



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Web API Developer Guide

Genesys Engage Digital (eServices) 8.5.0

Table of Contents

Web API Developer Guide	3
About the Chat Service	4
Chat REST API Samples	6
About the E-mail Service	9
E-mail REST API Samples	10

Web API Developer Guide

The eServices Web API Server enables you to write client applications that support chat and e-mail interactions.

These operations are performed by using REST API calls with a Web Application Server—such as Tomcat, Jetty, WebSphere, or WebLogic.

Architecture

Although built using the Genesys Platform SDK version 8.1.4, you do not need to use our SDKs to build a working client.

Genesys now provides developers with the capability to build applications that can interact with the Genesys Web API Server through a simple REST API, rather than requiring the use of a complicated software framework.

What is REST?

REST stands for Representational State Transfer, and is stateless, and Genesys uses a Hypertext Transfer Protocol (HTTP) REST API.

REST uses a client-server model, with cacheable communications protocol.

REST uses **resources** and **actions** instead of **methods** and **data types**. Resources are URLs that reference the object where you want to perform an action.

The Web API Server enables you to access a RESTful API to perform common functions such as conducting a chat session or sending an e-mail message.

You can learn more about REST at [Wikipedia](#) and other sites.

What is in this guide?

- **About the Chat Service**—This section describes the chat portion of the Genesys Web API Server REST API and provides guidance for developers building chat-related client applications.
- **About the Email Service**—This section describes the e-mail portion of the Genesys Web API Server REST API and provides guidance for developers building e-mail-related client applications.

About the Chat Service

Overview

This section describes the chat portion of the Genesys Web API Server REST API and provides guidance for developers building chat-related client applications.

To assist developers, we provide a few [Chat REST API Samples](#).

Life Cycle of a Chat Session

The chat service is stateless, so in order to know whether a chat interaction is active, we have a requirement to use long polling and do this with the [GetMessages](#) operation.

Once a chat session starts, your application must periodically refresh by sending a [GetMessages](#) request to keep the session alive.

The refresh frequency is dependent on your application. However, you can use the `flex-disconnect-timeout` configuration option in Chat Server to specify the maximum timeout between refresh requests.

Important

Ensure that the interval in which your application calls the [GetMessages](#) request falls within the `flex-disconnect-timeout` value.

This is a Chat-specific requirement.

Chat Server Status Codes

The Web API Server Chat Service now includes status codes in the HTTP response. Here is a list of the codes and their meanings:

- **statusCode 0**—The attempted communication with Chat Server was successful.
- **statusCode 1**—If Chat Server sends an error response that corresponds to a recoverable error, then there was a recoverable error. Otherwise, there was a connection error with Chat Server.
- **statusCode 2**—There was a non-recoverable error.

Status Codes and Outcomes

Status Code	chatEnded=true/false	Does the Chat Client Consider That The Session Has Ended?	Comment
Successful Outcomes			
0	false	No	Everything is fine
0	true	Yes	The chat has ended; terminate the session
Unsuccessful Outcomes			
1	false	No	The request was unsuccessful; retry
1	true	N/A	This case isn't possible, as the request was unsuccessful and the last successful chatEnded flag is inherited
2	true or false	Yes	Unrecoverable error; terminate the session

Chat REST API Samples

Overview

The following are sample code snippets for chat interactions using the REST API of the Web API Server.

A complete list of Chat interaction operations can be found in the [Web API Server API Reference](#).

Request Chat

[+] Sample

```
CreateSessionDetails sessionDetails = new CreateSessionDetails();
sessionDetails.setTenantName("Environment");
sessionDetails.setFirstName("John");
sessionDetails.setLastName("Doe");
sessionDetails.setNickname("Nickname");
sessionDetails.setEmailAddress("email@address");
sessionDetails.setSubject("Subject");

ObjectMapper mapper = new ObjectMapper();
String sessionDetailsString = mapper.writeValueAsString(sessionDetails);

Client client = new Client();
WebResource webResource = client.resource("http://localhost:8080/webapi/api/v2/
chats");

ClientResponse response =
webResource.type(MediaType.APPLICATION_JSON).post(ClientResponse.class, sessionDetailsString);

if (response.getStatus() != 200)
{
    throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());
}

String json = response.getEntity(String.class);
```

Send Message

[+] Sample

```
ChatOperationDetails operationDetails = new ChatOperationDetails();
operationDetails.setOperationName("SendMessage");
operationDetails.setAlias("117");
operationDetails.setSecureKey("very_secure");
operationDetails.setUserId("user1");
operationDetails.setTenantName("Environment");
```

```
    ObjectMapper mapper = new ObjectMapper();
    String operationDetailsString = mapper.writeValueAsString(operationDetails);

    Client client = new Client();
    WebResource webResource = client.resource("http://localhost:8080/webapi/api/v2/chats/0000KaA0C8XH03D1");

    ClientResponse response =
webResource.type(MediaType.APPLICATION_JSON).post(ClientResponse.class,
operationDetailsString);

    if (response.getStatus() != 200)
    {
        throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());
    }

    String json = response.getEntity(String.class);
```

Update User Data

[+] Sample

```
    ChatOperationDetails operationDetails = new ChatOperationDetails();
    operationDetails.setOperationName("UpdateUserData");
    operationDetails.setAlias("117");
    operationDetails.setSecureKey("very_secure");
    operationDetails.setUserId("user1");
    operationDetails.setTenantName("Environment");

    Map<String, String> userData = new HashMap<>();
    userData.put("key1", "value1");
    operationDetails.setUserData(userData);

    ObjectMapper mapper = new ObjectMapper();
    String operationDetailsString = mapper.writeValueAsString(operationDetails);

    Client client = new Client();
    WebResource webResource = client.resource("http://localhost:8080/webapi/api/v2/chats/0000KaA0C8XH03D1");

    ClientResponse response =
webResource.type(MediaType.APPLICATION_JSON).post(ClientResponse.class,
operationDetailsString);

    if (response.getStatus() != 200)
    {
        throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());
    }

    String json = response.getEntity(String.class);
```

Chat Complete

[+] Sample

```
ChatOperationDetails operationDetails = new ChatOperationDetails();
operationDetails.setOperationName("Complete");
operationDetails.setAlias("117");
operationDetails.setSecureKey("very_secure");
operationDetails.setUserId("user1");
operationDetails.setTenantName("Environment");

ObjectMapper mapper = new ObjectMapper();
String operationDetailsString = mapper.writeValueAsString(operationDetails);

Client client = new Client();
WebResource webResource = client.resource("http://localhost:8080/webapi/api/v2/chats/
0000KaA0C8XH03D1");

ClientResponse response =
webResource.type(MediaType.APPLICATION_JSON).post(ClientResponse.class,
operationDetailsString);

if (response.getStatus() != 200)
{
    throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());
}

String json = response.getEntity(String.class);
```


About the E-mail Service

Overview

Important

All APIs within Web API, including the existing Chat and Email APIs, are being consolidated under Genesys Mobile Services (GMS).

Please refer to Genesys Mobile Services documentation for more information:

- [Email API](#)

This section describes the e-mail portion of the Genesys Web API Server REST API and provides guidance for developers building e-mail-related client applications.

E-mail REST API Samples

Overview

The following is a sample code snippet for an e-mail interaction using the REST API of the Web API Server.

Send E-mail Interaction

[+] Sample

```
FormDataMultiPart formDataMultiPart = new FormDataMultiPart();
formDataMultiPart.field("firstName", "First");
formDataMultiPart.field("lastName", "Last");
formDataMultiPart.field("tenantName", "Environment");
formDataMultiPart.field("fromAddress", "from@address");
formDataMultiPart.field("subject", "subject");
formDataMultiPart.field("text", "very long email body");

final String value1 = "Hello World!";
FormDataContentDisposition dispo =
FormDataContentDisposition.name("files").fileName("test.txt").size(value1.getBytes().length).build();
FormDataBodyPart bodyPart = new FormDataBodyPart(dispo, value1);
formDataMultiPart.bodyPart(bodyPart);

final String value2 = "Goodbye World!";
dispo =
FormDataContentDisposition.name("files").fileName("test2.txt").size(value2.getBytes().length).build();
bodyPart = new FormDataBodyPart(dispo, value2);
formDataMultiPart.bodyPart(bodyPart);

Client client = new Client();
WebResource webResource = client.resource("http://localhost:8080/webapi/api/v2/
emails");

ClientResponse response =
webResource.type(MediaType.MULTIPART_FORM_DATA).post(ClientResponse.class, formDataMultiPart);

if (response.getStatus() != 200)
{
    throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());
}

String json = response.getEntity(String.class);
```