

Puzzel Callback on Web - API Reference

This document contains information on how to use the “Callback on Web” (COW) functionality in Puzzel’s web services.

COW enables Puzzel customers to offer a phone callback option on their website.

The following summarizes the callback scenario:

- The visitor of the website will enter their callback number (and optionally their name) and click a “Call Me” button
- The button will in turn send the configured variables, including the callback number, to a web page hosted by Puzzel
- The variables will be used to configure and add the callback request in the customer queue of choice
- The callback request will be allocated to the next available agent based on how the callback service is configured

Call us on 0125 35 5446, or let us call you back

Your name

Your phone number

➤ Call me back

Call us on 0125 35 5446, or let us call you back

Your name

Your phone number

➤ Call me back

A list called “_WebCallback” is required and must be added to the customer before a callback can be executed. This is done

by Puzzel. This list will contain either a blacklist or whitelist (or a combination) of numbers or variables).

A callback request will be valid for 72 hours. A callback will be activated as soon as an agent becomes available and the request is first in the queue. Puzzel recommends that callback requests are only offered to customers within the opening hours of the call centre.

Web forms and variables

In order for a user to initialize callback requests from the web, a callback web form must be created and built into the company's website. When a user fills in the form and orders the callback, the web page will send relevant information to the Puzzel platform through a Puzzel API.

This chapter explains in detail the variables available for processing on the "Callback on Web" endpoint hosted in the Puzzel platform.

The URL for the endpoint is: <https://api.puzzel.com/contactcentre/cow.aspx>

The endpoint uses the POST request method for processing the variables.

A web form is used as an example during the explanation of each respective variable, and the structure is as shown below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<!-- form data -->
<input type="submit" value="Call Me"/>
</form>
```

The form contains a URI for the "Callback on Web" endpoint, using the POST request method, as well as a "Call Me" button which executes the callback.

The following chapters explain the mandatory and optional variables.

Customer Key

The **customerKey** variable determines the identity of the company sending the variables. This variable will usually be hidden in the form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
```

The example shows that the web form variables are coming from customer 10000.

This variable is mandatory.

Access Point and Country Code

The **accessPoint** and **countryCode** variables determine the phone number and country code of a valid Puzzel access point set to handle callback requests from the web. These variables will usually be hidden in the form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="accessPoint" value="2149xxxx"/>
<input type="hidden" name="countryCode" value="NO"/>
```

The example shows that the callback will be using a Norwegian access point.

These variables are mandatory.

Callback Number

The **callbackNumber** variable will contain the phone number that the visitors enter. This variable will usually be an input in the form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
```

This variable is mandatory.

Queue Key

The **queueKey** variable determines what queue the callback request will be added to while waiting for an available agent. This variable will usually be hidden in the form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
```

The example shows that the callback request will be added to the queue “Q_PHONE_WEB” while waiting for an available agent.

This variable is mandatory, but may be overridden if a list name is provided.

CIQ Type

The **ciqType** variable determines who will be called first; the visitor (request) or the agent. Possible values are “**CallRequestFirst**” and “**CallAgentFirst**”. This variable will usually be hidden in the form, as shown in the example below.

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="hidden" name="ciqType" value="CallAgentFirst"/>
```

This variable is optional. Default is “CallRequestFirst”.

List Name and -Value

The **listName** and **listValue** variables determine what queue the callback request will be added to while waiting for an available agent.

The listName is the name of the customer list that contains the mapping information between the listValue and a queue key, if the listValue is available.

To better illustrate the usage of lists, take a look at the following scenarios:

Mapping the callback number to a specific queue

In this scenario, the visitor will be mapped to a specific queue based on the entered callback number.

Let’s say the list name is “Number2Area” and looks like this:

List content (records in the list)		
	key	value
▶	20000000:21999999	Q_AREA_NORTH
	22000000:23999999	Q_AREA_WEST
	24000000:25999999	Q_AREA_EAST
	26000000:27999999	Q_AREA_SOUTH

The example below shows how this could be configured:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="hidden" name="listName" value="Number2Area"/>
```

The default queue where the callback is added is “Q_PHONE_WEB”. If, however, a match is found in the “Number2Area” list, the matching queue key will override the default.

So if the callback number is:

- 22334455, the queue key will be “Q_AREA_WEST”
- 26262626, the queue key will be “Q_AREA_SOUTH”
- 38383838, the queue key will be “Q_PHONE_WEB”, the default value

Mapping the list value to a specific value

In this scenario, the visitor will be mapped to a specific queue based on the list value entered. The list value here is an input where the visitor will specify a postal code. Let’s say this list is named “PostalCode2Area”, and looks like this:

List content (records in the list)		
	key	value
▶	0000:1999	Q_AREA_NORTH
	2000:3999	Q_AREA_WEST
	4000:5999	Q_AREA_EAST
	6000:7999	Q_AREA_SOUTH

The example below shows how this could be configured:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="hidden" name="listName" value="PostalCode2Area"/>
<input type="text" name="listValue"/>
```

The default queue to which the callback is added is “Q_PHONE_WEB”. If, however, a match is found in the list “PostalCode2Area”, based on the list value the matched queue key will override the default.

So if the list value is:

- 2999, the queue key will be “Q_AREA_WEST”
- 6999, the queue key will be “Q_AREA_SOUTH”
- 8888, the queue key will be “Q_PHONE_WEB”, the default value

Max attempts and seconds between attempts

The **maxAttempts** and **secondsBetweenAttempts** variables determine the number of callback attempts, and the number

of seconds to wait between attempts. These variables will usually be hidden in the form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="hidden" name="maxAttempts" value="3"/>
<input type="hidden" name="secondsBetweenAttempts" value="300"/>
```

The example shows that the callback request will be executed three times with a minimum 300 seconds interval between each attempt.

These variables are optional. Default: maxAttempts = 1, secondsBetweenAttempts = 30

Scheduled Date Time

The **scheduledDateTime** variable determines when the callback should occur. The variable expects the data to be in ISO8601 format. In a web form, the variable would typically be filled using some sort of date time picker written in JavaScript.

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="text" name="scheduledDateTime"/>
```

The following shows a valid formatted variable:

2018-01-16T15:11:00+0100

This means 16th January 2018, 15:11PM, GMT+1 timezone, according to the following format key: {year}-{month}-{day}T{hour}:{minute}:{second}{timezone}

It's important to always include the timezone in the variable. Leaving this out may result in inconsistent and incorrect assignment of timezones to the scheduled request.

This variable is optional.

Request category and -description

The **requestCategory** and **requestDescription** variables determine the additional visual information that can be added to the callback request. This visual information is used by the Queue Overview functionality in the Puzzel Agent Application.

QUEUE	QUEUED	AGENTS	SLA	OFFERED	ANSWERED	UNANSWERED	ANSWER %	ANSWER % SLA	OUTBOUND
Web Callback	1	0		7	6	1	86 %		1

The above shows that a callback request has been added to a queue called Web Callback. When clicking 1 in the QUEUED column, the following will be shown:

Web Callback						
	QUEUE	NUMBER	HOLD TIME	FROM	VIP	OUTBOUND
1	Web Callback	21490658	14s		0	1

The above shows the callback number that has been added to the Web Callback queue. The requestCategory and requestDescription variables make it possible to add some visual information to the same request, e.g. like this:

Web Callback						
	QUEUE	NUMBER	HOLD TIME	FROM	VIP	OUTBOUND
1	Web Callback	21490658	6s	Ola Nordmann	0	1

The above shows the same callback number, but adds colour and name to the queue display. Note: To enable this functionality for the queue, the “category” and “description on caller” settings must be enabled in the Administration Portal.

The above screenshot is achieved by using the following values on the Request Category and Request Description:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="accessPoint" value="2149xxxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="hidden" name="requestCategory" value="Col:red;ColText:blue;"/>
<input type="text" name="requestDescription"/>
```

The requestCategory variable contains colour information, and the requestDescription variable contains the name of the visitor as entered in the form.

The following “colour words” are supported; black, blue, bluegreen, green, grey, magenta, orange, pink, red, turquoise, violet, and yellow.

Additionally, it is possible to use RGB values. The following produce identical results:

```
<input type="hidden"
name="requestCategory"
value="Col:red;ColText:blue;"/>

<input type="hidden"
name="requestCategory"
value="Col:#FF0000;ColText:#0000FF;"/>
```

These variables are optional.

Additional variables used by the IVR

It’s possible to add additional variables that can be used by the IVR when executing the callback request.

The name of these variables is provided by Puzzel in the IVR and callback design, but common for all of them are the prefix “kml”.

These variables may get input from a form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="accessPoint" value="2149xxxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="text" name="kml_VisitorAddress"/>
<input type="text" name="kml_ProductChoice"/>
```

The example shows two variables collected by the form, and passed to the IVR solution. These variables could, for example, be included in the screen popup shown in the Agent Application. The web service will remove the prefix before sending it to the IVR.

If the variable value contains the characters \$ or %, the IVR will interpret this as a variable-invalue. This means that the IVR will try to resolve the variable-in-value with an internal value. If an internal value is not found, the variable-in-value is removed.

So, if the variable/value looks like this: ‘kml_VisitorName=Ola’, nothing will happen. However, if the variable/value looks like this: ‘kml_VisitorName=Ola \$Surname’, the IVR will remove the ‘\$SurName’ from the value (if it does not exist).

Sometimes we may have a variable/value combination that will contain these two characters (for example in a URI), and the IVR should not try to resolve this. If this is the case, some special prefixes can be used. This is the list of prefixes, and their corresponding meaning:

- kml_ type is set to String and variable-in-value is resolved (old way)

- kmls_, type is set to String and variable-in-value is not resolved
- kmlsr_, type is set to String and variable-in-value is resolved
- kml_i_, type is set to Integer and variable-in-value is not resolved
- kmlir_, type is set to Integer and variable-in-value is resolved
- kmlb_, type is set to Boolean and variable-in-value is not resolved
- kmlbr_, type is set to Boolean and variable-in-value is resolved

Web request redirections

There are three web request redirection variables supported: redirectFull, redirectOK, and redirectError.

If a queue size limit is set on the queue key specified, and that limit is reached, the web page hosted by Puzzel will redirect the visitor to the URL set in the redirectFull variable.

If the callback request is added successfully, the web page hosted by Puzzel will redirect the visitor to the URL set in the redirectOK variable.

If the callback request is not added successfully, the web page hosted by Puzzel will redirect the visitor to the URL set in the redirectError variable.

These variables will usually be hidden in the form, as shown in the example below:

```
<form action="https://api.puzzel.com/contactcentre/cow.aspx" method="post">
<input type="hidden" name="accessPoint" value="2149xxx"/>
<input type="hidden" name="countryCode" value="NO"/>
<input type="text" name="callbackNumber"/>
<input type="hidden" name="customerKey" value="10000"/>
<input type="hidden" name="queueKey" value="Q_PHONE_WEB"/>
<input type="hidden" name="redirectFull" value="http://server/full.php"/>
<input type="hidden" name="redirectOK" value="http://server/ok.php"/>
<input type="hidden" name="redirectError" value="http://server/error.php"/>
```

The example shows the respective redirects that are wanted on each of the above outcome scenarios.

These variables are optional.