

## Queue Overview - Realtime Data

Let's have a look at the steps needed to get access to Queue Overview data similar to the data under the Real-time tab shown in Connect Control. In any use case, one of the first things to do is authenticate and get an 'accesstoken' which you will use for all subsequent REST calls.

### Step 1 Authenticate:

`https://api.intele.com:443/Connect/ContactCentre5/auth/credentials?userName=CustomerNumber%5CuserName&password=password`

Here we are sending 'customerNumber\userName' & 'password' as strings.

Result gives you the "Access Token", referred to as `accessToken` from hereon (the value of "result").

```
{
  "result": "csid!9920633A7AE022CD14D62B71E030B98EE44C3",
  "code": 0
}
```

### Step 2 AccessTokenInformationGet:

We need to get the details for this `accessToken` to determine the expiry and the `userId` of the authenticated user for subsequent REST calls.

`https://api.intele.com:443/Connect/ContactCentre5/accesstokeninformation?accessToken=csid!9920633A7AE022CD14D62B71E030B98EE44C3`

This request only needs the 'accessToken' we got earlier.

Result gives us the all important `userId` and the `accessTokenExpiry` values. You should re-use the `accessToken` until it expires.

```
{
  "result": {
    "customerId": 11295,
    "userGroupId": 20617,
    "userId": 158888,
    "languageId": 1,
    "languageCode": "EN",
    "accessTokenExpiry": "/Date(1452289413620-0000)/"
  },
  "code": 0
}
```

A note about the `dateTime` format. Intelcom's Microsoft Web API returns JSON dates as `/Date(1198908717056-0700)/`. So if you want to create a `Date` object you can do something like below.

```
var date = new Date(parseInt(jsonDate.substr(6)));
```

The `substr` function takes out the `"/Date("` part, and the `parseInt` function gets the integer and ignores the `"/"` at the end. The resulting number is passed into the `Date` constructor.

You could also use `moment.js` for better handling of `dateTime` response strings.

```
moment(string) // simply pass the string
moment("/Date(1198908717056-0700)/");
```

```
// would result in: 2007-12-28T23:11:57.056-07:00
```

### Step 3 VisualQueueStateAndTickerList:

We are now ready to request the data for all queues.

`https://api.intele.com:443/Connect/ContactCentre5/99206/visualqueues/stateinformation/All?accessToken=csid!9920633A7AE022CD14D62B71E030B98EE44C3&userId=158888`

In this request we send the 'accessToken' and the 'userId' along with an optional 'visualQueueResult'.

Notice that there are several visualQueueResult options we can ask for (All, UserRelevant, UserMember). This determines the scope of the data we get back.

(Refer to the API Explorer for descriptions).

We now have details of the current status of each queue as well as the metrics for requests so far (data for Today, starting from midnight, upto the moment we made this request).

```
{
  "result": [
    {
      "id": 23740,
      "description": "Chat",
      "waitTimeMaxSeconds": 0,
      "waitTimeAverageSeconds": 0,
      "queueSize": 0,
      "agentsLoggedIn": 0,
      "agentsInPause": 0,
      "agentsReady": 0,
      "agentsUnavailable": 0,
      "sla": 0,
      "queueSizeCiq": 0,
      "unblockedLoggedIn": -1,
      "unblockedUnavailable": 0,
      "queueSizePreferred": 0,
      "callsOfferedToday": 0,
      "callsAnsweredToday": 0,
      "ciqsOfferedToday": 0,
      "ciqsAnsweredToday": 0,
      "callsAnsweredWithinSla": 0
    },
    {
      "id": 23985,
      "description": "Email",
      "waitTimeMaxSeconds": 0,
      "waitTimeAverageSeconds": 0,
      "queueSize": 0,
      "agentsLoggedIn": 1,
      "agentsInPause": 1,
      "agentsReady": 0,
      "agentsUnavailable": 1,
      "sla": -1,
      "queueSizeCiq": 0,
      "unblockedLoggedIn": -1,
      "unblockedUnavailable": 0,
      "queueSizePreferred": 0,
      "callsOfferedToday": 0,
      "callsAnsweredToday": 0,
      "ciqsOfferedToday": 0,
      "ciqsAnsweredToday": 0,
      "callsAnsweredWithinSla": 0
    },
    {
      "id": 24017,
      "description": "Facebook",
      "waitTimeMaxSeconds": 0,
      "waitTimeAverageSeconds": 0,
      "queueSize": 0,
      "agentsLoggedIn": 1,
      "agentsInPause": 1,
      "agentsReady": 0,
      "agentsUnavailable": 1,
      "sla": 0,
      "queueSizeCiq": 0,
      "unblockedLoggedIn": -1,
      "unblockedUnavailable": 0,
      "queueSizePreferred": 0,
      "callsOfferedToday": 0,
      "callsAnsweredToday": 0,
      "ciqsOfferedToday": 0,
      "ciqsAnsweredToday": 0
    }
  ]
}
```

```

"callsAnsweredWithinSla": 0
},
{
  "id": 27584,
  "description": "Sales",
  "waitTimeMaxSeconds": 0,
  "waitTimeAverageSeconds": 0,
  "queueSize": 0,
  "agentsLoggedOn": 2,
  "agentsInPause": 1,
  "agentsReady": 1,
  "agentsUnavailable": 1,
  "sla": 0,
  "queueSizeCiq": 0,
  "unblockedLoggedIn": -1,
  "unblockedUnavailable": 0,
  "queueSizePreferred": 0,
  "callsOfferedToday": 0,
  "callsAnsweredToday": 0,
  "ciqsOfferedToday": 0,
  "ciqsAnsweredToday": 0,
  "callsAnsweredWithinSla": 0
},
{
  "id": 27585,
  "description": "Support",
  "waitTimeMaxSeconds": 0,
  "waitTimeAverageSeconds": 0,
  "queueSize": 0,
  "agentsLoggedOn": 1,
  "agentsInPause": 1,
  "agentsReady": 0,
  "agentsUnavailable": 1,
  "sla": 0,
  "queueSizeCiq": 0,
  "unblockedLoggedIn": -1,
  "unblockedUnavailable": 0,
  "queueSizePreferred": 0,
  "callsOfferedToday": 0,
  "callsAnsweredToday": 0,
  "ciqsOfferedToday": 0,
  "ciqsAnsweredToday": 0,
  "callsAnsweredWithinSla": 0
}
],
"code": 0
}

```

So at this stage you can decide on how to present the data. A simple table view might look like this:

Queue Name	Queueing Now	Agents LoggedOn	Requests Offered	Requests Answered	%Answered within SLA

"description" = the name of the queue.

"waitTimeMaxSeconds" = maximum wait time for the requests in queue right now.

"queueSize" = the number of requests in queue right now.

"callsOfferedToday" = number of requests offered to Queue so far today.

"callsAnsweredToday" = number of requests answered so far today.

"callsAnsweredWithinSla" = number of requests answered within the set SLA.