

## Puzzel Call-back

This article details Puzzels Call-back web form product, enabling Puzzel customers to enrich their websites with a call me feature. We use the same call-back principle as for voice/phone, except that the call-back is ordered from a web page rather than from a phone queue / IVR. Users are asked to enter the phone number they want to be reached on, and alternatively some additional information such as name, customer number, subject for the request, etc. The additional information can e.g. be used for customer list lookup, CRM pop-up, or merely to appear in the PAA when answering the call (screen pop-up).

Puzzels Web Call-back product can, in the same way as for call-back in a voice queue, be triggered to call first an available agent and then the person ordering the call-back (the number entered on the web page), or vice versa where the customer is called first. The default setting is to call an available agent first.

## Call-back form

In order for a user to initialise call-back requests from the web, a call-back web form must be created and integrated in the customers website.

When a customer fills out the form, the webpage passes the information entered to the Puzzel platform, where it is screened and processed.

- If the received content is OK, the user is redirected to a confirmation page on the customers web server, and a queue tag is created in the Puzzel solution
- If the received content is not OK, the user is redirected to an Error page on the PAA's web server, and a queue number will not be created in the Puzzel solution

When the queue tag moves to first in line, the call-back will take place.

An example of a call-back form for a website

The Call-back form can be customised and can contain a variety of input fields. The simplest way to create a call-back form is to only create a field for the customer to enter their telephone number.

One way of customising the web form is to add more input fields for the customer to fill out, thus getting more precise information about him and his request, e.g.:

- Select a queue (list of multiple queues, such as Product, Support, X, Y, Z)
- Select the subject of your request (e.g. for a list where every choice leads to a queue)
- Enter your post code (e.g. can lead to the correct local office queue)
- Enter your customer number (e.g. can lead to for example a customer list lookup / CRM pop-up / screen pop-up)

Call-back from the web works identically to call-back ordered from a phone queue. This means it is in real-time, so when a caller orders it, their place in queue will be immediately reserved. This means that if there is no check queue availability in the call-back form, and a caller reserves their place outside the contact centre opening hours, this caller will be the first person in line when the contact centre opens up.

As with all Puzzel services, the web call-back module is flexible. It is possible to define rules for which agents requests should go to and with what priority over other inbound requests. It is also possible to have multiple call-back requests within a website, and to define the queue into which they are sent. Web call-back increases website responsiveness by minimising the amount of effort required for a customer to get in touch.

## Call-back form code example

Below is an example HTML implementation of the call-back form. This code is usually pre-defined by Puzzel, but the highlighted fields can be adjusted if necessary. If you change any fields other than the highlighted ones, the solution may not work as it should.

```
<form method="post" action="https://api.puzzel.com/contactcentre/cow.aspx">
```

```

<input type="hidden" value="12345" name="customerKey">
<input type="hidden" name="accessPoint" value="81511569" />
<input type="hidden" name="countryCode" value="NO" />
<input type="hidden" name="queueKey" value="q_sales" />
<input type="hidden" name="ciqType" value="CallAgentFirst" />
<input type="hidden" name="maxAttempts" value="2" />
<input type="hidden" name="secondsBetweenAttempts" value="30" />
<input type="hidden" value="http://www.puzzel.com/thank-you/" name="redirectOK" />
<input type="hidden" value="http://www.puzzel.com/error-page/" name="redirectError" />
<label>Your Name</label>
<input type="text" name="requestDescription">
<label>Your Phone Number</label>
<input type="text" name="callbackNumber">
<input type="hidden" name="kml_crm_contact_id" value="123456789" />
<input type="submit" value="Call Me" />
</form>

```

## Check queue availability

The opening hours for the web call-back feature are not related to the solutions time modules, and must be defined on the customers website. However, a queue can be closed for web call-backs, even during opening hours, if certain pre-defined variables are met. These variables can be defined via JavaScript on the customers website to check if the queue is open before the Call Me button is displayed. For example, you can close the queue if there are no agents logged on to it. Callers will then not have the option to initiate call-backs from the page since the queue is closed.

Customers must have a jQuery library installed on their website in order to check queue availability. Both the customerKey and the visualQueueId variables are unique to each customer. customerKey is the customer number users use to log in with Puzzel, and visualQueueId is given out to a customer once the queue web access service is activated by a Puzzel technician.

### Note

A Puzzel technician must activate the queue web access feature before this information is made available. This is activated on a per-queue basis, enabling you to only have it activated on the relevant queue(s). When activated, the queue information listed in chapter 4.2 should be considered public.

## JavaScript example

```

var readyToClickCallback=false;
function myJsonMethod(r){
  if(r && r.Queue){
    if(r.Queue.AgentsLoggedIn >0){
      // Agents logged in, call-back is offered, hide closed message
      readyToClickCallback=true;
      $("callbackClosed").hide();
    } else {
      // Call-back closed
      // Here you can hide your callback elements and show a closed message on your webpage, example below.
      $("callbackOpen").hide();
      $("callbackClosed").show();
    }
  }
}
$(function(){
  $.jax({
    url:"https://chat.intele.com/Queue/GetQueueStatus?callback=?",
    type: "GET",
    crossDomain: true,
    contentType: "application/json; charset=utf-8",
    data: {
      customerKey:"CustomerNumber",
      visualQueueId: 10292
    },
    dataType: "jsonp",

```

```
jsonpCallback: "myJsonMethod",  
error: function (httpReq,status,exception) { }  
});  
});
```

### Example of retrieved queue information (JSON format)

```
"WaitTimeMax":0,  
"WaitTimeAverage":0,  
"QueueSize":0,  
"AgentsLoggedIn":0,  
"AgentsUnavailable":0,  
"Sla":-1,  
"QueueSizeCiq":0,  
"UnblockedLoggedIn":-1,  
"UnblockedUnavailable":0,  
"QueueSizePreferred":0,  
"CallsOfferedToday":0,  
"CallsAnsweredToday":0,  
"CiqsOfferedToday":0,  
"CiqsAnsweredToday":0,  
"CallsAnsweredWithinSla":0,  
"AlarmFlag":0,  
"Id":10292,  
"Description":"Chat salg",  
"OpenForWebAccess":true
```