



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.

Workforce Management API Reference

Workforce Management API Reference

Contents

- [1 Direct downloads](#)
 - [1.1 Viewing the documentation](#)
- [2 Using this API](#)



- Developer

Learn how to use the Workforce Management (WFM) Integration API to customize the way your web application integrates with Workforce Management.

Related documentation:

-

The Workforce Management Integration API Reference documentation is available for download from this page.

Important

This feature requires special provisioning to be performed by Genesys. For information about using this feature in your deployment, contact your Genesys representative.

Direct downloads

Find your release number in the table below and download the file to access the most recent Workforce Management Integration API Reference documentation.

Workforce Management Integration API Reference Documentation

Release	Direct Download	Last Updated	File Size
WFM 8.5 Integration API Reference Documentation	WFM_API_docs_85.zip	September 27, 2022	27.5 MB

Viewing the documentation

After downloading and extracting the zip file, go to the **docs** folder and open the **index.html** file in your browser.

Using this API

To use this API, you must first do the following:

-
- Use OAuth REST API (OAuth2) to authenticate the user and obtain an access token.
 - Once the access token is obtained, use the following to obtain WFM user information, including userId, user type, and other details:
WFM REST API method GET /wfm/api/v3/system/userinfo

Important

The /userinfo call can return a 403 (Forbidden) error code if an authenticated user is not provisioned in WFM.

The **LocateServerLogin** and **OpenSessionLogin** SOAP API methods are not supported for this API. Instead, you must use the corresponding **LocateServer** and **OpenSession** SOAP API methods. Otherwise, this SOAP API works similar to the API for WFM on-premises except that each request will have the access token attached as a **Bearer** token in the **Authorization** header.

This example shows an access token being attached to the request:

```
private static void setAccessToken(T port, String accessToken)
{
    BindingProvider bp = (BindingProvider) port;
    java.util.Map context = bp.getRequestContext();

    Map > requestHeaders = (Map >) context.get(MessageContext.HTTP_REQUEST_HEADERS);

    if (requestHeaders == null) {
        requestHeaders = new HashMap > ();
        context.put(MessageContext.HTTP_REQUEST_HEADERS, requestHeaders);
    }

    requestHeaders.put("Authorization", Collections.singletonList("Bearer " + accessToken));
    context.put(MessageContext.HTTP_REQUEST_HEADERS, requestHeaders);
}

...
WFMLocatorService800Soap locator = getService800Soap(WFMLocatorService800Soap.class,
"/LocatorService/WFMLocatorService800.wsdl", "WFMLocatorService800",
"WFMLocatorService800Soap", "https://");

setAccessToken(locator, access_token); // Attach access token to the request

locator.LocateServer(...)
```