

# API Endpoints Reference

## Overview

The Genesys Cloud API provides REST endpoints for managing contacts, conversations, users, and other platform resources. This reference covers the most commonly used endpoints for integration work.

**Base URL:** `https://api.mypurecloud.com/api/v2`

**Authentication:** OAuth 2.0 Bearer Token (see Chapter 11: OAuth Client Management)

---

## Contacts API

### Get All Contacts

**Endpoint:** `GET /contacts`

**Description:** Retrieve a paginated list of contacts.

**Query Parameters:**

- `pageSize` (integer, optional): Contacts per page (default: 25, max: 100)
- `pageNumber` (integer, optional): Page number (default: 1)
- `sortOrder` (string, optional): `asc` or `desc` (default: `asc`)

**Example Request:**

```
curl -X GET "https://api.mypurecloud.com/api/v2/contacts?pageSize=50&pageNumber=1" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

**Example Response:**

```
{
  "entities": [
    {
      "id": "contact-123",
      "firstName": "John",
      "lastName": "Doe",
      "email": "john.doe@example.com",
      "phoneNumbers": [
        {
          "number": "+15551234567",
          "type": "MOBILE"
        }
      ],
      "externalOrganization": {
        "id": "org-456",
        "name": "Acme Corp"
      },
      "externalId": "sf-contact-789",
      "dateCreated": "2026-03-14T10:30:00Z",
      "dateModified": "2026-03-14T10:30:00Z"
    }
  ],
  "pageNumber": 1,
  "pageSize": 50,
  "total": 2500
}
```

---

# Create a Contact

**Endpoint:** `POST /contacts`

**Description:** Create a new contact.

**Request Body:**

```
{
  "firstName": "Jane",
  "lastName": "Smith",
  "email": "jane.smith@example.com",
```

```
"phoneNumbers": [  
  {  
    "number": "+15559876543",  
    "type": "WORK"  
  }  
],  
"externalOrganization": {  
  "id": "org-789",  
  "name": "Tech Solutions Inc"  
},  
"externalId": "sf-contact-001"  
}
```

### Required Fields:

- `firstName` (string): Contact's first name
- `lastName` (string): Contact's last name

### Optional Fields:

- `email` (string): Email address
- `phoneNumbers` (array): Phone number objects with `number` and `type`
- `externalOrganization` (object): Org reference with `id` and `name`
- `externalId` (string): External system ID (for deduplication)

### Example Response:

```
{  
  "id": "contact-999",  
  "firstName": "Jane",  
  "lastName": "Smith",  
  "email": "jane.smith@example.com",  
  "phoneNumbers": [  
    {  
      "number": "+15559876543",  
      "type": "WORK"  
    }  
  ],  
  "dateCreated": "2026-03-14T11:00:00Z"  
}
```

# Update a Contact (Full)

**Endpoint:** `PUT /contacts/{contactId}`

**Description:** Replace entire contact record.

**Example Request:**

```
curl -X PUT "https://api.mypurecloud.com/api/v2/contacts/contact-999" \  
-H "Authorization: Bearer {ACCESS_TOKEN}" \  
-H "Content-Type: application/json" \  
-d '{  
  "firstName": "Jane",  
  "lastName": "Smith",  
  "email": "jane.smith.new@example.com"  
}'
```

---

# Update a Contact (Partial)

**Endpoint:** `PATCH /contacts/{contactId}`

**Description:** Partially update contact (new in March 2026). Reduces risk of data overwrites.

**Example Request:**

```
curl -X PATCH "https://api.mypurecloud.com/api/v2/contacts/contact-999" \  
-H "Authorization: Bearer {ACCESS_TOKEN}" \  
-H "Content-Type: application/json" \  
-d '{  
  "email": "jane.smith.new@example.com",  
  "phoneNumbers": [  
    {  
      "number": "+15551111111",  
      "type": "MOBILE"  
    }  
  ]  
}'
```

# Search for Contacts by Phone Number

**Endpoint:** GET /contacts/search

## Query Parameters:

- `q` (string): Search query (phone number, email, name)

## Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/contacts/search?q=%2B15551234567" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

## Example Response:

```
{  
  "results": [  
    {  
      "id": "contact-123",  
      "firstName": "John",  
      "lastName": "Doe",  
      "phoneNumbers": [  
        {  
          "number": "+15551234567",  
          "type": "MOBILE"  
        }  
      ]  
    }  
  ],  
  "total": 1  
}
```

---

# Delete a Contact

**Endpoint:** DELETE /contacts/{contactId}

**Description:** Remove a contact.

## Example Request:

```
curl -X DELETE "https://api.mypurecloud.com/api/v2/contacts/contact-999" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

**Response:** 204 No Content (success)

---

# Conversations API

## Get Recent Conversations

**Endpoint:** GET /conversations

### Query Parameters:

- `pageSize` (integer, optional): Conversations per page (default: 25, max: 100)
- `pageNumber` (integer, optional): Page number (default: 1)

### Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/conversations?pageSize=50" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

### Example Response:

```
{  
  "entities": [  
    {  
      "id": "conversation-111",  
      "conversationType": "phone",  
      "participants": [  
        {  
          "id": "user-agent-1",  
          "name": "Agent Smith",  
          "purpose": "agent",  
          "state": "connected"  
        },  
        {  
          "id": "customer-call",  
          "name": "John Doe",
```

```
    "purpose": "customer",
    "state": "disconnected"
  }
],
"startTime": "2026-03-14T09:15:00Z",
"endTime": "2026-03-14T09:25:00Z"
}
],
"pageNumber": 1,
"pageSize": 50,
"total": 500
}
```

# Get Conversation Detail

**Endpoint:** `GET /conversations/{conversationId}`

**Description:** Get full details including participants, recordings, transcripts.

## Query Parameters:

- `expand` (string, optional): Comma-separated fields to expand. Options: `recordings`, `transcript`, `participants`

## Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/conversations/conversation-111?expand=recordings,transcript" \
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

## Example Response:

```
{
  "id": "conversation-111",
  "conversationType": "phone",
  "participants": [
    {
      "id": "user-agent-1",
      "name": "Agent Smith",
      "purpose": "agent"
    }
  ]
}
```

```
  },
  {
    "id": "customer-call",
    "name": "John Doe",
    "purpose": "customer"
  }
],
"recordings": [
  {
    "id": "recording-222",
    "state": "AVAILABLE",
    "durationMilliseconds": 600000,
    "conversationId": "conversation-111",
    "mediaUris": {
      "download": "https://..."
    }
  }
],
"transcript": {
  "id": "transcript-333",
  "displayName": "conversation-111 Transcript",
  "dateCreated": "2026-03-14T09:30:00Z",
  "status": "AVAILABLE"
}
}
```

## Get Recording Media URL

**Endpoint:** `GET /recordings/{recordingId}/media`

**Description:** Get download URL for a recording (expires in 1 hour).

### Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/recordings/recording-222/media" \  
-H "Authorization: Bearer {ACCESS_TOKEN}" \  
-L
```

**Response:** 307 redirect to presigned S3 URL

---

# Search Conversations

**Endpoint:** `POST /conversations/search`

**Description:** Advanced search using Genesys query syntax.

## Request Body:

```
{
  "types": ["phone"],
  "query": "select * where conversations.participants.emails contains 'john.doe@example.com' and startTime >= '2026-03-01'",
  "pageSize": 25,
  "pageNumber": 1
}
```

## Example Response:

```
{
  "results": [
    {
      "conversation": {
        "id": "conversation-111",
        "conversationType": "phone",
        "startTime": "2026-03-14T09:15:00Z",
        "endTime": "2026-03-14T09:25:00Z"
      }
    }
  ],
  "pageNumber": 1,
  "pageSize": 25,
  "total": 15
}
```

---

# Users API

# Get User Details with Presence

**Endpoint:** `GET /users/{userId}`

## Query Parameters:

- `expand` (string, optional): Fields to expand. Options: `presence`, `locations`, `routingStatus`, `availability`

## Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/users/user-123?expand=presence,routingStatus" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

## Example Response:

```
{  
  "id": "user-123",  
  "name": "Agent Smith",  
  "email": "agent.smith@company.com",  
  "presence": {  
    "id": "presence-123",  
    "definition": {  
      "id": "ON_QUEUE",  
      "name": "On Queue"  
    },  
    "primary": true,  
    "source": "USER"  
  },  
  "routingStatus": {  
    "status": "ON_QUEUE",  
    "statusMessage": "Available"  
  }  
}
```

---

# Update User's Presence/Availability

**Endpoint:** `PATCH /users/{userId}`

**Description:** Update agent's presence/availability status.

## Request Body:

```
{
  "routingStatus": {
    "status": "OFF_QUEUE",
    "statusMessage": "In Training"
  }
}
```

## Example Request:

```
curl -X PATCH "https://api.mypurecloud.com/api/v2/users/user-123" \
  -H "Authorization: Bearer {ACCESS_TOKEN}" \
  -H "Content-Type: application/json" \
  -d '{
    "routingStatus": {
      "status": "OFF_QUEUE",
      "statusMessage": "Break"
    }
  }'
```

## Valid Status Values:

- `ON_QUEUE` - Ready to receive interactions
- `OFF_QUEUE` - Not available for interactions
- `IDLE` - No activity
- `ON_A_CALL` - Currently on a call
- `CUSTOM` - Custom status (use `statusMessage`)

---

# Get All Users in a Queue

**Endpoint:** `GET /queues/{queueId}/members`

## Query Parameters:

- `pageSize` (integer, optional): Members per page (default: 25, max: 100)
- `pageNumber` (integer, optional): Page number (default: 1)
- `sortOrder` (string, optional): `asc` or `desc`

## Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/queues/queue-456/members?pageSize=50" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

### Example Response:

```
{  
  "entities": [  
    {  
      "id": "user-123",  
      "name": "Agent Smith",  
      "email": "agent.smith@company.com",  
      "state": "ACTIVE"  
    },  
    {  
      "id": "user-124",  
      "name": "Agent Jones",  
      "email": "agent.jones@company.com",  
      "state": "ACTIVE"  
    }  
  ],  
  "pageNumber": 1,  
  "pageSize": 50,  
  "total": 2  
}
```

# Search Users

**Endpoint:** `GET /users/search`

### Query Parameters:

- `q` (string): Search query (name, email, phone)
- `pageSize` (integer, optional): Results per page
- `pageNumber` (integer, optional): Page number

### Example Request:

```
curl -X GET "https://api.mypurecloud.com/api/v2/users/search?q=smith&pageSize=25" \  
-H "Authorization: Bearer {ACCESS_TOKEN}"
```

---

# Common Query Parameters

## Pagination

All list endpoints support:

- `pageSize` (1-100): Items per page
- `pageNumber` (starts at 1): Which page

**Example:** `?pageSize=50&pageNumber=2` gets items 51-100

## Sorting

- `sortBy` (string): Field to sort by (varies by endpoint)
- `sortOrder` (string): `asc` or `desc`

**Example:** `?sortBy=dateCreated&sortOrder=desc` (newest first)

## Expansion

Use `expand` to include related data without extra API calls:

```
GET /conversations/conversation-111?expand=recordings,transcript,participants
```

---

## Rate Limits

All endpoints are subject to Genesys Cloud's rate limiting:

- Standard: 600 requests per minute per organization
  - Some endpoints may have different limits
  - Check response headers: `X-Rate-Limit-Limit`, `X-Rate-Limit-Remaining`, `X-Rate-Limit-Reset`
-

# Errors

All endpoints return standard HTTP status codes:

Code	Meaning	Action
200	Success	Continue
201	Created	Item created successfully
204	No Content	Success (no response body)
400	Bad Request	Fix request parameters
401	Unauthorized	Token invalid or expired
403	Forbidden	Insufficient permissions
404	Not Found	Resource doesn't exist
429	Rate Limited	Wait before retrying
500	Server Error	Retry after delay

## Best Practices

1. **Use Pagination:** Always use `pageSize` and `pageNumber` for list endpoints
2. **Expand Efficiently:** Only expand data you need
3. **Handle Pagination:** Don't assume all data fits in one page
4. **Check Status Codes:** Always handle errors appropriately
5. **Use External IDs:** Link records across systems with `externalId`
6. **Respect Rate Limits:** Monitor `X-Rate-Limit` headers

## Related Topics

- Chapter 11: Error Handling & Retry Strategy
- Chapter 11: Rate Limiting & Throttling
- Chapter 11: OAuth Client Management
- Chapter 5: Data Actions (using these APIs in flows)

Revision #1

Created 15 March 2026 00:41:35 by Cesar Gzz

Updated 15 March 2026 00:41:49 by Cesar Gzz