

Last Updated: March 15, 2018

## Agent Management API

### Overview

The Agent Management API consists of HTTPS POST requests that handle the creation of a new agent within an account, the changes to existing agent information, and the removal of an agent from the account. The requests respond with a status code indicating the result of the request.

### Agent Management API Requests

The HTTPS interface requires a FORM to POST to information to a Live Help On Demand server using the following request:

```
https://admin.instant-service.com/AgentAPI
```

### Request Authentication

The following authentication mechanisms are checked before retrieving the data for a request:

- *Requesting IP Address* - The authorized IP addresses (contact Oracle Live Help On Demand Customer Support to configure these addresses).
- *API Password* - The password for the API (contact Oracle Live Help On Demand Customer Support to configure this password).

---

**Note:** The configured authorized IP addresses should be as restrictive as possible. Configuring IP ranges is strongly discouraged.

---

### Response Status Codes

The following response codes are returned indicating the status of a request:

Status Code	General Description
100	Success
200	Bad Request Syntax
201	HTTPS Request Only
202	POST Request Only
300	Account Not Available (disabled, or invalid account ID)
301	API not enabled for account
302	Unauthorized IP address
303	Invalid API password
304	Csr Not Found, or Role Not Found
305	Username already exists
306	Password Criteria Not Met
400	Unforeseen server failure

## Create Agent Request

The following fields and values must be used in the form. The field names are **case-sensitive** and must match exactly as listed in the following table:

PARAM	VALUE	TYPE	REQUIRED
ai	account ID	integer	YES
pd	api password	string	YES
action	create	string	YES
fname	csr first name	string	YES
lname	csr last name	string	YES
email	e-mail address of csr/agent	string	YES
username	agent user name	string	YES
passwordformat	agent password format; "plaintext" or "hashed"	string	YES
password	agent password	string	YES
roleid	a valid, existing role ID for the account	integer	NO
assignalldepts	assign agent to all departments; "true" or "false"	string	NO
forcepasswordchange	force agent to change password; "true" or "false"	string	NO
forcepasswordprompt	prompt for agent to change password; "true" or "false"	string	NO
neverexpirepassword	designate the agent password as one that never expires; "true" or "false"	string	NO
customdata1	user-defined value	string	NO
customdata2	user-defined value	string	NO
customdata3	user-defined value	string	NO
customdata4	user-defined value	string	NO
customdata5	user-defined value	string	NO
trackingid	Tracking ID used for integration with third-party analytics tracking	string	NO

## Response

If successful, a response is generated in the following format:

<i>Format:</i> <code><message> <new csrid>	<i>Example:</i> 100 Agent Created 917123441
--	---

If not successful, you receive an error code. Check the **Status Code section** for possible error codes and their descriptions.

## Update Agent Request

The following fields and values must be used in the form. The field names are **case-sensitive** and must match exactly as listed in the following table:

PARAM	VALUE	TYPE	REQUIRED
ai	account ID	integer	YES
pd	api password	integer	YES
ci	csrID	string	YES
action	update	string	YES
fname	csr first name	string	NO
lname	csr last name	string	NO
email	e-mail address of csr/agent	string	NO
username	agent user name	string	NO
passwordformat	agent password format; "plaintext" or "hashed"	string	NO
password	agent password	string	NO
roleid	A valid, existing role ID for the account	integer	NO
forcepasswordchange	force agent to change password; "true" or "false"	string	NO
forcepasswordprompt	prompt for agent to change password; "true" or "false"	string	NO
neverexpirepassword	designate the agent password as one that never expires; "true" or "false"	string	NO
customdata1	user-defined value	string	NO
customdata2	user-defined value	string	NO
customdata3	user-defined value	string	NO
customdata4	user-defined value	string	NO
customdata5	user-defined value	string	NO
trackingid	Tracking ID used for integration with third-party analytics tracking	string	NO

## Response

If successful, a response is generated in the following format:

<i>Format:</i> <code><message>	<i>Example:</i> 100 Agent Updated
-----------------------------------	--------------------------------------

If not successful, you receive an error code. Check the **Status Code section** for possible error codes and their descriptions.

## Delete Agent Request

The following fields and values must be used in the form. The field names are **case-sensitive** and must match exactly as listed in the following table:

PARAM	VALUE	TYPE	REQUIRED
ai	account ID	integer	YES
pd	api password	string	YES
ci	csrID	integer	YES
action	delete	string	YES

## Response

If successful, a response is generated in the following format:

<i>Format:</i> <code><message>	<i>Example:</i> 100 Agent Deleted
-----------------------------------	--------------------------------------

If not successful, you receive an error code. Check the **Status Code section** for possible error codes and their descriptions.

## Availability API

### Overview

The Availability API is an interface allowing for the retrieval and/or modification of availability settings of chat departments and of agents logged into the Oracle Live Help On Demand Agent Console.

The API accepts HTTPS Form POST requests as input, and returns either XML data as a successful response or a simple error code if the request was not successful.

### Activating the Availability API

Before you can start using the Availability API, an Account Administrator must first activate the API by doing the following:

- Enabling the API for the Account.
- Defining client IP restrictions for API requests (to limit sites that can access the API).
- Generating a password for accessing the API.

### Locating the Chat Server Hosting the Account

To begin using the API, determine which chat server (if any) is hosting the account. To determine the chat server, an API request using an HTTPS Form POST should be made to the following URL:

`https://admin.instantservice.com/AvailabilityAPI`

The following fields are required by this request:

PARAM	VALUE	MAX SIZE	REQUIRED
ai	account ID	n/a	YES
pd	api password	10	YES

The result of the previous call is either a simple error code indicating a problem has occurred with the request or XML output indicating the request was successfully processed.

Possible error codes returned from API calls to the admin server are listed in the **Error Responses** section of this document. All error responses are sent back with a content-type of "text/plain; charset=utf-8".

```
<?xml version="1.0" encoding="UTF-8"?>
<availabilityapi version="2.0">
<account account_id="####" chat_server="XXXX"/>
</availabilityapi>
```

The XML contains the ID of the account in the `account_id="####"` attribute. The chat server hosting the account can be found in the `chat_server="XXXX"` attribute. If the account is not currently on any chat server, this attribute is blank.

### Requesting Availability from the Chat Server

Requests for availability details must be made to the specific chat server hosting the account using an HTTPS Form POST to the following URL:

```
https://<chat_server>.instantservice.com/<account_id>/AvailabilityAPI
```

where `<chat_server>` is the name of the chat server retrieved from the initial API call to the admin server, and `<account_id>` is your account id.

The following parameters are used with this request to indicate which agents' availability is being requested:

PARAM	VALUE	MAX SIZE	REQUIRED
ai	account ID	n/a	YES
pd	api password	10	YES
action	<b>get</b> (default if no <b>action</b> param sent)	3	NO
di	comma separated list of departmentids	*	NO**
ci	comma separated list of csrids	*	NO**
cti	comma separated list of csr tracking integration ids	*	NO**
filter	one of the following: <b>avail</b> – request only available agents and/or departments <b>unavail</b> – request only unavailable agents and/or departments <b>inchat</b> – request only agents in a chat <b>notinchat</b> – request only agents not in a chat	9	NO

\* Refer to the Length Limitations section below for details on the maximum size allowed for FORM POST parameters.

\*\* Additional Notes:

- 1) If the `di`, `ci`, and `cti` parameters are all omitted from the request, the API returns all online departments and all online agents for the requested account.
- 2) You may use the `di`, `ci`, and `cti` all within the same request. Using a combination of the `di`, `ci`, and `cti` parameters will merge agent and department results in the returned set.
- 3) If the `di` parameter is present, the API will return online departments matching the passed in department ids, as well as all agents assigned to those departments.
- 4) If either the `ci` or `cti` parameter is present, the API will return online agents matching the passed in `csr` ids or tracking ids, respectively.

The result of the **get request** is either a simple error code indicating a problem has occurred with the request or XML output indicating the request was successfully processed.

Possible error codes returned from API calls to the chat server are listed in the **Error Responses** section of this document. All error responses are sent back with a content-type of "text/plain; charset=utf-8".

```

<?xml version="1.0" encoding="UTF-8"?>
<availabilityapi version="2.0"> <account account_id="####"/>
<dept_state dept_id="####"/>
  <availability_state>XXXXXX</availability_state>
  <queue_hours_state>XXXXXX</queue_hours_state>
</dept_state>
  <agent_state agent_id="####" agent_tracking_integration_id="XXXXX">
  <availability_state>XXXXXX</availability_state>
  <chats_in_session>###</chats_in_session>
  <max_chats>###</max_chats>
  <max_reply_mail>###</max_reply_mail>
  <reply_mail_in_session>###</reply_mail_in_session>
  <availability_state_start_time>#####</availability_state_start_time>
  </agent_state>
</availabilityapi>

```

Each department returned in the request has a separate <dept\_state> element which contains all pertinent information to that department.

The <dept\_state> element contains the following attributes:

dept_id	The department id of the department as defined in the Live Help On Demand Account Admin.
---------	--

Within the <dept\_state> element there are four child elements:

<availability_state>	The availability state that the department is currently in; is either <b>available</b> or <b>unavailable</b> .
<queue_hours_state>	The hours of operation policy that department queue is operating under, <b>open all hours</b> , <b>open shift hours</b> , or <b>close all hours</b> .

Each agent returned in the request has a separate <agent\_state> element which contains all pertinent information to that agent.

The <agent\_state> element contains the following attributes:

agent_id	The CSRID of the agent as defined in the Live Help On Demand Account Admin.
agent_tracking_integration_id	The Tracking Integration ID as defined in the Live Help On Demand Account Admin.

Within the <agent\_state> element, there are the following child elements:

<availability_state>	The availability state that the agent is currently in; is either <b>available</b> , <b>unavailable</b> , or <b>unavailable (external)</b> .
<chats_in_session>	The current # of chats in session with this agent.
<max_chats>	The maximum number of chats that can be taken by the agent.
<reply_mail_in_session>	The current # of reply mail messages in session with this agent.
<max_reply_mail>	The maximum number of mail messages that can be taken by this agent.
<availability_state_start_time>	The time at which the availability state was last changed.

### Setting Availability

Requests to set availability must be made to the specific chat server hosting the account using an HTTPS Form POST to the following URL:

https://<chat\_server>.instantservice.com/<account\_id>/AvailabilityAPI

where <chat\_server> is the name of the chat server retrieved from the initial API call to the admin server, and <account\_id> is your account id.

### Setting Department Availability

The following parameters can be sent with this request to set the chat queue hours of operation for a department:

PARAM	VALUE	MAX SIZE	REQUIRED
ai	<i>account ID</i>	n/a	YES
pd	<i>api password</i>	10	YES
action	<b>set</b>	3	YES
di	<i>comma separated list of department ids</i>	*	NO**
qh	<i>openallhours, openshifthours, or closeallhours ***</i>	14	NO**

\* Refer to the Length Limitations section below for details on the maximum size allowed for FORM POST parameters.

\*\* Setting department chat queue hours requires both the di and qh parameters.

\*\*\* After changing department chat queue hours, the updated department's availability state may take up to 20 seconds to take effect.

### Setting Agent Availability

The following parameters can be sent with this request to set the availability state of agents:

PARAM	VALUE	MAX SIZE	REQUIRED
ai	<i>account ID</i>	n/a	YES
pd	<i>api password</i>	10	YES
action	<b>set</b>	3	YES
ci	<i>comma separated list of csids</i>	*	NO**
cti	<i>comma separated list of csr tracking integration ids</i>	*	NO**
av	<i>available or unavailable***</i>	11	NO**

\* Refer to the Length Limitations section below for details on the maximum size allowed for FORM POST parameters.

\*\* Setting agent availability requires either the ci or cti parameters (or both), as well as the av parameter.

\*\*\* Generally, an agent's current availability state is not retained should the agent log out of the Agent Console or become disconnected. When the agent logs back in, the availability state is automatically set to the setting configured for the agent in the **Agents: Access: Permissions** page of the Account Administration. Consequently, it is best to configure the agent's initial state to an **unavailable** setting in the Account. This would ensure the agent remains in an unavailable state should the agent log out or become disconnected when the agent's current state is **unavailable (external)** as set by the Availability API.

### Response

The result of the set call is either a simple error code indicating a problem has occurred with the request or XML output indicating the request was successfully processed. The response does not indicate success or failure. To check if a **set** request was successful, you must submit a subsequent **get** request.

Possible error codes returned from API calls to the chat server are listed in the Error Responses section of this document. All error responses are sent back with a content-type of "text/plain; charset=utf-8".

A successful response returns XML with a content-type of "text/xml; charset=utf-8". The following example shows what the XML looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<availabilityapi version="2.0">
<account account_id="####"/> </availabilityapi>
```

The XML contains the ID of the account in the `account_id="####"` attribute of the `<account>` tag.

With the HTTP POST request, the length of the body (for example, the length of the combined total of the parameters passed) must be no longer than 4096 bytes. If the combined total length of the request exceeds this limit the request must be broken up into several requests.

### Request Authentication

The following authentication mechanisms are checked before accepting the request:

- **Requesting IP Address** – The configured authorized IP addresses (contact Live Help On Demand to configure these addresses).
- **API Password** – The password passed to the API must match that configured by Live Help On Demand.

---

**Note:** The configured authorized IP addresses should be as restrictive as possible. Configuring IP ranges is strongly discouraged.

---

### Error Responses

Possible error codes returned from API calls to either the admin server or the chat server are listed below.

Code	General Description
200	Bad Request Syntax
201	HTTPS Request Only
202	POST Request Only
300	Account Not Available (offline, or invalid account) *
301	API Not Enabled for Account
302	Unauthorized IP Address
303	Invalid API Password
400	Unforeseen Server Failure

\* The 300 error code is returned by the Admin Server only when the passed account id parameter is invalid; it does not take into consideration whether the account is currently online or offline.

### Response XML Schema Definition

Refer to the separate `AvailabilityAPI.dtd` file that contains the XML Schema definition for the Agent Availability API XML response.