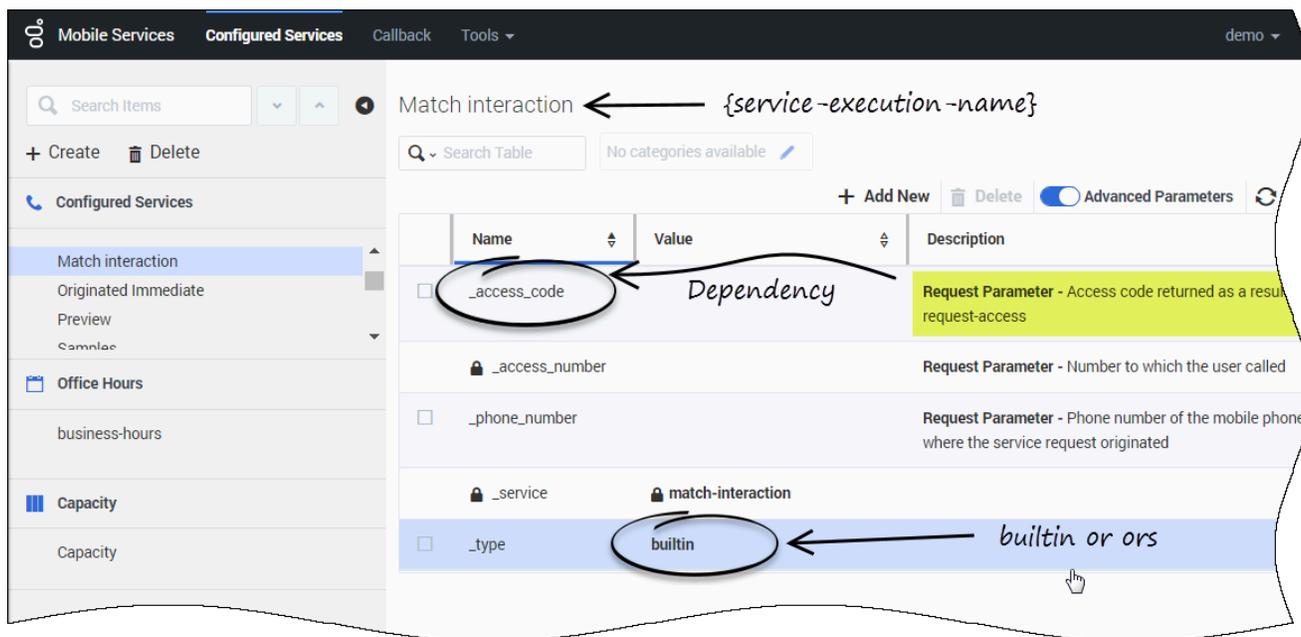
The URLs used by the Service API are dependent on the execution name of the service that you have just created. Services are available at the following URL:

http://<host>:<port>/genesys/1/service/{service-execution-name}

For instance, if you create a service named match-interaction, then {service-execution-name} is match-interaction and the service is available at:

http://<host>:<port>/genesys/1/service/match-interaction

Important

To use a service, you should start by allocating resources to this service with a create service request. Note that for some builtin services, this may not be necessary.

Customized ORS Scenarios
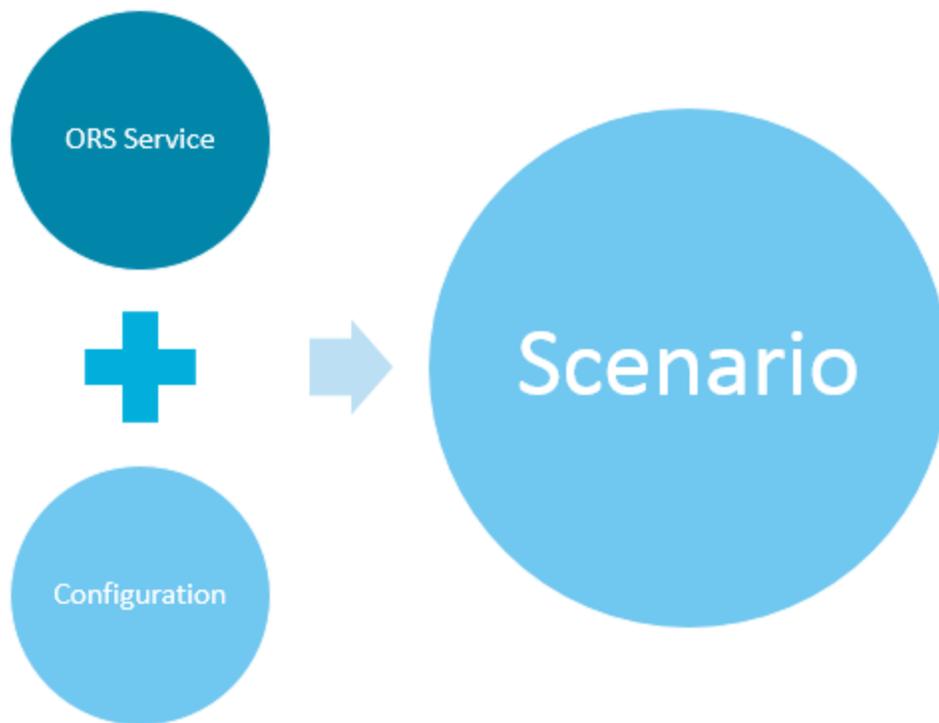
To implement a customized ORS scenario, you will need to **create** an ORS service and select the **Callback** template.

Choose your scenario in the list that the template displays. Then, refer to the pages of this chapter for configuration details and sequence diagrams.

[+] See the list.
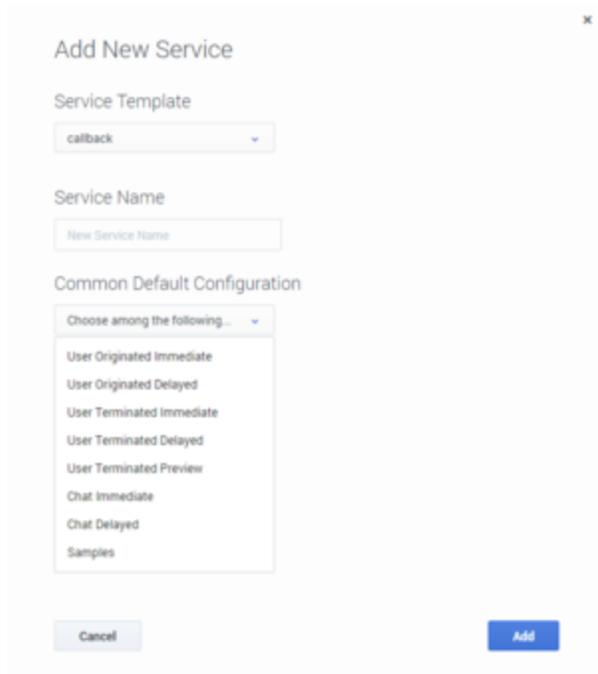
Advanced Customization

If you are an advanced user of Composer, you can customize the SCXML and VXML of the Classic Callback sample that includes a Composer project.

Services Cross-Dependencies

According to the services that you plan to use, you may need to create and configure the following services in the **Configured Services** panel, even for services of type `builtin`.

- **request-access** — Use the `request-access` template to create a **request-access**.
- **match-interaction** — Use the `match-interaction` template to create a **match-interaction** service.

The following table presents the builtin and ORS services that require one of these services (or both).

Builtin or Scenario name	request-access	match-interaction
match-interaction	✔	

Builtin or Scenario name	request-access	match-interaction
request-interaction	✓	✓
User Terminated Delayed Voice (notification)	✓	✓
User Terminated Immediate Voice ORS Service	✓	✓
User Terminated Scheduled Voice	✓	✓
User Terminated Delayed Voice Agent Preview	✓	✓

Request-interaction

Type: **builtin**

The Request-interaction scenario illustrates a Simple Voice Inbound–Immediate Call.

Overview

This is a basic service that helps an application/end user contact the contact center. It has the following characteristics:

- It supports only customer initiated voice contacts.

- It stores and maintains application data with the service.

- It returns access information in the response of the Create API.

- It supports very basic access number allocation (random and locking)

- It supports reserving the access information when allocated for the application for a configurable period of time.

- It support the following types of access information:

  - Access Number (DNIS) which is to be called by the application

  - Access code which is to be supplied by the customer/application when the contact is being established. This provides an extra level of authentication.

Sequence Diagrams

request-interaction - No Delay
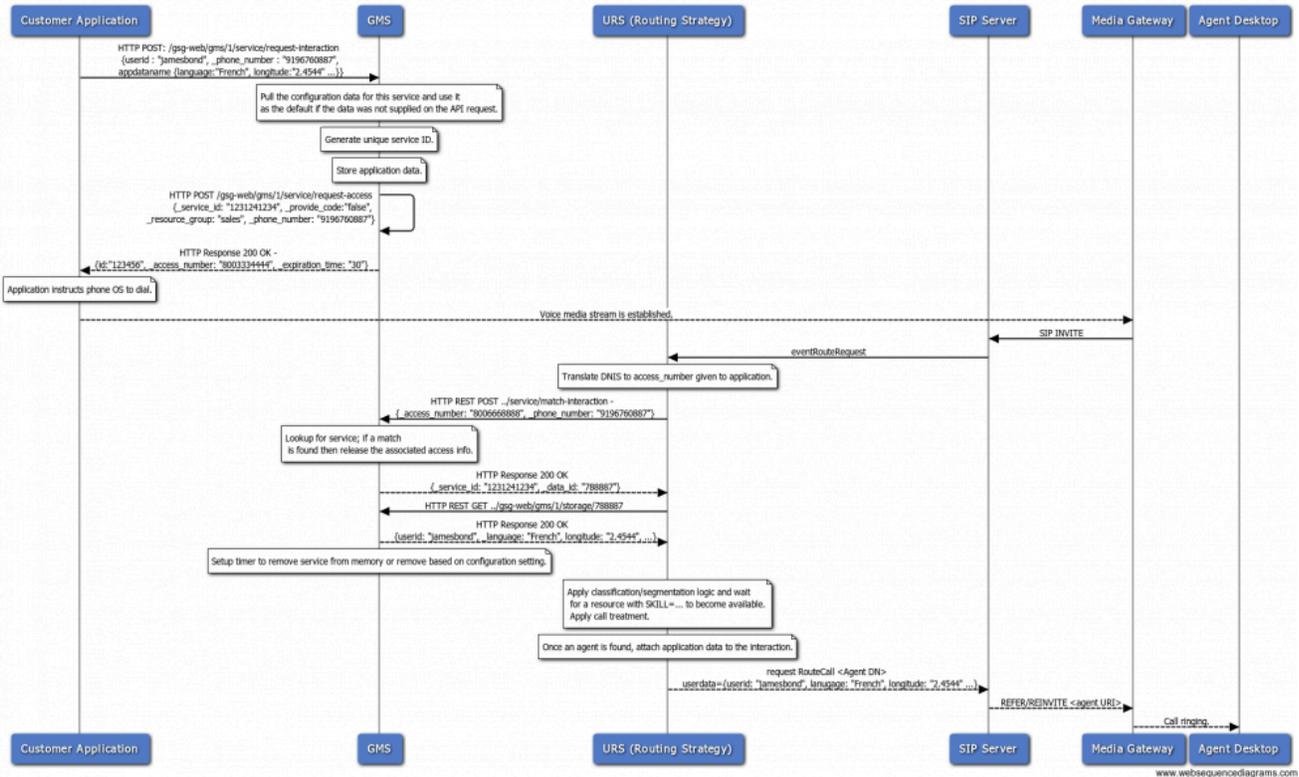
request-interaction - No Delay, access code
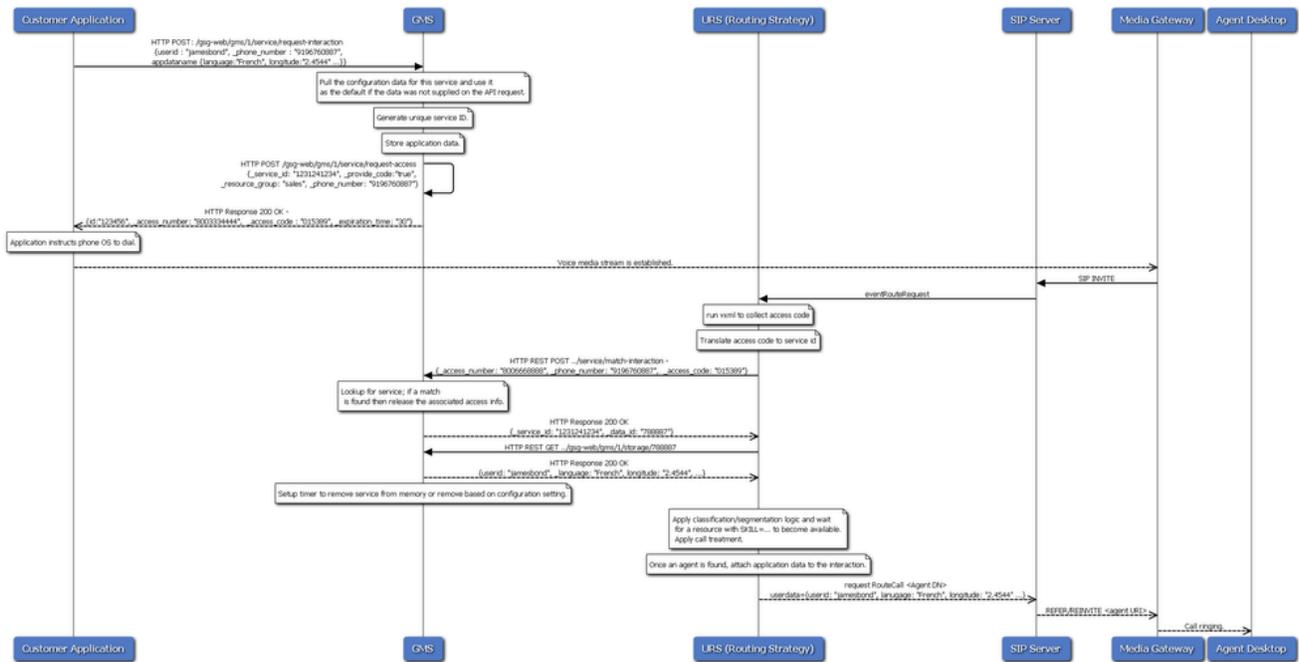
Request, Response, and Events Reference

Create Request Data

These are the service specific parameters that will be supplied on the Create service API.

Parameters	Mandatory	Description
_phone_number	No	The phone number of the device that the application is running on. This data will be used to match the specified data when the device/application calls the supplied access number.
_provide_code	No	This indicates if the service should return an access code which will add more security and reliability when trying to correlate the incoming call with the service. The value is a boolean. If not present, then GMS will use the value that was configured for the service. If it is not configured and not supplied on the Create API request then

Parameters	Mandatory	Description
		the value will be false.
_resource_group	No	This identifies the type of resource group that is need to help this end user. This maps to a configured set of access numbers. If not present, then GMS will use the group that was configured for the service. If it is not configured and not supplied on the Create API request then the request will be rejected.
{appdataname}	No	This is data that is supplied by the application and used to help the contact center resources better service the end user. The application can supply as many application data parameters as they want. These parameters may be string values or files. They should add to the multi-part structure in the body.

Create Response Data

These are the service specific parameters that will be supplied on the Create service API response.

Parameters	Mandatory	Description
_access_number	Yes	This is the access number which was allocated for this application. The application should use this number to contact the contact center.
_access_code	No	This is the access code that should be supplied by the application or end user when the call is established to further authentication the application/user. This will be present when the Create API specifies that it needs a access code (_provide_code = true).
_expiration_time	No	This is the amount time (in seconds) that this access information will be locked/reserved for the service.

Specific Requests

There are no specific requests for this service.

Events

There are no events associated with this service.

Customization

You can customize your own services based on the request-interaction service. The way you do this is by defining your custom service in the Mobile Engagement UI.

1. Create a request-interaction service.

2. You then specify the configuration options and appropriate values for your service. These options are detailed in the request-interaction Section of the Service Options reference.

Most of these options are parameters that will be passed to a request-interaction service but have been given pre-defined values via configuration. For details on the configuration options see the section below. This enables you to simplify the API signature for your service. Once the new service is defined, the application can use it. The following is an example:

Configuration Options

You can configure the following options in the Mobile Engagement UI if you create a customized service based on the request-interaction template:

Option	Description
_service = **request-interaction**.	• For Genesys Mobile Services-based services: The name of the matching service. • For Orchestration Server-based services: The URL of the service's SCXML application. • Other SCXML Callback services: The URL of the service's SCXML callback application. In that case, you must set _service_type to callback.
_type = **builtin**	• For Genesys Mobile Services-based services: builtin • For Orchestration Server-based services: ors
_ttl	Specifies the default time to live for all stored data in seconds.
_resource_group	Resource group from which the access number

Option	Description
	must be allocated.
_provide_code	If set to true, the service returns the access code along with the access number.

> **Important**
>
> If you pass one of the request-interaction parameters in a Create API query for a customized version of the service, the configuration option value will supercede the passed value (that is, the passed value will be ignored).

Example

You can create an iPhoneService section under the services section and set the following configuration options:

Option	Value
_type	built-in
_service	request-interaction
_ttl	7200
_provide_code	false
_resource_group	iPhoneService

The following is the example Create API invocation for the iPhoneService service:

```
Request URL:http://localhost:8080/gms-web/gms/1/service/iPhoneService
Request Method:POST
Accept:*/*
Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.3
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:xxxx
Content-Type:multipart/form-data; boundary=----Boundary
Request Payload
------Boundary
Content-Disposition: form-data; name="_phone_number"
6504669999
------WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="current_location_latitude"
48.8583
------WebKitFormBoundaryy16qocbN6tmPORZL
Content-Disposition: form-data; name="current_location_longitude"
```

```
2.2944
------WebKitFormBoundaryy16qocbN6tmPORZL
```

Get and Basic Get

Type: **builtin**

This is a *get* service that returns node information. You can use the *get* service (with the associated Service Template) to isolate a GMS node based on GMS status. This service has the following characteristics:

- It returns an HTTP response code of either 200 OK or 503 Service Unavailable, by default.

- For 200 OK responses, it also returns all options specified in the configuration.

You can create a service of this type in your environment, and configure your load balancer to use the service as the health check. To take a node out of service (for load balancing purposes and based on your load balancer settings), simply configure the service to return the required HTTP code (503, 404, and so on).

Sequence Diagrams

[+] Show diagrams

Health Check (GMS is offline)

LoabBalancer   **GMS Node**

HTTP GET /genesys/1/service/lb-service

Check status, status is OFFLINE

503 Service Not Available

LoabBalancer   **GMS Node**

Change GMS status

Administrator   **GMS Node**   **SCS**

HTTP POST /genesys/1/admin/changestatus/OFFLINE

Check if user is autenticated,
if yes, process the request

Change Application status to "Suspended"

Ok

200 OK

HTTP POST /genesys/1/admin/changestatus/ONLINE

Check if user is autenticated,
if yes, process the request

Change Application status to "Running"

Ok

200 OK

HTTP POST /genesys/1/admin/changestatus/SHUTDOWN

Check if user is autenticated,
if yes, process current requests
and wait for all requests to finish.
Wait 10 seconds and stop GMS

200 OK

Administrator   **GMS Node**   **SCS**

Upload and Configure the Get Service

Using the Service Templates Interface, load the Get Service service template, and configure the Get service.

The Mobile Engagement UI provides the following attributes:

Option	Description
_service = **get**	- For Genesys Mobile Services-based services: The name of the matching service. - For Orchestration Server-based services: The URL of the service's SCXML application. - Other SCXML Callback services: The URL of the service's SCXML callback application. In that case, you must set _service_type to callback.
_type = **builtin**	- For Genesys Mobile Services-based services: builtin - For Orchestration Server-based services: ors
_online_code	HTTP code to return when GMS is ONLINE.
_offline_code	HTTP code to return when GMS is OFFLINE.

You can also add any custom values as key-value pairs by clicking **Add New** in your Get service. These custom values will be returned in the service responses.

> ### Warning
> Do not use a leading underscore with the custom parameters (for example, _my_custom_key). Custom parameters with leading underscores will not be returned in responses from the service.
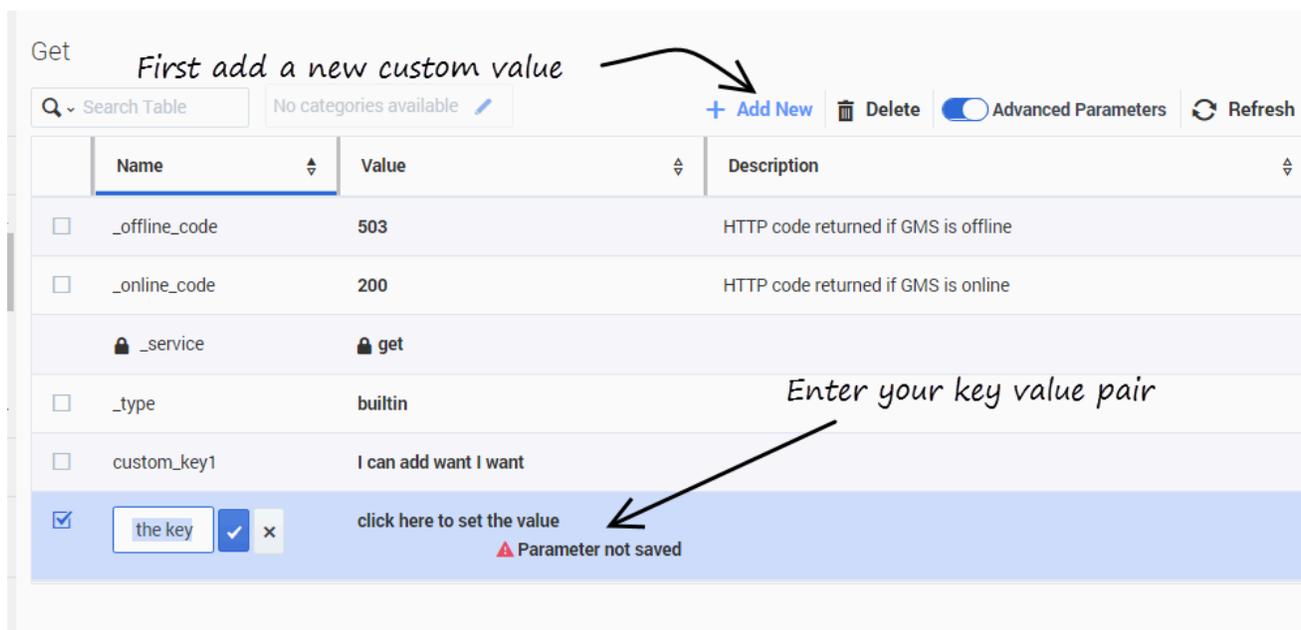
Get API

Enable / Disable Node

Enables changing the GMS node status:

- ONLINE: GMS is OK to process requests.
- OFFLINE: GMS is running, accepts requests, but Load Balancer will be aware that it will need to remove this GMS from the active GMS list (for future maintenance). The Application in Solution Control Interface (SCI) will appear as Suspended.
- SHUTDOWN: GMS shuts down.

> ### Important
> The following two URLs are protected by Basic Authentication:
> - POST http://127.0.0.1:8080/genesys/1/admin/node/changestatus/OFFLINE
> - POST http://127.0.0.1:8080/genesys/1/admin/node/changestatus/ONLINE

Operation

Method	POST		
URL	/genesys/1/admin/node/changestatus/{status}		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
status	string	mandatory	ONLINE, OFFLINE, SHUTDOWN

Response

HTTP code	200
HTTP message	OK

Get Node Status

Operation

Method	GET		
URL	/genesys/1/service/{serviceName}		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
serviceName	string	mandatory	Name of the builtin "GET" service defined in Service Management UI.

Response

HTTP code	200
HTTP message	OK
Body	A JSON object with list of defined parameters

Example

To perform this request, you must first create a service of type Get in the Admin UI, called
LoadBalancer-checker.

```
GET http://127.0.0.1:8080/genesys/1/service/LoadBalancer-checker HTTP/1.1
Accept-Encoding: gzip,deflate
gms_user: dd
Host: 127.0.0.1:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Response in case of GMS being online:

```
HTTP/1.1 200 OK
Date: Tue, 12 Nov 2013 15:38:55 GMT
```

```
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json;charset=UTF-8
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{"custom_parameter_1":"value1","custom_parameter_2":"value2"}
```

Basic Get Service

This is a *basic_get* service that returns service information. You can use the *basic_get* service to isolate a GMS service based on the Configuration Manager option. This service has the following characteristics:

- It returns an HTTP response code based on the `result` option defined in Configuration Manager. If the result is 404, a `404 error` will be thrown on the service response.

- It returns a JSON list of values defined in Configuration Manager if the value of the `result` option is 200.

Example

The following table shows an example for the **section service.basicget** service in Configuration Manager:

Option Name	Option Value	Description
_service	`basic_get`	Service name.
_type	builtin	Genesys Mobile Services-based services.
error_code	404 Not Found	HTTP error code.
result	404	HTTP response code.

Where:

Option name	Option type	Restriction on value	Description
result	String	Mandatory Valid values: 200,401,403,404,503, and so on.	A valid HTTP response code (See rfc2616). If result=200, the response body will contain a JSON list of options defined in Configuration Manager (options that do not have a leading underscore, for example, `error_code` or `result`).

Example of response:

```
[service.basic_get]
_service=basic_get
_type=builtin
error_code=200 OK
result=200
HTTP Response:

{
    "result": "200",
    "error_code": "200 OK"
}
```

Urs-stat

Type: builtin
Updated in: 8.5.109

Create a GMS built-in service using the **urs-stat** template that provides the following benefits:

- Statistics caching of the statistic to reduce load on URS. The `_caching_policy` parameter sets the cache period in seconds (see below).

- Load balancing and scaling across multiple GMS nodes.

- A single point of contact for your app.

Create a urs-stat Service

To create this GMS built-in service, select the **urs-stat** template when creating a new service.
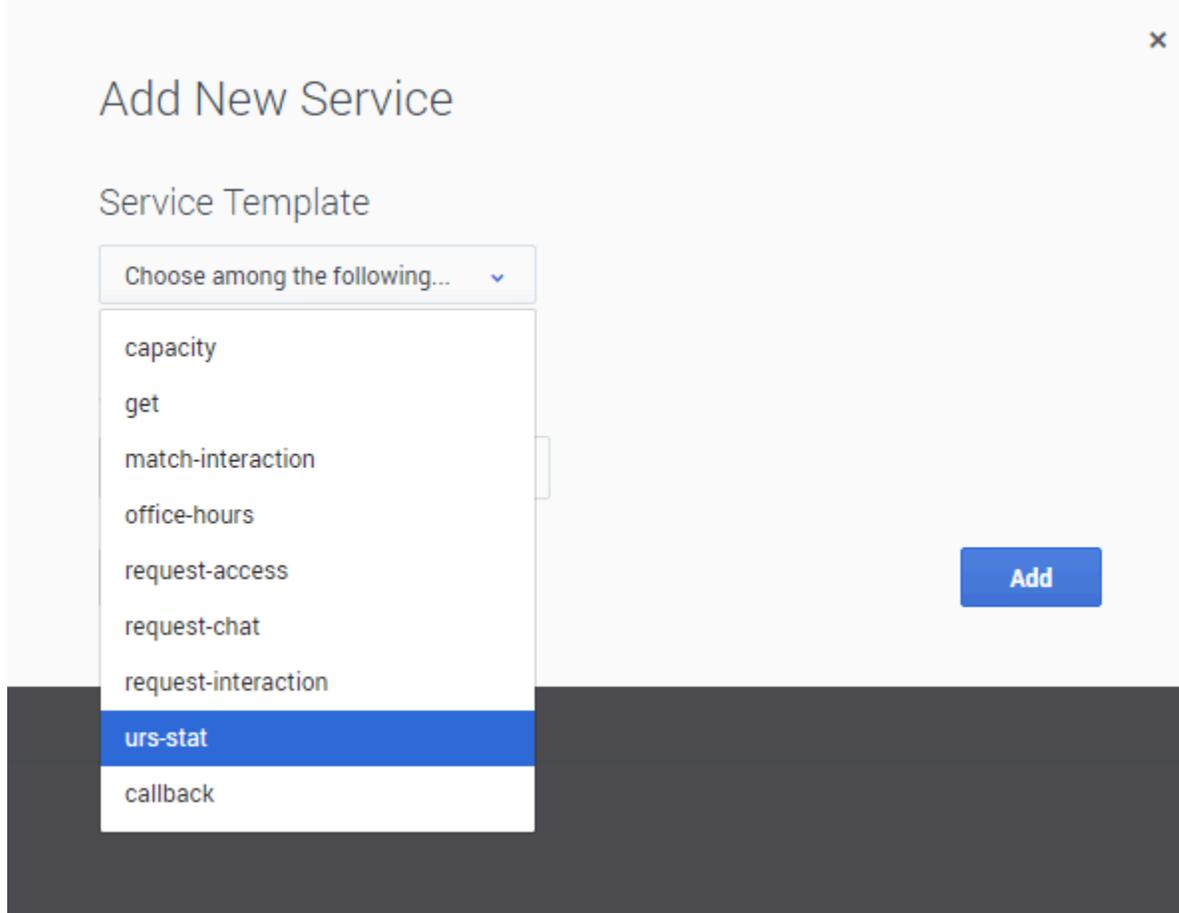
Configure urs-stat parameters

Configure the following parameters in your <name-of-urs-stat-service> service:

Option	Description
_urs_url	The URS URL formatted as follows: http://<urshost>:<ursport>/urs/call/max/lvq. This option can also point to the load balancer in front of the URS.
_urs_stat_url_parameters	Additional URS lvq input parameters (url-encoded format). For example: name=<VQ_Name>&tenant=<Tenant_Name>&aqt=urs
_caching_policy	URS Statistic caching policy in seconds.

For example:

```
_caching_policy=30 # Cache refresh time in seconds
_service=urs-stat
_type=builtin
_urs_stat_url_parameters=name=<VQ_Name>&tenant=<Tenant_Name>&aqt=urs
_urs_url=http://<urshost>:<ursport>/urs/call/max/lvq
```

Where: VQ_Name, Tenant_Name, urshost, and ursport match the environment and Callback service's Virtual Queue (VQ). The following screenshot shows the creation and configuration of the **my-urs-stat** service.

> **Important**
>
> The _urs_url option can point to the load balancer in front of the URS that should be configured as part of the GMS provisioning steps in that scenario.

Query EWT Using the urs-stat Service

The following query example shows the resulting response that you get when you call the service:

```
GET http://<gmshost>:<gmsport>/genesys/1/service/<name-of-urs-stat-service>

Response:
{ "wcalls" : 20, "wpos" : 21, "time" : 1467922222, "hit" : 95, "calls" : 20,
 "wt" : 0, "ewt" : 300, "pos" : 21, "aqt" : 300 }
```

> **Important**
>
> • The value of interest here is ewt: the time unit is seconds and can be a float value.

- An empty object will be returned if there is no activity for the VQ.

You can use a single service for multiple VQs by omitting the _urs_stat_url_parameters option from the service and including the value for that option (for example, name of virtual queue, tenant ID, or statistical method) in the HTTP request as follows:

```
http://<gmshost>:<gmsport>/genesys/1/service/<name-of-urs-stat-service>
?name=<one-of-the-callback-VQs>&tenant=<tenant-name>&aqt=urs
```

The URS stat service will append the content of the _urs_stat_url_parameters option and the HTTP request parameters to the URS query. To view additional URS lvq input parameters and output information, open a browser with URS running and run the help method for lvq as follows:

```
http://<urshost>:<ursport>/urs/help/call/lvq
```

The help method is described in the Universal Routing 8.1 Reference Manual, Appendix C, "Supported Methods."

If, for example, you set the following configuration for the <name-of-urs-stat-service> service:

```
_caching_policy=5
_service=urs-stat
_type=builtin
_urs_stat_url_parameters=scale=true&tenant=Environment&aqt=urs
_urs_url=http://<ursloadbalancer>:<ursport>/urs/call/max/lvq
```

You can can use this service for multiple VQs by specifying only the name of a virtual queue in the HTTP request as follows:

```
http://<gmshost>:<gmsport>/genesys/1/service/<name-of-urs-stat-service>
?name=<one-of-the-callback-VQs>
```

Match-interaction

This service will do the following for **ALL** Services:

- It looks through all the services for one that matches the input criteria.
- Only the service id (session id) and data id will be returned to the requester by default. Since 8.5.102, to retrieve user data, you can set the _return_user_data option to `true` in your query.
- The matched service access information will be unreserved (access resources returned).

> ### Important
> The user of this service can use the Storage APIs to retrieve any data that was associated with the matched service.

Create Request Data

These are the service specific parameters that will be supplied on the Create service API.

Parameters	Mandatory	Description
_phone_number	No	This is the phone number of the party that is calling and will be matched against the _phone_number property of the services.
_access_number	Yes	This is the number that the party called and will be matched against the _access_number property associated with the services.
_access_code	No	This is the code assigned to the party that is calling will be matched against the _access_code property assigned to the services.
_return_multiple_matches [since 8.5.102]	No	Set this option to `true` to retrieve an array of matching results in the JSON response in the following format: `{ "matches" : [ { _id...}, { _id...} ] }`
_return_data	No	Set this option to `true` to retrieve

Parameters	Mandatory	Description
[since 8.5.102]		the user data instead of `data_id` only; `false` by default.
_delete_service [since 8.5.102]	No	Set this option to `true` to delete the service; `false` by default. Note that, in any case, GMS releses the `_access_number` and `_access_code` resources.

Create Response Data

Parameters	Mandatory	Description
_id	Yes	Identifier of the matching service.
_data_id	Yes	Identifier of the matching service's data that is in GMS Storage.
user_data [since 8.5.102]	No	User Data attached to service, if you set the `_return_user_data` option to `true`.

Specific Requests

There are no specific requests for this service.

Events

There are no events associated with this service.

Configuration

The following are the configuration options that you can access or modify in the Mobile Engagement UI:

Option	Description
_type = builtin	• For Genesys Mobile Services-based services: builtin • For Orchestration Server-based services: ors
_service = match-interaction	• For Genesys Mobile Services-based services: The name of the matching service.

Option	Description
	• For Orchestration Server-based services: The URL of the service's SCXML application. • Other SCXML Callback services: The URL of the service's SCXML callback application. In that case, you must set _service_type to callback.
_phone_number	Note: Request parameter. Number of the mobile phone where the service request originated.
_access_code	Access code returned as a result of request-access. Note: Request parameter.
_access_number	Number that the customer called. Note: Request parameter.

Request-access

Type: builtin
Updated in: 8.5.109

This service provides a service with access information that has been allocated for it and can then be used to contact the contact center. This service will do the following for any service:

- It will validate that the requesting service is active and running.

- It will acquire the appropriate access information based on the basic allocation algorithm.

  - It can reserve the access information for a configurable period of time

  - Simple random or round-robin allocation

- It supports the following types of access information:

  - Access Number (DNIS) which is to be called by the application

  - Access code which is to be supplied by the customer/application when the contact is being established. This provides an extra level of authentication.

Create Request Data

These are the service-specific parameters that will be supplied on the Create service API.

Parameters	Mandatory	Description
_id	Yes	Identifier of the service which the allocated access information should be associated with.
_provide_code	No	If true, the service should return an access code which will add more security and reliability when trying to correlate the incoming call with the service. The value is a boolean. The default is false.
_phone_number	No	Phone number that is to be associated with the reserved access information.
_resource_group	Yes	Resource group from which an available access number will be taken.
_booking_expiration_timeout	No	Expiration time in seconds used to book the resource. This parameter will override the value in **Resource** options and in configuration options. This

Parameters	Mandatory	Description
		integer value must be between 5 and 1800; if not, the _expiration_time parameter returned in the response is 30.

Create Response Data

Parameters	Mandatory	Description
_id	Yes	This is the identifier of the service which the allocated access information should be associated with.
_access_number	Yes	This is the access number which was allocated for this application. The application should use this number to contact the contact center.
_access_code	No	This is the access code that should be supplied by the application or end user when the call is established to further authentication the application/ user. This will be present when the Create API specifies that it needs an access code (_provide_code = true).
_expiration_time	Yes	This is the amount time that this access information will be locked/ reserved for the service.

Example

```
POST genesys/1/service/request-access
_id=1212-887ad635-22b1-49c5-ae7b-3b271bc10289
_resource_group=DNIS

<Response xmlns="http://10.10.26.36/genesys/1/service/request-access">
    <_access_number>6504664136</_access_number>
    <_expiration_time>29</_expiration_time>
    <_id>1213-2fb1b27e-81b7-4f21-832c-39cf00ac1822</_id>
</Response>
```

Specific Requests

There are no specific requests for this service.

Events

There are no events associated with this service.

Configuration

The following are the configuration options that are defined for this service:

Option	Description
_access_code_length	Length of the access code which can be allocated. Note: Request parameter.
_id	Identifier of the service which the allocated access information should be associated with. Note: Request parameter.
_phone_number	Phone number to associate with the reserved access information. Note: Request parameter.
_provide_code	If set to true, the service returns an access code which will add more security and reliability when trying to correlate the incoming call with the service. Note: Request parameter.
_resource_group = GMS_Resources	Resource group from which an available access number will be taken. Note: Request parameter.

Request-chat

Type: builtin
Updated in: 8.5.109

This service is responsible for receiving the GMS request and providing a URL to start the chat interaction. This is a basic chat service which helps a customer application to contact the call center. It has the following characteristics:

- It supports only customer-initiated chat sessions.

- It stores and maintains the application data within the service.

- It is responsible for routing the the chat interaction to a specified (or configured) interaction endpoint.

- It supports both poll and async (via cometd) mode of message delivery.

Basic Chat API

For further details, refer to the Chat API Version 1 page.

Create a basic chat service

This API allows the application to create basic chat service session and then initiate chat interaction immediately or when user is ready.

> ### Important
>
> If the agent availability needs to be checked before the chat interaction is started, use one of the advanced sessions.

Operation

Method	POST		
URL	/genesys/1/service/request-chat		
Parameter	**Type**	**Mandatory**	**Description**
URI Parameters			
'request-chat'	String	yes	Name of the preconfigured basic chat service
Body: The body will be x-www-form-urlencoded form consisting of different items representing the key/value pairs associated with the request.			
Body Properties: The following are the properties:			

Method	POST

- _verbose - This will allow the application to get all the detail attributes associated with the chat session in the corresponding response.

- ... - Any other business data attributes can also be passed.

Response

HTTP code	200
HTTP message	OK
Body	A chat JSON object for details on the properties of the object. See the section on data structures for more details.
Notes	None

HTTP code	503
HTTP message	Service Unavailable
Body	None
Notes	This is send if the service has not sent a notification to the application that an agent is available.

Example Request:

```
POST http://localhost:8080/genesys/1/service/request-chat HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
_verbose=true
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 10 Jun 2012 07:49:46 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(7.6.0.v20120127)
{
"_chatIxnAPI-CREATE-URL":
"/genesys/1/service/81f0ef4e-99dd-43ea-8366-8d27a2cbd605/ixn/chat",
"_id":"81f0ef4e-99dd-43ea-8366-8d27a2cbd605"
}
```

Configuration

The following are the configuration options that are defined for this service:

_ttl

Section: request-chat
Default Value: 3600
Valid Values: Any positive integer
Changes Take Effect: Immediately

This option is mandatory.

Specifies the default time to live for the chat interaction in seconds.

_chat_endpoint

Section: request-chat
Default Value: gms_builtin
Valid Values: String
Changes Take Effect: Immediately

The endpoint configured on the Chat Server that will be used to submit Chat interactions for this service.

Capacity

Type: **ors**

The Capacity Service enables you to define the number of scheduled callbacks that are allowed for Callback for a given time slot in the week. Then, your Callback service refers to your Capacity service and to your Office Hours service to adjust the agent availability and the number of scheduled callbacks. You can also implement exceptions that allow you to set a specific capacity for a given date.

If your Callback Service needs to define its scheduled callback capacity, you must map the _capacity_service parameter value with the name of the Capacity service that you have created. Depending on the defined capacity and on the defined business hours, the number of scheduled callbacks during certain days or hours will increase or decrease.

> ### Important
>
> Callback services that need fixed capacity levels can continue to use the _max_request_by_time_bucket option. But, if your Callback service includes both _capacity_service and _max_request_by_time_bucket options, then _max_request_by_time_bucket is ignored.

REST API

The Capacity service is similar to Office Hours and is accessible through REST API for external queries.

Refer to the API Reference for further details.

Configuration

Option	Value	Comment
_type	builtin	Mandatory option.
_service	capacity	Mandatory
_capacity_*	JSON-formatted String	Multiple properties that start with a prefix _capacity_ and describe capacity allocation through the course of the week. The JSON structure specifies the day of the week, and capacity for hourly

Option	Value	Comment

		slots during that day. Days of the week are numbered as recommended by ISO-8601 from 1 (Monday) to 7 (Sunday).
		
		`_capacity_1 : {`
		`  1 : { // Monday`
		`    "0900" : 5,`
		`    "1000" : 7,`
		`    "1100" : 10,`
		`    "1200" : 10 }`
		`}`
		`_capacity_2 : {`
		`  2 : { // Tuesday`
		`    "0900" : 3,`
		`    "1000" : 5,`
		`    "1100" : 7,`
		`    "1200" : 7 }`
		`}`

_capacity_add*	JSON-formatted String	Multiple properties that start with the _capacity_add_ prefix and describe the capacity exceptions for additional working days.
		
		The format is similar to the format of _capacity_* properties but instead of a weekday, the full date for the extra day is used to prefix the capacity exception. This date is entered in the format of yyyyMMdd (year, month, day of the month).
		
		`_capacity_add_20160508 : {`
		`  20160508 : { // May 8, 2016`
		`    "0900" : 5,`
		`    "1000" : 7,`
		`    "1100" : 10 }`
		`}`
		`_capacity_add_20161111 : {`
		`  20161111 : { // November 11, 2016`
		`    "0900" : 3,`
		`    "1000" : 5,`
		`    "1100" : 7 }`
		`}`

_timezone		Timezone for your capacity service. For instance, if you configured "EST", or "PST" timezones with the CME, your parameters must use the timezones defined for Java such as "America/Toronto", or "Europe/Paris". See here Wikipedia to get the list of correct timezones.

Option	Value	Comment

Examples of Call Flows for Access Number Allocation

The following call flows show how you can use the following builtin services to implement access number allocation.

No Locking

DNIS only - no match is done here
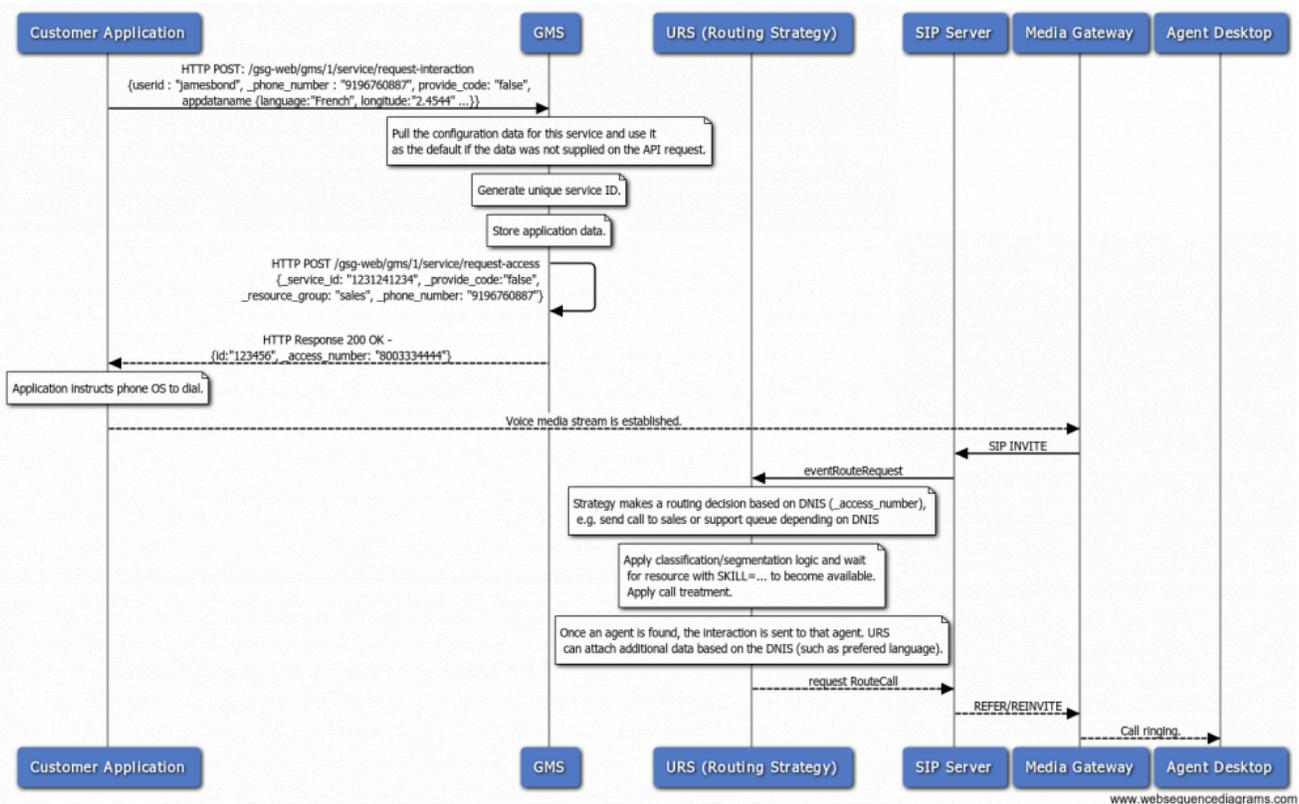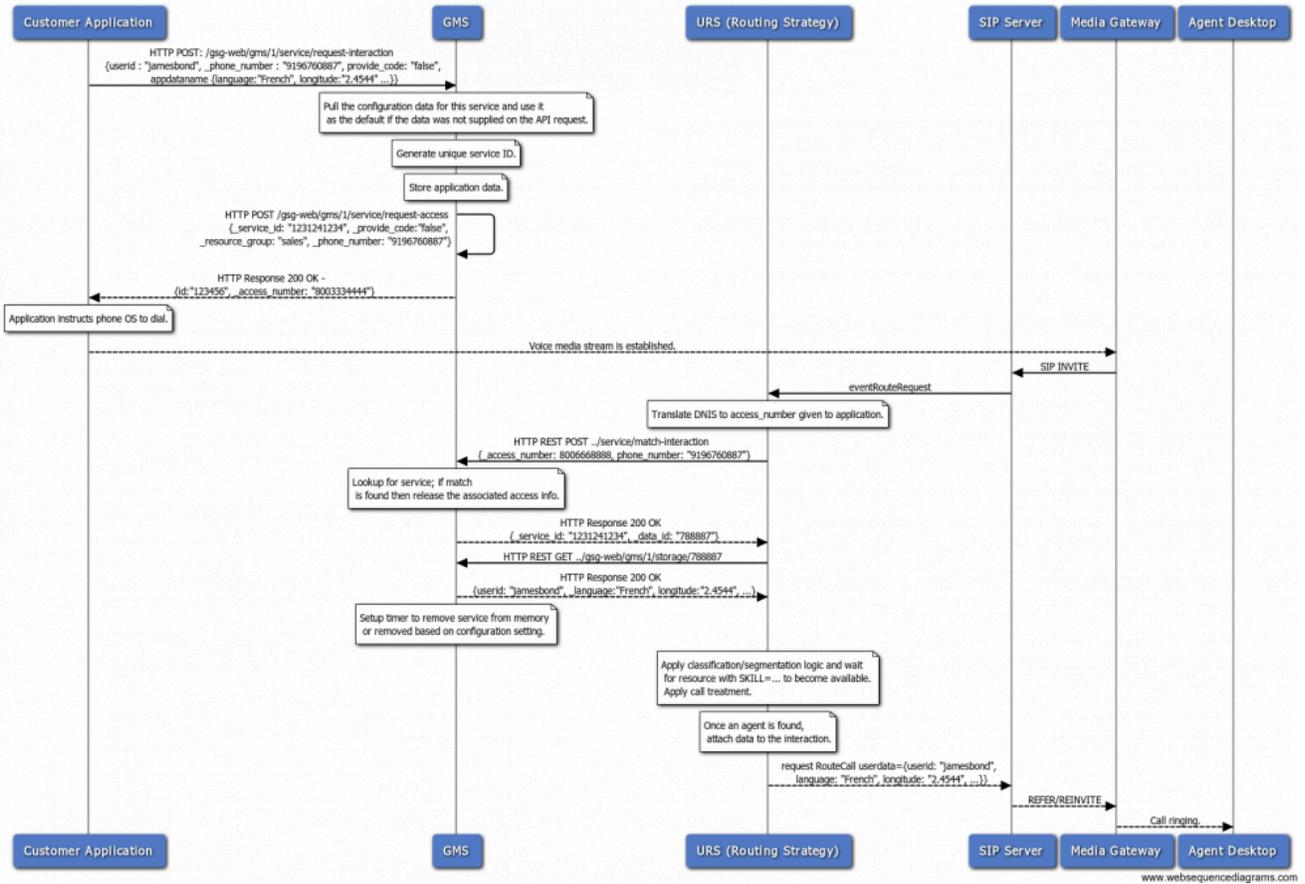
Match DNIS + ANI

Match DNIS + Access Code
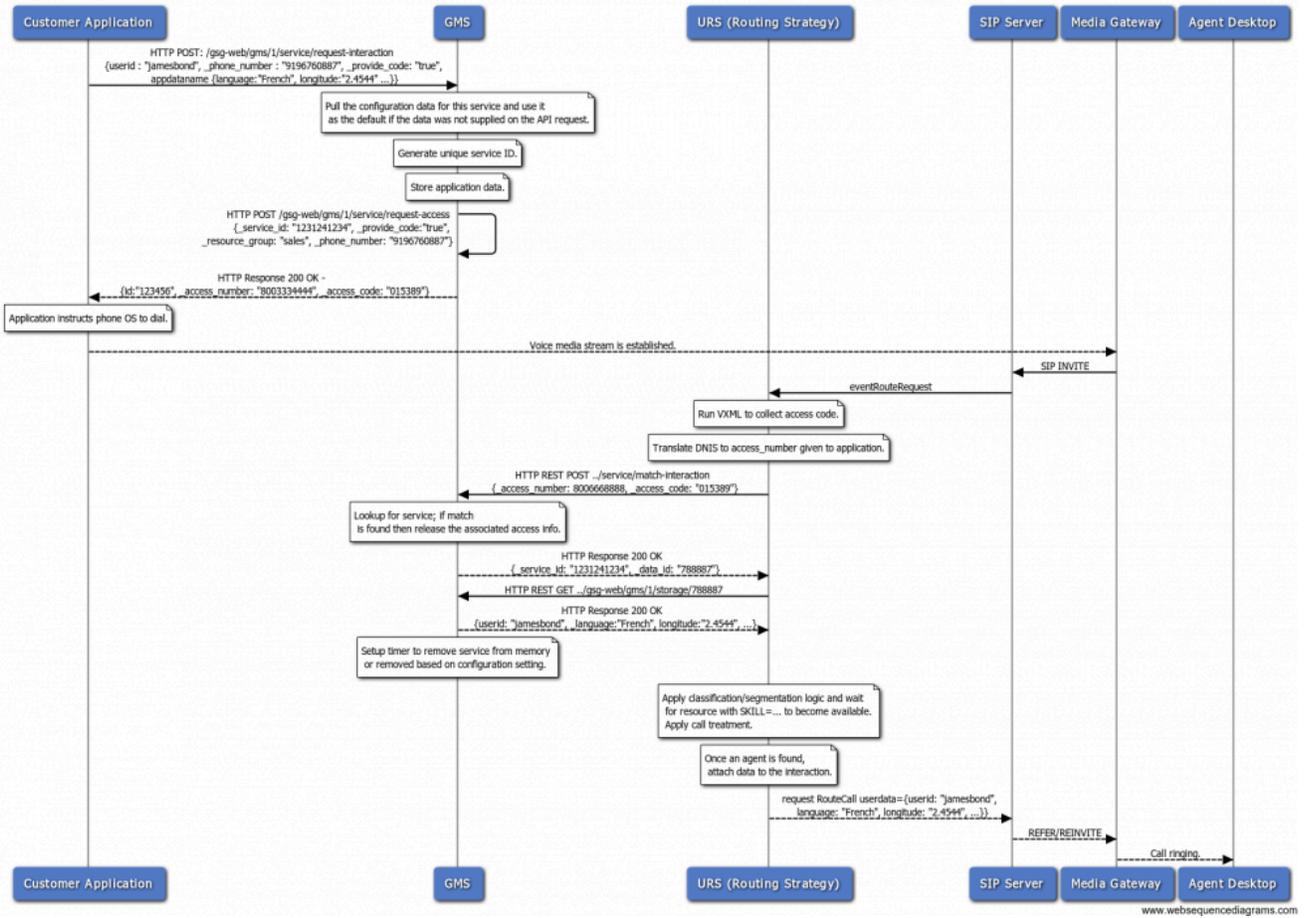
No Match
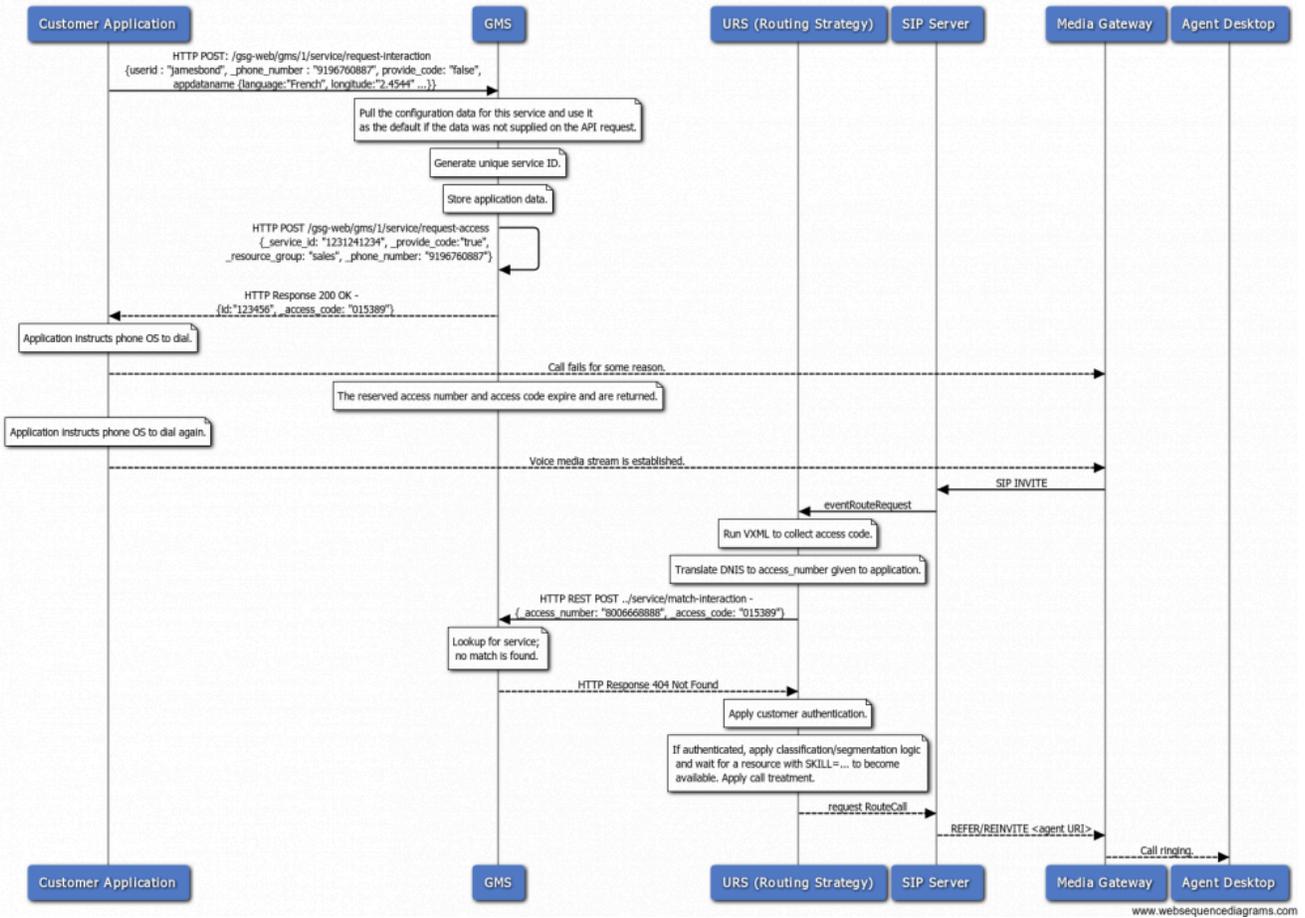
Locking

The only difference between locking and non-locking call flows is the algorithm that determines how access numbers are handed out. In this case, a given access number is only assigned to a single service which allows for a more reliable match.

Match DNIS + ANI

Same as the equivalent no lock case.

Match DNIS + Access Code

Same as the equivalent no lock case.

No Match

Same as the equivalent no lock case.

User Originated Immediate

Type: **ors / inbound voice service**

Scenario

The customer wishes to contact the Call Center immediately. The Callback service provides an access number and an access code (optional) that the customer can dial. Then, when the customer's inbound call is processed and routed to an agent.

This Callback scenario goes through the following stages:

Start Callback

- Callback service: Returns access information immediately to the mobile device.
- Callback service: Waits for the voice call to arrive.
- Next: The Mobile device is expected to dial the access number.

Dial Access Number

- Inbound service: Locates the GMS service associated with the arrived voice call.
- Inbound service: Delegates the call to be processed by the Callback service.
- Callback service: Plays treatment until the target is available.
- Callback service: Reserves target to route call.
- Callback service: Routes the call to the target.
- Callback service terminates.

How to Configure this Scenario

Open the GMS Service Management UI to create the service and set the configuration options.

On the *Services* > *Configured Services* tab, add a Callback service with *User-Originated-Immediate* as the Common Default Configuration (see Configured Services). When you add this service and default configuration, many options are automatically populated with the appropriate default values. Some options, however, will require you to enter your own values. See the sections below for further guidance. See the Reference section below for options' details.

Default Values for Inbound Immediate

These are the default values, which are automatically populated when using the pre-defined User-Originated-Immediate service. Do not change these values:

Option	Description
_wait_for_agent = false	True to wait for an agent to connect. If this option is set to true, the service will wait for the agent to initiate the interaction and to send the notification to the customer. If the option is set to false, the interaction can start right after the creation of the service instance. In voice scenarios, the access information will be returned immediately with the service ID. **This option is mandatory.**
_call_direction = USERORIGINATED	This is a default value, automatically populated when using the predefined User-Terminated scenario. You do not need to change this value. • If this option is set to USERORIGINATED, the customer's device will initiate the call to get connected to the agent. • If this option is set to USERTERMINATED, the agent or the system will initiate the call to contact the customer.
_userterminated_first_connect_party = CUSTOMER	First party to connect when _call_direction is set to USERTERMINATED. Set this option to CUSTOMER to call the customer first; set this option to AGENT to call the agent first. **This option is mandatory.**
_ttl = 86400	Duration (in seconds) for which the service will be kept in storage after the Desired Time is passed (Time To Live). Once expired, the service is removed from the system. For example, if you want the callbacks to be visible in the Service Management UI for one week past the execution time, then you should set 7 days of Time To Live, which means _ttl=604800. **This option is mandatory.**
_type = ors	• For Genesys Mobile Services-based services: builtin

Option	Description
	• For Orchestration Server-based services: ors
_provide_code = false	If true, returns a randomly generated code to be used for the authentication of the user originated (inbound) call. **This option is mandatory.**

See the Reference section below for options' details.

Additional Required Options

You must enter a string value for the following options:

Option	Description
_resource_group="DNIS"	Resource group from which access number is to be allocated. **This option is mandatory.**
_urs_virtual_queue="MyVirtualQueue"	Virtual queue (alias) to which the service request will be added.
_target="MyTarget@StatServer.GA"	Routing target that specifies the agent/queue resource that will process this request. • Starting in 8.5.108.02, you can set multiple targets in this option, limited to 5. • Starting in 8.5.114.09, the limit is increased to 15. ### Single Target For a **single** target, format the string according to the URS target specification: `<Target String>@<StatServer name>.<Target Type>` where `Target Type` is one of the following: • A (Agent) • AP (Agent Place) • GA (Group of Agents)

Option	Description

	• GP (Group of Places)
	
	• GC (Campaign Group)
	
	`<Target String>` can be a skill expression. In that case, `<Target String>` must start with '?:'.
	For example:
	
	• `Billing@StatServer.GA`—Routes to Agent Group "Billing".
	
	• `?:English=20&;Loans=2@StatServer.GA`—Routes to any agent matching the skill expression.
	
	See the Universal Routing Server (URS) documentation for additional information about URS targets.
	
	## Multiple Targets
	
	To set multiple targets, create a JSON-formatted string array of maximum 15 elements as follows:
	
	`[`
	`    {`
	`        "target": "<Target String>@<StatServer name>.<Target Type>",`
	`        "timeout": "<integer>",`
	`        "clear":<true/false>,`
	`        "stat_to_check": "<stat name>",`
	`        "stat_operator": "< or >",`
	`        "stat_value": "1"`
	`    }`
	`]`
	
	• The `timeout` property specifies how long to wait in seconds before switching of targets.
	
	• The `stat_to_check` property can be set to any of the values supported by the Statistics parameter passed to the IRD function `SData(Target, Statistics)`, unless `target` is a skill expression. If `target` is a skill expression, you must choose one of the following values:
	
	  • `RStatAgentsReadyvoice`—agents ready for voice media.
	
	  • `RStatAgentsReady`—agents ready for any media.
	
	  • `RStatAgentsTotal`—agents logged in.
	
	• The `stat_value` property specifies the threshold for the statistic passed in `stat_to_check`. If the condition set by the combination of `stat_to_check`, `stat_operator`, and `stat_value` is met, the current target is skipped, except if it is the last

Option	Description
	target of the list. • If `clear=true`, the target will be overridden when switching to the next target; if `clear=false`, the target will be expanded with the next target. **Important** If you set multiple targets in this option, then `_urs_queued_ttl` should be set to the total queue time across all targets. more…
_urs_prioritization_strategy = WaitForTarget _urs_strategy_update_sub_routine = SetRouteDelay	These options respectively match the name of the URS strategy and subroutine that you imported into IRD. If you changed one of these names, update the corresponding option to reflect the correct name.

See the Reference section below for options' details.

Customization

All of the options in the Voice-User Originated section are applicable. You can use the default values, or you can set your own values.

Option	Description
Section: Voice-User Originated	
_booking_expiration_timeout = 30	Used to book the access number resource for a period of time. The customer needs to make the call within the specified timeout to ensure a successful match. **This option is mandatory.**
_provide_code= false	If true, returns a randomly generated code to be used for the authentication of the user originated (inbound) call. **This option is mandatory.**
_userorig_connect_limit=3	Maximum number of times that the customer can request a connection or a reconnection.

Option	Description
Section: Voice Treatment	
_treatment_find_agent_fail = GMSApplications/<treatmentfile1>	Music file to be played when the service fails to find the agent in the time specified by the Max Time To Wait For Agent on the Call parameter. This parameter accepts a URI as a string or as a JSON-formatted string. See also _treatment_waiting_for_agent. By default, this option has an empty value and Callback will use the <GMS installation>/Resources/SampleTreatments/all_agents_busy.wav file available in the callback template.
_treatment_waiting_for_agent = GMSApplications/<treatmentfile2>	Music file to play when the customer is waiting for an agent. This parameter accepts a URI as a string or as a JSON-formatted string. If you do not set this option, Callback will use the default <GMS installation>/Resources/SampleTreatments/next_customer_rep.wav file of the callback template.

> **Important**
>
> In the Voice Treatment section, the GMSApplications/<treatmentfile> path is applicable if you are using the treatments builtin to the Callback strategy. If you are not using the builtin treatments, enter the path where you have placed your voice treatment files.

Sample Request and Response Sequence

Create inbound immediate service

```
Request URL:http://localhost:8080/genesys/1/service/callback/voice-userorig-immediate
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:44
Content-Type:multipart/form-data; boundary=----WebKitFormBoundarypn9IDb0VLCgiULed
Cookie:JSESSIONID=142p9me7hc3ht635n8mkx3eit; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundarypn9IDb0VLCgiULed--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 06:37:45 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=5lpxopl7oanu50iexlwaqrvg;Path=/genesys
Transfer-Encoding:chunked

Response Body:
{
    "_access_code": "n/a",
    "_access_number": "6504663204",
    "_action": "DialNumber",
    "_dialog_id": "0",
    "_expiration_time": "29",
    "_id": "369-37f2ab38-ec98-4316-a28d-dec01d622ae8",
    "_label": "Connecting ...",
    "_tel_url": "tel:6504663204"
}
```

Sequence Diagram

Click the diagram to access full resolution or right-click and open in another tab to zoom in.

User Originated Delayed

Type: **ors** / **inbound voice service**

Scenario

The customer wishes to contact the Call Center and accepts to wait for an agent. The Callback service notifies the mobile when an agent is ready, then provides an access number and an access code (optional) that the customer can dial. Then, the customer's inbound call is processed and routed to an agent.

This Callback scenario is an inbound voice service that goes through the following stages:

Start Callback

- Callback service: Returns session id to the user.
- Callback service: Waits for an agent to be available.
- Callback service: When an agent is available, notifies the mobile device that agent is available.
- Next: the Mobile device is expected to send connect request to confirm the user's availability.

Connect

- Callback service: Returns access information immediately to the mobile device.
- Callback service: Waits for the voice call to arrive.
- Next: the Mobile device is expected to dial the access number.

Dial Access Number

- Inbound service: Locates the GMS service associated with the arrived voice call.
- Inbound service: Delegates the call to be processed by the Callback service.

- Callback service: Reserves target to route call.

- Callback service: Routes the call to the target.

- Callback service terminates.

How to Configure this Scenario

The sections below list the key options applicable to this scenario. Use the GMS Service Management UI to set the User Originated Delay service and configure options. On the *Services > Configured Services* tab, add a Callback service with *User-Originated-Delayed* as the Common Default Configuration (see Configured Service). When you add this service and default configuration, many options are automatically populated with the appropriate default values. Some options, however, will require you to enter your own values. See the Comments column in the following table for these details.

Default Values for User Originated Delay

These are the default values, which are automatically populated when using the pre-defined User-Originated-Immediate service. You must not change these values.

Option	Description
_wait_for_agent = true	True to wait for an agent to connect. If this option is set to true, the service will wait for the agent to initiate the interaction and to send the notification to the customer. If the option is set to false, the interaction can start right after the creation of the service instance. In voice scenarios, the access information will be returned immediately with the service ID. **This option is mandatory.**
_wait_for_user_confirm = true	True to wait for confirmation of the customer's availability. If this option is set to true, the service sends a push notification to the customer's device to get confirmation that the customer is ready to have a conversation with the agent. This scenario is possible only if the _wait_for_agent option is set to

Option	Description
	true.
_call_direction = USERORIGINATED	This is a default value, automatically populated when using the predefined User-Terminated scenario. You do not need to change this value. • If this option is set to USERORIGINATED, the customer's device will initiate the call to get connected to the agent. • If this option is set to USERTERMINATED, the agent or the system will initiate the call to contact the customer.
_userterminated_first_connect_party = CUSTOMER	First party to connect when _call_direction is set to USERTERMINATED. Set this option to CUSTOMER to call the customer first; set this option to AGENT to call the agent first. **This option is mandatory.**
_ttl = 86400	Duration (in seconds) for which the service will be kept in storage after the Desired Time is passed (Time To Live). Once expired, the service is removed from the system. For example, if you want the callbacks to be visible in the Service Management UI for one week past the execution time, then you should set 7 days of Time To Live, which means _ttl=604800. **This option is mandatory.**
_type = ors	• For Genesys Mobile Services-based services: builtin • For Orchestration Server-based services: ors

Option	Description
_provide_code = false	If true, returns a randomly generated code to be used for the authentication of the user originated (inbound) call. **This option is mandatory.**
_use_debug_push_certificate = false	Use debug certificates for the push notification provider

Additional Required Options

You must enter a string value for the following options:

Option	Description
_resource_group="DNIS"	Resource group from which access number is to be allocated. **This option is mandatory.**
_urs_virtual_queue="MyVirtualQueue"	Virtual queue (alias) to which the service request will be added.
_target="MyTarget@StatServer.GA"	Routing target that specifies the agent/queue resource that will process this request. • Starting in 8.5.108.02, you can set multiple targets in this option, limited to 5. • Starting in 8.5.114.09, the limit is increased to 15.

Option	Description

	Single Target
	
	For a **single** target, format the string according to the URS target specification: `<Target String>@<StatServer name>.<Target Type>` where `Target Type` is one of the following:
	
	• A (Agent)
	• AP (Agent Place)
	• GA (Group of Agents)
	• GP (Group of Places)
	• GC (Campaign Group)
	
	`<Target String>` can be a skill expression. In that case, `<Target String>` must start with '?:'. For example:
	
	• `Billing@StatServer.GA`—Routes to Agent Group "Billing".
	• `?:English=20&;Loans=2@StatServer.GA`—Routes to any agent matching the skill expression.
	
	See the Universal Routing Server (URS) documentation for additional information about URS targets.
	
	Multiple Targets
	
	To set multiple targets, create a JSON-formatted string array of maximum 15 elements as follows:
	
	`[`
	`    {`
	`        "target": "<Target String>@<StatServer name>.<Target Type>",`
	`        "timeout": "<integer>",`
	`        "clear":<true/false>,`
	`        "stat_to_check": "<stat name>",`
	`        "stat_operator": "< or >",`

Option	Description
| | ```
        "stat_value": "1"
    }
]
```<br><br>• The `timeout` property specifies how long to wait in seconds before switching of targets.<br><br>• The stat_to_check property can be set to any of the values supported by the Statistics parameter passed to the IRD function `SData(Target, Statistics)`, unless target is a skill expression. If `target` is a skill expression, you must choose one of the following values:<br><br>  • RStatAgentsReadyvoice—agents ready for voice media.<br>  • RStatAgentsReady—agents ready for any media.<br>  • RStatAgentsTotal—agents logged in.<br><br>• The stat_value property specifies the threshold for the statistic passed in stat_to_check. If the condition set by the combination of stat_to_check, stat_operator, and stat_value is met, the current target is skipped, except if it is the last target of the list.<br><br>• If `clear=true`, the target will be overridden when switching to the next target; if `clear=false`, the target will be expanded with the next target.<br><br>**Important**<br>If you set multiple targets in this option, then \_urs_queued_ttl should be set to the total queue time across all targets.<br><br>more... |
| _urs_prioritization_strategy = `WaitForTarget`

_urs_strategy_update_sub_routine = `SetRouteDelay` | These options respectively match the names of the URS strategy and subroutine that you imported into IRD. If you changed one of these names, update the corresponding option to reflect the correct name. |

Customization

All of the options in the section are applicable. You can use the default values, or you can set your own values.

Option	Description
Voice-User Originated	
_booking_expiration_timeout = 30	Used to book the access number resource for a period of time. The customer needs to make the call within the specified timeout to ensure a successful match. **This option is mandatory.**
_provide_code= false	If true, returns a randomly generated code to be used for the authentication of the user originated (inbound) call. **This option is mandatory.**
_userorig_connect_limit=3	Maximum number of times that the customer can request a connection or a reconnection.
Voice Treatment section	
_treatment_find_agent_fail = GMSApplications/<treatmentfile1>	Music file to be played when the service fails to find the agent in the time specified by the Max Time To Wait For Agent on the Call parameter. This parameter accepts a URI as a string or as a JSON-formatted string. See also _treatment_waiting_for_agent. By default, this option has an empty value and Callback will use the <GMS installation>/Resources/SampleTreatments/all_agents_busy.wav file available in the callback template.

Option	Description
_treatment_waiting_for_agent = GMSApplications/<treatmentfile2>	Music file to play when the customer is waiting for an agent. This parameter accepts a URI as a string or as a JSON-formatted string. If you do not set this option, Callback will use the default `<GMS installation>`/Resources/SampleTreatments/`next_customer_rep.wav` file of the callback template.

Important

In the Voice Treatment section, the `GMSApplications/<treatmentfile>` path is applicable if you are using the treatments builtin to the Callback strategy. If you are not using the builtin treatments, enter the path where you have placed your voice treatment files. This path may or may not point to files on a Genesys Media Server.

Sample Request and Response Sequence

Create inbound delay service

```
Request URL:http://localhost:8080/genesys/1/service/callback/voice-userorig-delay
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:753
Content-Type:multipart/form-data; boundary=----WebKitFormBoundary4KngrF26eEEFKIu5
Cookie:JSESSIONID=5lpxopl7oanu50iexlwaqrvg; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
```

```
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundary4KngrF26eEEFKIu5
Content-Disposition: form-data; name="type"

Gold
------WebKitFormBoundary4KngrF26eEEFKIu5
Content-Disposition: form-data; name="_customer_number"

6504661232
------WebKitFormBoundary4KngrF26eEEFKIu5
Content-Disposition: form-data; name="usr_customer_name"

Bob Markel
------WebKitFormBoundary4KngrF26eEEFKIu5
Content-Disposition: form-data; name="usr_reason"

billing question
------WebKitFormBoundary4KngrF26eEEFKIu5
Content-Disposition: form-data; name="_device_notification_id"

b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
------WebKitFormBoundary4KngrF26eEEFKIu5
Content-Disposition: form-data; name="_device_os"

iOS
------WebKitFormBoundary4KngrF26eEEFKIu5--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 06:51:31 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=fbnacjg4ginj1xlbzkcv5j7wq;Path=/genesys
Transfer-Encoding:chunked

Response Body:
```

```
{
    "_id": "369-90f389cf-ae63-46ec-a3f7-7b76c0522e2f",
    "_text": "We will notify you when agent is available"
}
```

Push notification data:

```
{
    "id": "db508720258611e300006072543ed1ff",
    "message": {
        "_action": "get-dialog-user-confirmation-provide_code-false",
        "_id": "369-90f389cf-ae63-46ec-a3f7-7b76c0522e2f"
    },
    "tag": "service.agentavailable.369-90f389cf-ae63-46ec-a3f7-7b76c0522e2f"
}
```

Connect (user confirmation/request access info)

```
Request URL:http://localhost:8080/genesys/1/service/369-90f389cf-ae63-46ec-a3f7-7b76c0522e2f/connect
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:44
Content-Type:multipart/form-data; boundary=----WebKitFormBoundarylgOFYFBusYz5okcj
Cookie:JSESSIONID=febiiafprxso1n7bokxbrqwqv; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundarylgOFYFBusYz5okcj--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:312
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
```

Date:Tue, 30 Jul 2013 06:54:21 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=1t65gbvy9mrwdwysh123gpb0f;Path=/genesys

Response Body:
{
    "_dialog_id": "0",
    "_label": "Connecting ...",
    "_action": "DialNumber",
    "_tel_url": "tel:6504663205",
    "_access_code": "n/a",
    "_access_number": "6504663205",
    "_expiration_time": "29",
    "_id": "369-90f389cf-ae63-46ec-a3f7-7b76c0522e2f"
}

Sequence Diagram

Click the diagram to access full resolution.

Chat Immediate

Type: ors / chat

Scenario

The customer wishes to chat immediately with an agent of the Call Center. The Callback service provides an access URL that the customer can connect. Then, the customer's chat interaction is processed and routed to an agent.

This Callback scenario is a chat service that goes through the following stages:

Start Callback

- Callback service: Returns the URL to initiate chat interaction, immediately to the mobile device.

- Callback service: Waits for chat interaction to arrive.

- Next: Mobile device is expected to initiate chat interaction.

Initiate Chat

- Inbound Chat service: Locates the GMS service associated with the arrived chat interaction.

- Inbound Chat service: Delegates the chat interaction to be processed by the Callback service.

- Callback service: Reserves target to route chat interaction.

- Callback service: Routes the chat interaction to the target.

- Callback service terminates.

Setting up Acceptance of a Chat Interaction

In order for the Callback service to accept a chat interaction, you must set up the `inbound_chat.scxml` matching the service through Configuration Manager (or Genesys Administrator). The steps detailed below use Configuration Manager.

Note: The `inbound_chat.scxml` file is included in your GMS installation.

Start

1. In Configuration Manager, create an enhanced script object (for example, *GMSInbound*).

2. On the *Annex* tab, configure the *Application/url* option as follows:
     url = http://<gmshost>:<gmsport>/genesys/1/document/service_template/callback/src/inbound_chat.scxml

3. On the *Annex* tab, configure the *ApplicationParms* section by adding the below options:
     app_find_agent_timeout = <timeout in seconds when routing to agent>

     app_selected_agent_group = <agent_group for GMS match fails>

4. Configure an Interaction Queue to use the enhanced routing object that you just created. To do this, on the *Annex* tab, configure the *Orchestration/application* option as follows:
     script:GMSinbound

End

Configuration Options

The sections below list the key options applicable to this scenario.

Use the GMS Service Management UI to set the configuration options. On the *Services > Configured Services* tab, add a Callback service with *Chat-Immediate* as the Common Default Configuration (see Configured Services). When you add this service and default configuration, many options are automatically populated with the appropriate default values. Some options, however, will require you to enter your own values.

Predefined Values

These are the default values, which are automatically populated when using the pre-defined User-Originated-Immediate service. You must not change these values.

Option	Description
_media_type=chat	Media type of the interaction that the service is expected to handle. This option enables URS to select an agent who has the appropriate media capabilities. This is a default value, automatically populated when using the predefined User-Terminated scenario. You do not need to change this value. **This option is mandatory.**
_wait_for_agent = false	True to wait for an agent to connect. If this option is set to true, the service will wait for the agent to initiate the interaction and to send the notification to the customer. If the option is set to false, the interaction can start right after the creation of the service instance. In voice scenarios, the access information will be returned immediately with the service ID. **This option is mandatory.**
_wait_for_user_confirm = false	True to wait for confirmation of the customer's availability. If this option is set to true, the service sends a push notification to the customer's device to get confirmation that the customer is ready to have a conversation with the agent. This scenario is possible only if the _wait_for_agent option is set to true.
_ttl = 86400	Duration (in seconds) for which the service will be kept in storage after the Desired Time is passed (Time To Live). Once expired, the service is removed from the system. For example, if you want the callbacks to be visible in the Service Management UI for one week past the execution time, then you should

Option	Description
	set 7 days of Time To Live, which means _ttl=604800. **This option is mandatory.**
_type = ors	- For Genesys Mobile Services-based services: builtin - For Orchestration Server-based services: ors
_use_debug_push_certificate = false	Use debug certificates for the push notification provider

Additional Required Options

You must enter a string value for the following options:

Option	Description
_target	Routing target that specifies the agent/queue resource that will process this request. - Starting in 8.5.108.02, you can set multiple targets in this option, limited to 5. - Starting in 8.5.114.09, the limit is increased to 15. ### Single Target For a **single** target, format the string according to the URS target specification: `<Target String>@<StatServer name>.<Target Type>` where `Target Type` is one of the following:

Option	Description

	- A (Agent) - AP (Agent Place) - GA (Group of Agents) - GP (Group of Places) - GC (Campaign Group) `<Target String>` can be a skill expression. In that case, `<Target String>` must start with '?:'. For example:
	
	    `Billing@StatServer.GA`—Routes to Agent Group "Billing". `?:English=20&;Loans=2@StatServer.GA`—Routes to any agent matching the skill expression. See the Universal Routing Server (URS) documentation for additional information about URS targets.
	
	## Multiple Targets
	
	To set multiple targets, create a JSON-formatted string array of maximum 15 elements as follows:
	
	<pre>[
	    {
	        "target": "<Target String>@<StatServer name>.<Target Type>",
	        "timeout": "<integer>",
	        "clear":<true/false>,
	        "stat_to_check": "<stat name>",
	        "stat_operator": "< or >",
	        "stat_value": "1"
	    }
	]</pre>
	- The `timeout` property specifies how long to wait in seconds before

Option	Description

	switching of targets.
	
	• The stat_to_check property can be set to any of the values supported by the Statistics parameter passed to the IRD function SData(Target, Statistics), unless target is a skill expression. If target is a skill expression, you must choose one of the following values:
	
	  • RStatAgentsReadyvoice—agents ready for voice media.
	  • RStatAgentsReady—agents ready for any media.
	  • RStatAgentsTotal—agents logged in.
	
	• The stat_value property specifies the threshold for the statistic passed in stat_to_check. If the condition set by the combination of stat_to_check, stat_operator, and stat_value is met, the current target is skipped, except if it is the last target of the list.
	
	• If clear=true, the target will be overridden when switching to the next target; if clear=false, the target will be expanded with the next target.
	
	Important
	If you set multiple targets in this option, then _urs_queued_ttl should be set to the total queue time across all targets.
	
	more...

_urs_virtual_queue	Virtual queue (alias) to which the service request will be added.
_urs_prioritization_strategy = WaitForTarget _urs_strategy_update_sub_routine = SetRouteDelay	By default, these options respectively match the names of the URS strategy and subroutine that you imported into IRD. If you changed one of these names, update the corresponding option to reflect the correct name.

Sample Request and Response Sequence

When your chat immediate service is configured, you can use this service to retrieve your new service ID associated with your chat session. Then, you will be able to use this ID in the Chat API v1.

Create chat immediate service

```
Request URL:http://localhost:8080/genesys/1/service/callback/chat-immediate
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:660
Content-Type:multipart/form-data; boundary=----WebKitFormBoundaryYbCe8VCHJCMGeynE
Cookie:JSESSIONID=1bob1qjgrmiy589bpyvtmrhow; BAYEUX_BROWSER=f3d8-3vwgdclvyroghjsaf87i1c60
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundaryYbCe8VCHJCMGeynE
Content-Disposition: form-data; name="_customer_number"

6502388511
------WebKitFormBoundaryYbCe8VCHJCMGeynE
Content-Disposition: form-data; name="usr_customer_name"

Bob Markel
------WebKitFormBoundaryYbCe8VCHJCMGeynE
Content-Disposition: form-data; name="usr_reason"

billing question
------WebKitFormBoundaryYbCe8VCHJCMGeynE
Content-Disposition: form-data; name="_device_notification_id"
```

b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
------WebKitFormBoundaryYbCe8VCHJCMGeynE
Content-Disposition: form-data; name="_device_os"

comet
------WebKitFormBoundaryYbCe8VCHJCMGeynE--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 13 Aug 2013 22:15:31 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=e6yzey0kup8v8l7l8lurnhzs;Path=/genesys
Transfer-Encoding:chunked

Response Body:
```
{
    "_dialog_id": "1",
    "_action": "StartChat",
    "_label": "Start Chat",
    "_start_chat_url": "http://localhost:8080/genesys/1/service/369-01d32409-b4bb-4884-b266-02d8216fe5e3/ixn/chat",
    "_comet_url": "http://localhost:8080/genesys/cometd",
    "_user_header": "b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673",
    "_id_to_jump_before": "exit://",
    "_chat_parameters": {
        "subject": "None"
    },
    "_id": "369-01d32409-b4bb-4884-b266-02d8216fe5e3"
}
```

Sequence Diagram

Click the diagram to access full resolution.

Chat Immediate - High Level

| CustomerApp | GMS | ORS | URS | TLIB | AgentDesktop | ChatServer |

[1] Request new Callback

Callback API: Start-Callback

[2] Create ORS (Callback) Session

ORS API: HTTP /session/start

[3] ORS Session Id

Wait for start event from GMS

[4] Event Start Callback

ORS API: HTTP <sessionid>/request/start

[5] Response to start event

Action: StartChat

[6] Action: StartChat

[7] Place new virtual interaction in VQ

Start WaitForTarget Strategy

VirtualIxn status: NOT ROUTABLE

[8] URS ConnId

Callback Status: QUEUED

loop [until virtual ixn deletion]

[9] Heartbeat request (keep virtual ixn)

[10] send userevent with reporting (initial) metrics

Wait for chat interaction

[11] Start Chat interaction

Callback API: HTTP <gmsserviceid>/ixn/chat

[12] Create Chat Interaction

interaction.present

[13] Attach URS udata and GMS User Data to Chat ixn

[14] Queue chat ixn to a dummy reporting vq

Real Interaction Status: CREATED

Callback Status: PROCESSING

[15] Change virtual interaction to routable

VirtualIxn Status: ROUTABLE

Wait for Selected Agent info from URS

[16] Selected Agent information

[17] Route chat ixn to agent

[18] OK

Real Interaction Status: ROUTED (Agent Customer Connected)

VirtualIxn Status: DELETED

[19] Wait for chat ixn to be delete

Chat Delayed

Type: ors / chat

Scenario

The customer wishes to chat with an agent of the Call Center. The Callback service provides an access URL that the customer can connect and waits for an agent. Then, the Callback Service notifies the customer's mobile and the customer's chat interaction is processed and routed to an agent.

This Callback scenario is a chat service that goes through the following stages:

Start Callback

- Callback service: Returns session id to the user.
- Callback service: Waits for an agent to be available.
- Callback service: When an agent is available, notifies mobile device that agent is available.
- Next: Mobile device is expected to send connect request to confirm the user's availability.

Connect

- Callback service: Returns URL to initiate chat interaction, immediately to the mobile device.
- Callback service: Waits for chat interaction to arrive.
- Next: Mobile device is expected to initiate chat interaction.

Initiate Chat

- Inbound Chat service: Locates the GMS service associated with the arrived chat interaction.
- Inbound Chat service: Delegates the chat interaction to be processed by the Callback service.
- Callback service: Reserves target to route chat interaction.

- Callback service: Routes the chat interaction to the target.

- Callback service terminates.

Setting up Acceptance of a Chat Interaction

In order for the Callback service to accept a chat interaction, you must set up the inbound_chat.scxml matching service through Configuration Manager (or Genesys Administrator). The steps shown below use Configuration Manager.

> ### Important
> The inbound_chat.scxml file is included in your GMS installation.

Start

1. In Configuration Manager, create an enhanced script object (for example, *GMSInbound*).

2. On the *Annex* tab, configure the *Application/url* option as follows:
       url = http://<gmshost>:<gmsport>/genesys/1/document/service_template/callback/src/inbound_chat.scxml

3. On the *Annex* tab, configure the *ApplicationParms* section by adding the below options:
       app_find_agent_timeout = <timeout in seconds when routing to agent>

       app_selected_agent_group = <agent_group for GMS match fails>

4. Configure an Interaction Queue to use the enhanced routing object that you just created. To do this, on the *Annex* tab, configure the *Orchestration/ application* option as follows:
       script:GMSinbound

End

Configuration Options

The sections below list the key options applicable to this scenario.

Use the GMS Service Management UI to set the configuration options. On the *Services > Configured Services* tab, add a Callback service with *Chat-Immediate* as the Common Default Configuration (see Configured Services). When you add this service and default configuration, many options are automatically populated with the appropriate default values. Some options, however, will require you to enter your own values. Predefined Values[edit]

These are the default values, which are automatically populated when using the pre-defined *Chat-Delayed* service. You must not change these values.

Option	Description
_media_type=chat	Media type of the interaction that the service is expected to handle. This option enables URS to select an agent who has the appropriate media capabilities. This is a default value, automatically populated when using the predefined User-Terminated scenario. You do not need to change this value. **This option is mandatory.**
_wait_for_agent = true	True to wait for an agent to connect. If this option is set to true, the service will wait for the agent to initiate the interaction and to send the notification to the customer. If the option is set to false, the interaction can start right after the creation of the service instance. In voice scenarios, the access information will be returned immediately with the service ID. **This option is mandatory.**
_wait_for_user_confirm = true	True to wait for confirmation of the customer's availability. If this option is set to true, the service sends a push notification to the customer's device to get confirmation that the customer is ready to have a conversation with the agent. This scenario is possible only if the _wait_for_agent option is set to true.

Option	Description
_ttl = 86400	Duration (in seconds) for which the service will be kept in storage after the Desired Time is passed (Time To Live). Once expired, the service is removed from the system. For example, if you want the callbacks to be visible in the Service Management UI for one week past the execution time, then you should set 7 days of Time To Live, which means _ttl=604800. **This option is mandatory.**
_type = ors	• For Genesys Mobile Services-based services: builtin • For Orchestration Server-based services: ors
_use_debug_push_certificate = false	Use debug certificates for the push notification provider

Additional Required Options

You must enter a string value for the following options:

Option	Description
_target	Routing target that specifies the agent/queue resource that will process this request. • Starting in 8.5.108.02, you can set multiple targets in this option, limited to 5.

Option	Description

	• Starting in 8.5.114.09, the limit is increased to 15.
	
	## Single Target
	
	For a **single** target, format the string according to the URS target specification: `<Target String>@<StatServer name>.<Target Type>` where `Target Type` is one of the following:
	
	• A (Agent)
	• AP (Agent Place)
	• GA (Group of Agents)
	• GP (Group of Places)
	• GC (Campaign Group)
	
	`<Target String>` can be a skill expression. In that case, `<Target String>` must start with '?:'. For example:
	
	• `Billing@StatServer.GA`—Routes to Agent Group "Billing".
	• `?:English=20&;Loans=2@StatServer.GA`—Routes to any agent matching the skill expression.
	
	See the Universal Routing Server (URS) documentation for additional information about URS targets.
	
	## Multiple Targets
	
	To set multiple targets, create a JSON-formatted string array of maximum 15 elements as follows:
	
	`[`
	`    {`
	`        "target": "<Target String>@<StatServer name>.<Target Type>",`
	`        "timeout": "<integer>",`
	`        "clear":<true/false>,`

Option	Description
| | ```
        "stat_to_check": "<stat name>",
        "stat_operator": "< or >",
        "stat_value": "1"
    }
]
```<br><br>• The `timeout` property specifies how long to wait in seconds before switching of targets.<br><br>• The `stat_to_check` property can be set to any of the values supported by the Statistics parameter passed to the IRD function `SData(Target, Statistics)`, unless target is a skill expression. If `target` is a skill expression, you must choose one of the following values:<br><br>   • RStatAgentsReadyvoice—agents ready for voice media.<br><br>   • RStatAgentsReady—agents ready for any media.<br><br>   • RStatAgentsTotal—agents logged in.<br><br>• The `stat_value` property specifies the threshold for the statistic passed in `stat_to_check`. If the condition set by the combination of `stat_to_check, stat_operator,` and `stat_value` is met, the current target is skipped, except if it is the last target of the list.<br><br>• If `clear=true`, the target will be overridden when switching to the next target; if `clear=false`, the target will be expanded with the next target.<br><br>**Important**<br>If you set multiple targets in this option, then `_urs_queued_ttl` should be set to the total queue time across all targets.<br><br>more... |
| _urs_virtual_queue | Virtual queue (alias) to which the service request will be added. |

Option	Description
_urs_prioritization_strategy = WaitForTarget _urs_strategy_update_sub_routine = SetRouteDelay	By default, these options respectively match the names of the URS strategy and subroutine that you imported into IRD. If you changed one of these names, update the corresponding option to reflect the correct name.

Sample Request and Response Sequence

Create chat (delay) service

```
Request URL:http://localhost:8080/genesys/1/service/callback/chat-delay
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:660
Content-Type:multipart/form-data; boundary=----WebKitFormBoundary0WnE36LruxJ4S5nu
Cookie:JSESSIONID=mjjvtphwb8lpce7io23ggxcu; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundary0WnE36LruxJ4S5nu
Content-Disposition: form-data; name="_customer_number"

4082652649
------WebKitFormBoundary0WnE36LruxJ4S5nu
Content-Disposition: form-data; name="usr_customer_name"
```

```
Bob Markel
------WebKitFormBoundary0WnE36LruxJ4S5nu
Content-Disposition: form-data; name="usr_reason"

billing question
------WebKitFormBoundary0WnE36LruxJ4S5nu
Content-Disposition: form-data; name="_device_notification_id"

b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
------WebKitFormBoundary0WnE36LruxJ4S5nu
Content-Disposition: form-data; name="_device_os"

comet
------WebKitFormBoundary0WnE36LruxJ4S5nu--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 07:07:35 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=kwe77jz60uum1u16urvv8vubd;Path=/genesys
Transfer-Encoding:chunked

Response Body:
{
    "_id": "369-166652d2-aed6-443c-9781-6bdff370f9a9",
    "_text": "We will notify you when agent is available"
}

Push Notification Data:
{
    "id": "01afcd60258a11e300006072543ed1ff",
    "message": {
        "_action": "get-dialog-user-confirmation-provide_code-false",
        "_id": "369-166652d2-aed6-443c-9781-6bdff370f9a9"
    },
    "tag": "service.agentavailable.369-166652d2-aed6-443c-9781-6bdff370f9a9"
}
```

Connect (user confirmation)

Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/connect
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:44
Content-Type:multipart/form-data; boundary=----WebKitFormBoundaryNY84ld7wm7oHB9fp
Cookie:JSESSIONID=1b81btxjbrb1wybz5a93i24io; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundaryNY84ld7wm7oHB9fp--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:26
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 07:04:35 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=mjjvtphwb8lpce7io23ggxcu;Path=/genesys

Response Body:
```
{
    "_dialog_id": "1",
    "_action": "StartChat",
    "_label": "Start Chat",
    "_start_chat_url": "http://localhost:8080/genesys/1/service/369-8cea2901-1eba-4f5a-8c76-edf83dd26480/ixn/chat",
    "_comet_url": "http://localhost:8080/genesys/cometd",
    "_user_header": "b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673",
    "_id_to_jump_before": "exit://",
    "_chat_parameters": {
        "subject": "None"
```

```
    },
    "_id": "369-166652d2-aed6-443c-9781-6bdff370f9a9"
}
```

Check estimated wait time (EWT) and position in the URS queue (check-queue-position)

Mobile is expected to use this API to poll for current ewt and position.

```
Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/check-queue-position
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:44
Content-Type:multipart/form-data; boundary=----WebKitFormBoundary2gBrAJX9qPSafKwk
Cookie:JSESSIONID=61voqouznyas1vrp9fjljmbwr; BAYEUX_BROWSER=f3d8-3vwgdclvyroghjsaf87i1c60
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundary2gBrAJX9qPSafKwk--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:123
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Fri, 02 Aug 2013 23:07:59 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=1co1o30bhrm6719u8lfv9kvpei;Path=/genesys

Response Body (when eta is NOT provided by URS):
{
    "_position": 2,
    "_eta": "n/a",
```

```
    "_total_waiting": 2,
    "_agent_ready_threshold_passed": true,
    "_agent_ready_threshold_passed_reason": "eta n/a"
}
Response Body 2 (when eta is provided by URS):
{
    "_position": 2,
    "_eta": 30,
    "_total_waiting": 2,
    "_agent_ready_threshold_passed": true,
    "_agent_ready_threshold_passed_reason": "eta  <= 35 and position <=5 "
}
```

Create Chat Interaction

```
Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/ixn/chat
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:651
Content-Type:multipart/form-data; boundary=----WebKitFormBoundaryteXJ8ZpAGGDTAMFN
Cookie:JSESSIONID=kwe77jz60uum1u16urvv8vubd; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN
Content-Disposition: form-data; name="notify_by"

comet
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN
Content-Disposition: form-data; name="firstName"

John
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN
```

```
Content-Disposition: form-data; name="lastName"

Harry
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN
Content-Disposition: form-data; name="_verbose"

false
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN
Content-Disposition: form-data; name="subject"

testing
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN
Content-Disposition: form-data; name="email"

j.h@gmail.com
------WebKitFormBoundaryteXJ8ZpAGGDTAMFN--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:77
Content-Type:application/json;charset=UTF-8
Content-Type:text/plain;charset=ISO-8859-1
Date:Tue, 30 Jul 2013 07:08:53 GMT
Pragma:no-cache

Response Body:
{
    "_id": "369-166652d2-aed6-443c-9781-6bdff370f9a9",
    "comet_channel": "/_genesys"
}
```

Send notification from agent desktop to mobile (internal-notification, previously poke)

```
Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/internal-notification
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
```

```
Content-Length:320
Content-Type:multipart/form-data; boundary=----WebKitFormBoundary8rbhQqxP5LoJ61i1
Cookie:JSESSIONID=kwe77jz60uum1u16urvv8vubd; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundary8rbhQqxP5LoJ61i1
Content-Disposition: form-data; name="_display_message"

Message to be displayed to user
------WebKitFormBoundary8rbhQqxP5LoJ61i1
Content-Disposition: form-data; name="_application_message"

MESSAGE_FOR_MOBILE_APP_INTERNAL_USE
------WebKitFormBoundary8rbhQqxP5LoJ61i1--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:17
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 07:10:11 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=z1lrha8utmxw1xslu08z473ty;Path=/genesys

Response Body:
{
    "message": "Notification Sent"
}
```

Retrieve any notifications from agent, when notifications are not enabled (retrieve-notifications)

This is necessary only in case of a poll scenario.

```
Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/retrieve-notifications
Request Method:POST
Status Code:200 OK
```

Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:44
Content-Type:multipart/form-data; boundary=----WebKitFormBoundary3I4BzfVbhmlJvPRC
Cookie:JSESSIONID=z1lrha8utmxw1xslu08z473ty; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundary3I4BzfVbhmlJvPRC--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:2
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 07:10:58 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=18gvvxkjquop1x8er65g7ferp;Path=/genesys

Response Body (with push enabled):
[]

Note: Since CometD push was enabled the message was already sent to the device. If push was not configured then the response body would
have contained the message (below)
Push: {
  "_internal_message": "MESSAGE_FOR_MOBILE_APP_INTERNAL_USE",
  "_id": "369-166652d2-aed6-443c-9781-6bdff370f9a9",
  "_display_message": "Message to be displayed to user"
}

Response Body (with push not configured):
[{
  "_internal_message": "MESSAGE_FOR_MOBILE_APP_INTERNAL_USE",
  "_id": "369-166652d2-aed6-443c-9781-6bdff370f9a9",
  "_display_message": "Message to be displayed to user"

```
}]
```

Send notification from mobile to agent

```
Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/agent-notification
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:324
Content-Type:multipart/form-data; boundary=----WebKitFormBoundaryyN11ULST36Tmhzro
Cookie:JSESSIONID=18gvvxkjquop1x8er65g7ferp; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundaryyN11ULST36Tmhzro
Content-Disposition: form-data; name="_display_message"

Message to be displayed to agent
------WebKitFormBoundaryyN11ULST36Tmhzro
Content-Disposition: form-data; name="_application_message"

MESSAGE_FOR_AGENT_DESKTOP_INTERNAL_USE
------WebKitFormBoundaryyN11ULST36Tmhzro--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:17
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 07:15:04 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=1cfd36s0cn6213lqge05h7y6l;Path=/genesys
```

```
Response Body:
{
    "message": "Notification Sent"
}
```

Retrieve notifications from Mobile to Agent (retrieve-agent-notifications)

Agent desktop is expected to poll for messages since push notifications via cometd to agent desktop are not supported.

```
Request URL:http://localhost:8080/genesys/1/service/369-166652d2-aed6-443c-9781-6bdff370f9a9/retrieve-agent-notifications
Request Method:POST
Status Code:200 OK
Request Headersview source
Accept:*/*
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Connection:keep-alive
Content-Length:44
Content-Type:multipart/form-data; boundary=----WebKitFormBoundaryIimjtpB6hZzFZRQ3
Cookie:JSESSIONID=1cfd36s0cn6213lqge05h7y6l; BAYEUX_BROWSER=86721orubxagcqhw0hj14cpyaqk2
gms_user:b16416334828b1d26ef14f329628b55b5a8c631d8928a371a5584722dd7fb673
Host:localhost:8080
Origin:http://localhost:8080
Referer:http://localhost:8080/gmstester/chat.html
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36
Request Payload
------WebKitFormBoundaryIimjtpB6hZzFZRQ3--
Response Headersview source
Cache-Control:no-cache
Cache-Control:no-store
Content-Length:242
Content-Type:application/json;charset=UTF-8
Content-Type:application/json;charset=UTF-8
Date:Tue, 30 Jul 2013 07:16:17 GMT
Expires:Thu, 01 Jan 1970 00:00:00 GMT
Pragma:no-cache
Set-Cookie:JSESSIONID=1x5o3d8bkgczy1pxjb3dac0mmp;Path=/genesys

Response Body:
[
    {
```

```
        "message": "{\"_id\":\"369-166652d2-aed6-443c-9781-6bdff370f9a9\",
\"_internal_message\":\"MESSAGE_FOR_AGENT_DESKTOP_INTERNAL_USE\", \"_display_message\":\"Message to be displayed to agent\"}",
        "tag": "agent.message.369-166652d2-aed6-443c-9781-6bdff370f9a9"
    }
]
```

Sequence Diagram

Click the diagram to access full resolution.

Chat Delayed - High Level

| CustomerApp | GMS | ORS | URS | TLIB | AgentDesktop | ChatServer |

[1] Request new Callback

Callback API: Start-Callback

[2] Create ORS (Callback) Session

ORS API: HTTP /session/start

[3] ORS Session Id

Wait for start event from GMS

[4] Event Start Callback

ORS API: HTTP <sessionid>/request/start

[5] Response to start event

Message: We will notify you when agent is available

[6] Message: We will notify you when agent is available

[7] Place new virtual interaction in VQ

Start WaitForTarget Strategy

VirtualIxn status: NOT ROUTABLE

[8] URS ConnId

Callback Status: QUEUED

loop [until virtual ixn deletion]

[9] Heartbeat request (keep virtual ixn)

[10] send userevent with reporting (initial) metrics

Wait for Agent available notification
Key differentiator in delay scenario
options _wait_for_agent=true

alt [notifications enabled]

  loop [for configured confirmation attempts with retry delay interval]

[11] Agent available notification

[12] Push notification

Confirmation options: Connect/Snooze/Cancel

  alt [confirmation response: Cancel]

[13] Callback status update

COMLETED.CANCELLED

[14] service.exit

CALLBACK EXIT

[alt confirmation response: Snooze]

[15] Wait for snooze duration and retry confirm loop

[16] Agent will be available, dial now

confirmation response: Connect

[17] connect request

Callback API: HTTP <gmsserviceid>/connect

[18] Event Connect Callback

ORS API: HTTP <sessionid>/request/connect

Wait for chat interaction

[19] Start Chat interacion

Callback API: HTTP <gmsserviceid>/ixn/chat

[20] Create Chat Interaction

interaction.present

[21] Attach URS udata and GMS User Data to chat ixn

[22] Queue chat ixn to a dummy reporting vq

Real Interaction Status: CREATED

Callback Status: PROCESSING

[23] Change virtual interaction to routable

VirtualIxn Status: ROUTABLE

Wait for Selected Agent info from URS

[24] Selected Agent information

[25] Route real interaction to agent

Sample

This sample is a Javascript Web interface, available through the Service Management UI.

- This sample illustrates how to implement a Desktop/Mobile browser web application that communicates with GMS and performs supported contact scenarios. It is primarily meant to be used by developers as a reference to build a Javascript-based web application with GMS.

- To enable the sample in the Service Management UI, configure enable-sample = `true` in your GMS configuration.

> ### Important
> You can also use this sample to test your GME deployment.

Access the Web demo of the Sample

Make sure that GMS is started. By default, the **samples** template is loaded and a **samples** service should be available in the list of **Services**.

To access the sample, start the Service Management UI and navigate through **Services and Tools > Tools > Sample**.

You can try a list of scenarios (1) by selecting a scenario, (2) click Connect to submit your query to GMS.

The following screens are available by clicking the corresponding tabs.

- `GMS` - The application home screen showing which GMS scenario can be executed.

- `Log` - Displays log messages related to client-server communication and application debug messages.

- `Queue` - For delay scenarios, checks the status of the interaction in the queue (when a request has been placed and is waiting for an agent).

- `Settings` - Application settings can be made on this screen.

Configure the Sample

[+] See the list of configuration steps

Resource Group—Add Access Number

Why:

GMS provides this access number to the user, and the user dials into this access number.

How:

GMS Service Management UI

Procedure:

1. Go to the GMS Service Management UI > Tools > Resources.
2. Add the access number to the DNIS group.

GMS Service—Create Service request-interaction

Why:

This service is responsible for receiving the GMS request and providing an access number to the user.

How:

GMS Service Management UI

Procedure:

1. Go to the GMS Service Management UI > Services > Configured Services.
2. Click Add Service.
3. Set Configure Service = request-interaction.
4. Set Service Name = request-interaction.
5. Click Save.

GMS Service—Create Service match-interaction

Why:

This service helps to match a voice call with an existing GMS service

responsible for providing the access number.

How:

GMS Service Management UI

Procedure:

1. Go to the GMS Service Management UI > Services > Configured Services.
2. Click Add Service.
3. Set Configure Service = match-interaction.
4. Set Service Name = match-interaction.
5. Click Save.

GMS Service—Create Service request-access

Why:

This service lets you:

- Create a new access to a service
- Allocate a new DN in the resource group

How:

GMS Service Management UI

Procedure:

1. Go to the **GMS Service Management UI > Services > Configured Services**.
2. Click Add Service.
3. Set Configure Service = request-access.
4. Set Service Name = request-access.
5. Click Save.

Inbound SCXML Service—Voice

Why:

The inbound service matches the voice call with an existing GMS service. If a

matching service is found, the GMS user data is attached to the interaction, and the call is routed to the agent.

How:

- Configuration Manager > Switches > SIP_Switch
- Configuration Manager > Scripts

Procedure:

1. Create a route point associated with the access number configured in the procedure Resource Group Add Access Number.

2. Set Annex > Orchestration section > `application` = `script:GMSInbound.Voice.GMSMatchBuiltin`.

3. Create an enhanced routing script `GMSInbound.Voice.GMSMatchBuiltin`.

4. Set Annex > Application section > `url = http://<gmshost:gmsport>/genesys/1/document/ service_template/callback/src-gen/IPD_Voice_GMSMatch.scxml`.

5. In Annex > `ApplicationParms`, set:
   `app_find_agent_timeout = 30`

   `app_match_gms_builtin = true`.

   `app_match_target = <target>` (Example: `Customer_Service@stat_server.GA`).

   `app_no_match_target = <target>` (Example: `All_Standard_Agents@stat_server.GA`).

   `app_require_access_code = false`.

   `app_require_ani = true`.

   `app_treatment_waiting_for_agent = <blank>` (A blank value will force the service to use a packaged music file.).

6. Make sure that MSML capabilities are configured and working to play treatments. This step is required because this service includes play treatments, and has a dependency on Media Server.

Interaction Workspace—Display GMS Attached Data

Why:

GMS attaches data to the call prior to routing it to the agent. This attached data is displayed to the agent when the call arrives at the agent desktop (Interaction Workspace), and helps the agent to understand the source of the call, as well as to understand the additional information sent from the customer's device when creating the Callback.

How:

Configuration Manager > Business Attributes

1. Create a new business GMSCaseData attribute of type `Interaction Operational Attribute`.

2. Create new attribute values:

   - `first_name`
   - `last_name`
   - `location_lat`
   - `location_long`
   - `GMS_Call_Direction`
   - `GMS_MatchMethod_AccessNumber`
   - `GMS_MatchMethod_ANI`
   - `GMS_MatchResult`
   - `GMS_MatchReason`
   - `GMS_ServiceName`
   - `GMS_UserData`

3. Set the following Application > InteractionWorkspace options:

   - `interaction-workspace > interaction.case-data.format-business-attribute =` `GMSCaseData`
   - `interaction-workspace > toast.case-data.format-business-attribute = GMSCaseData`

Implemented Scenarios

This sample supports the scenarios described in the Callback Scenarios. These scenarios are server-driven, which means that the server instructs the client with the actions needed to carry out the scenario. The client just needs to perform these actions and the follow-up dialog with the server. Therefore, the client is flexible enough to support any scenario that is built using the same kind of actions. The following actions are supported:

- `DialNumber` - The app makes a phone call when running on a mobile browser.
- `ConfirmationDialog` - The app displays a message requesting the user to confirm a follow-up action.
- `DisplayMenu` - The app displays a menu for the user to select an item that may affect how the scenario proceeds.
- `get-dialog*` - Retrieves the dialog details and displays the dialog to the user. Dialogs are limited to alerts.

This sample also supports the `request-interaction` scenario.

Push notifications through CometD are supported. Delayed scenarios are supported by using push notifications only; the app will not poll the server to be notified about agent availability.

Testing Built-in Scenarios

[+] See the instructions to test the built-in scenarios

> **Important**
> In the following scenarios, if the GMS Match fails, there will be no user data attached
> to the interaction.

Scenario request-interaction Test Procedure
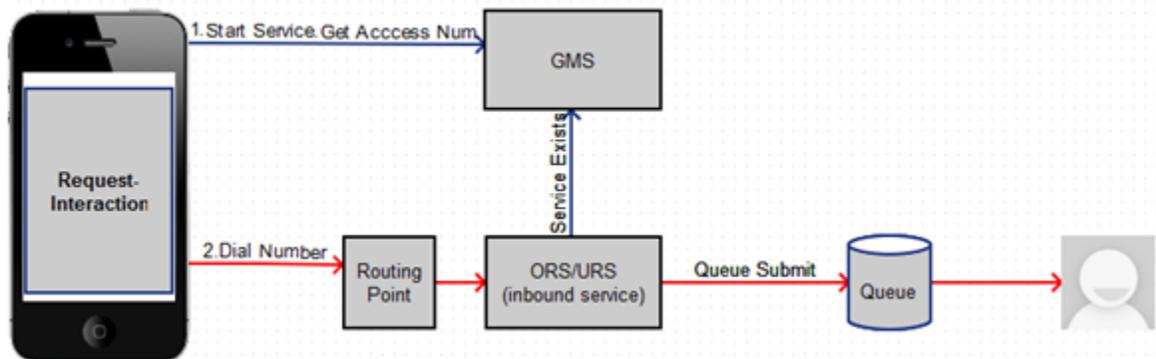
1. On the Agent Desktop:
   - Log in agent.
   - Make voice ready.

2. Using the Javascript sample: Service Management UI > Lab > Sample:
   - Log in agent and make voice ready.
   - Set `Contact#` = `<customer phone from which call will be dialed>`
   - Set `Scenario` = `REQUEST-INTERACTION`
   - Click `Connect`.
   - Dial displayed Number to Call.

3. Expected result:

- Treatment is played.

- Call is routed to agent.

- Toast is displayed with attached data.

- Call is connected to agent.

- For a successful GMS call, `GMS_MatchResult = SUCCESS` is displayed in the agent desktop as attached data.

Compiling and Running the Sample

> ## Important
>
> This step is required only if you download the code sample in order to modify the source code.

Prerequisites

In order to use this sample app, you need to have GMS installed and running, and the services that you want to make use of must be deployed. The source code of this sample is available via a downloadable zip file: Genesys Mobile Services JavaScript Sample Zip File

Access the Sample

1. Download and unzip the .zip file from the above link.

2. Check the **readme.md** file in the unzipped files for the details on starting the sample and configuring it.

> ## Important
>
> - The CometD client is automatically started when the application loads in the browser.
>
> - Make sure that your URL starts with the value specified in GMS > Server > `external_url_base` when you access the Service Management UI.

About the Code

The majority of the code is in two files:

- `index.html` - Controls the presentation aspects of the application, which includes the GMS response

handler.

- `gms.js` - Responsible for interfacing with GMS and as well as managing the CometD connection.

index.html

The following screens are presented to the user and can be displayed by clicking the corresponding tabs.

- `GMS` - The application home screen showing which GMS scenario can be executed.

- `Log` - Displays log messages related to client-server communication and application debug messages.

- `Queue` - For delay scenarios, checks the status of the interaction in the queue (when a request has been placed and is waiting for an agent).

- `Settings` - Application settings can be made on this screen.

gms.js

Two objects are implemented in this file:

- `gmsInterface` - Allows the creation of GMS callback services and delegates responses to `index.html::onResponseRecieved`.

- `gmsNotificationClient` - Responsible for starting the CometD client and connecting to the GMS CometD channel. When the message is received, the callback function `index.html::onCometNotification` is invoked.

Disclaimer