



Prerequisites

- CIC 4.0 SU6 or later
- I3_FEATURE_ICWS_SDK license on the server

Core Concepts

Ports

- HTTP <http://<server>:8018/icws>
 HTTPS <https://<server>:8019/icws>

HTTP Status Codes

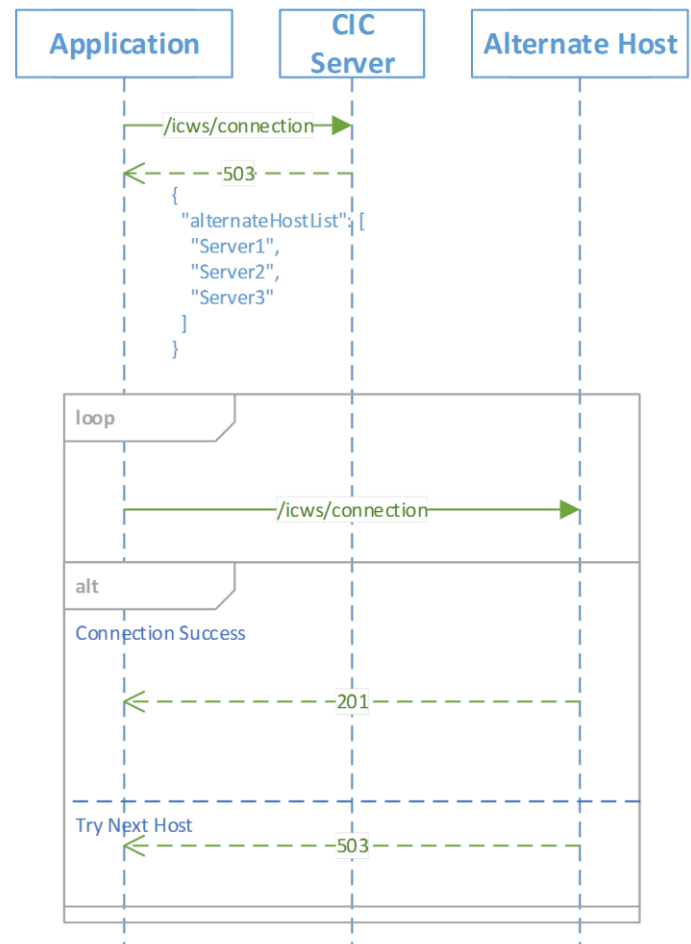
200	OK - The request was successful
201	Created
202	Accepted
204	No content – the request to delete was acknowledged.
400	Bad Request - The request was invalid. This can occur when the message body is malformed or missing required items. This will also occur if a required header or query string parameter is missing or invalid.
401	Unauthorized - The specified session token is missing, invalid, or has expired.
403	Forbidden – The session user does not have permission to perform the action.
404	Not found – The object does not exist
410	Gone - This resource has been deprecated and removed from the API.
500	Internal Server Error - An unexpected error occurred while processing the request.
503	Service Unavailable

Session Lifetime

An ICWS session cookie becomes invalid when the corresponding session on the server is disconnected. A session will be disconnected if the connection resource is deleted via API request. A session may also be disconnected on the server for other reasons, like if the same user account begins a session from another client or if the ICWS service is restarted. If a session is active but the client is not making regular requests to the API, the session will eventually expire and be disconnected on the server. Subsequent API access will require a new authenticated session to be started.

Connecting to Switchover Pair or Off Server Session Manager

When connecting to a CIC server that is part of a switchover pair or has off server session managers associated with it, the initial call to [POST /icws/connection](#) can be made to any of those servers. If there is a better server to connect to, a **503** HTTP status code will be returned as well as an alternate host list. The alternate host list is an ordered list of servers to connect to. In an environment with multiple off server session managers, the alternate host list will be ordered for load balancing so the first server in the list is the one with the least amount of load. Continue calling [POST /icws/connection](#) on each server in that list in order as long as a **503** status code is returned. When a **201** status code is returned, that is the server to continue making requests to.



Date/Time Formatting

Date and Time formats are expressed in JSON objects as string values with a specified format. All values are always expressed in Coordinated Universal Time (UTC). The format, which is a strict subset of ISO 8601, is as follows:

Date - YYYYMMDD

Time - hhmmssZ or hhmmss.fffZ

Date Time - YYYYMMDDThhmmssZ or YYYYMMDDThhmmss.fffZ

The 'T', 'Z', and '.' characters are literal and YYYY, MM, DD, hh, mm, ss are all fixed width representations. If the value does not have millisecond precision, the '.' and millisecond digits are omitted.

Versioning

The ICWS API is versioned in two important ways. Groups of related APIs, called features, are versioned together as Feature Versions. API consumers should use the Feature Versions when determining which APIs are supported by the server. The Interaction Center server installation also has a version number is made available for informational purposes.

Feature Version

The Versions for all Feature on an ICWS server can be acquired using [GET /icws/connection/features](#). If an application is interested in a specific Feature, [GET /icws/connection/features/{featureId}](#) can be used to get a single Version. If a feature is not included, it can be assumed that the server does not support that feature.

Server Version

A server's overall installed version information can also be queried using [GET /icws/connection/version](#). This general version information is primarily for informational purposes but in rare cases may be useful to programmatically determine whether required functionality is available for broader API additions that were introduced with a release.

Curl Examples

```
curl http://morbo.dev2000.com:8018/connection/features
```

```
{"featureInfoList":[{"featureId":"activations","version":1}, {"featureId":"statistics","version":1}, {"featureId":"configuration","version":1}, {"featureId":"connection","version":1}, {"featureId":"directories","version":1}, {"featureId":"interactions","version":1}, {"featureId":"licenses","version":1}, {"featureId":"messaging","version":1}, {"featureId":"queues","version":1}, {"featureId":"session","version":1}, {"featureId":"status","version":1}, {"featureId":"wfm","version":1}]}
```

```
curl http://morbo.dev2000.com:8018/icws/connection -H "Accept-Language:en-us" -d
```

```
"{\"__type\":\"urn:inin.com:connection:icAuthConnectionRequestSettings\",\"applicationName\":\"Example\",\"userID\":\"kevin.glinski\",\"password\":\"1234\"}"
```

```
{"csrfToken":"WA1rZXZpbi5nbGluc2tpV0V4YW1wbGVYJDlyNjU0NDEwLTU4ZjltNDkyZC05ZDE4LTU0ZDhkZTcwOWM5OFgOMTcyLjE3LjEwMC4xMjg=", "sessionId":"857584001", "alternateHostList":["Morbo.DEV2000.com"], "userID":"kevin.glinski", "userDisplayName":"Kevin Glinski"}
```

```
curl http://morbo.dev2000.com:8018/icws/857584001/status/user-statuses/kevin.glinski -H "ININ-ICWS-CSRF-
```

```
Token:WA1rZXZpbi5nbGluc2tpV0V4YW1wbGVYJDlyNjU0NDEwLTU4ZjltNDkyZC05ZDE4LTU0ZDhkZTcwOWM5OFgOMTcyLjE3LjEwMC4xMjg=" -H "cookie:icws_857584001=62791157-df5b-47db-9f8f-2fe20e5d5035; Path=/icws/857584001"
```

```
{"userId":"kevin.glinski","statusId":"Available, No
```

```
ACD","statusChanged":"20140505T153146Z","icServers":[],"stations":[],"loggedIn":false,"onPhone":false,"onPhoneChanged":"20140503T014837Z"}
```