

# Java Client Example

The following code is an example on how the REST API can be accessed from Java Code.

The code can be run with any java 11 JDK. For example: <https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/downloads-list.html>

```
package com.mycompany.fews.webservices;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.StringWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public final class RestServiceClient {

    private RestServiceClient() {
    }

    public static void main(String[] args) throws IOException {

        // Example on how to call a GET request with parameters.
        filtersExample();

        // Example on how to run a workflow with a POST request with parameters.
        runTaskExample();

        // Example on how to get warm state times and download a warm state file.
        getWarmStateTimes();
        downloadWarmState();

    }

    private static void getWarmStateTimes() throws IOException {
        List<RequestParameter> requestParameters = new ArrayList<>();
        requestParameters.add(new RequestParameter("moduleInstanceIds", "ModuleInstanceId"));
        requestParameters.add(new RequestParameter("documentFormat", "PI_JSON"));
        String warmStatesUrl = "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/warmstates/times";
        HttpURLConnection connection = get(warmStatesUrl, requestParameters);
        if (connection.getResponseCode() == 200) {
            String response = getHttpResponse(connection);
            // the response contains the warmstate files in PI_JSON format.
            // JSON can be parsed with for example the jackson json library: https://github.com/FasterXML
/jackson.
            System.out.println(response);
        } else {
            // Handle error.
            String errorResponse = getHttpResponse(connection);
            System.out.println(errorResponse);
        }
    }

    private static void downloadWarmState() throws IOException {
        List<RequestParameter> requestParameters = new ArrayList<>();
        requestParameters.add(new RequestParameter("moduleInstanceId", "ModuleInstanceId"));
        requestParameters.add(new RequestParameter("stateTime", "2023-01-24T08:00:00Z"));
        String warmStatesUrl = "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/warmstates";
```

```

        HttpURLConnection connection = get(warmStatesUrl, requestParameters);
        if (connection.getResponseCode() == 200) {
            String tmpdir = Files.createTempDirectory("warmstates").toFile().getAbsolutePath();
            File zipFile = new File(tmpdir, "warmstate.zip");
            downloadFile(connection, zipFile);
            System.out.printf("The Warm State file can be found at: %s\n", zipFile.getAbsolutePath());
        } else {
            // Handle error.
            String errorResponse = getHttpResponse(connection);
            System.out.println(errorResponse);
        }
    }

    private static void filtersExample() throws IOException {
        List<RequestParameter> requestParameters = new ArrayList<>();
        requestParameters.add(new RequestParameter("documentFormat", "PI_XML"));
        String filtersUrl = "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/filters";
        HttpURLConnection connection = get(filtersUrl, requestParameters);
        if (connection.getResponseCode() == 200) {
            String response = getHttpResponse(connection);
            // the response contains the filters in PI_XML format.
            System.out.println(response);
        } else {
            // Handle error.
            String errorResponse = getHttpResponse(connection);
            System.out.println(errorResponse);
        }
    }

    private static void runTaskExample() throws IOException {
        List<RequestParameter> requestParameters = new ArrayList<>();
        requestParameters.add(new RequestParameter("workflowId", "Import_Forecasts"));
        String runTasksUrl = "http://localhost:8080/FewsWebServices/rest/fewspiservice/v1/runtask";
        List<RequestParameter> bodyParameters = new ArrayList<>();
        HttpURLConnection connection = post(runTasksUrl, requestParameters, bodyParameters);
        if (connection.getResponseCode() == 200) {
            String response = getHttpResponse(connection);
            // the response contains the task id.
            System.out.println(response);
        } else {
            // Handle error.
            String errorResponse = getHttpResponse(connection);
            System.out.println(errorResponse);
        }
    }

    public static HttpURLConnection post(String urlString, List<RequestParameter> requestParameters,
        List<RequestParameter> bodyParameters) throws IOException {
        URL url = createUrl(urlString, requestParameters);
        var connection = (HttpURLConnection) url.openConnection();
        connection.setRequestMethod("POST");
        connection.setDoOutput(true);
        connection.setDoInput(true);
        try (StringWriter stringWriter = new StringWriter(16)) {
            for (RequestParameter param : bodyParameters) {
                stringWriter.append(URLEncoder.encode(param.getKey(), StandardCharsets.UTF_8));
                stringWriter.append("=");
                stringWriter.append(URLEncoder.encode(param.getValue(), StandardCharsets.UTF_8));
                stringWriter.append("&");
            }
            connection.getOutputStream().write(stringWriter.toString().getBytes());
        }
        connection.connect();
        return connection;
    }

    public static HttpURLConnection get(String urlString, List<RequestParameter> requestParameters) throws
        IOException {
        URL url = createUrl(urlString, requestParameters);
        var connection = (HttpURLConnection) url.openConnection();
    }

```

```

        configureTimeout(connection);
        return connection;
    }

    private static void configureTimeout(URLConnection connection) {
        var connectionTimeout = 60_000;
        connection.setConnectTimeout(connectionTimeout);
        var readTimeout = 60_000;
        connection.setReadTimeout(readTimeout);
        connection.setDoInput(true);
    }

    private static URL createUrl(String urlString, List<RequestParameter> requestParameters) throws
    MalformedURLException {
        StringBuilder urlBuilder = new StringBuilder(16);
        urlBuilder.append(urlString);
        if (!requestParameters.isEmpty()) urlBuilder.append("?");
        for (var requestParam: requestParameters) {
            urlBuilder.append(URLEncoder.encode(requestParam.getKey(), StandardCharsets.UTF_8));
            urlBuilder.append("=");
            urlBuilder.append(URLEncoder.encode(requestParam.getValue(), StandardCharsets.UTF_8));
            urlBuilder.append("&");
        }
        return new URL(urlBuilder.toString());
    }

    public static String getHttpResponse(URLConnection connection) throws IOException {
        try (var bufferedReader = new BufferedReader(new InputStreamReader(connection.getResponseCode() <
        HttpURLConnection.HTTP_BAD_REQUEST ? connection.getInputStream() : connection.getErrorStream(),
        StandardCharsets.UTF_8))) {
            return bufferedReader.lines().collect(Collectors.joining("\n"));
        }
    }

    private static void downloadFile(URLConnection connection, File zipFile) throws IOException {
        try (OutputStream outputStream = new FileOutputStream(zipFile)) {
            try (var inputStream = connection.getInputStream()) {
                byte[] buffer = new byte[1024];
                int bytesRead;
                while ((bytesRead = inputStream.read(buffer)) != -1) {
                    outputStream.write(buffer, 0, bytesRead);
                }
            }
        }
    }

    private static final class RequestParameter {
        private final String key;
        private final String value;

        private RequestParameter(String key, String value) {
            this.key = key;
            this.value = value;
        }

        private String getKey() {
            return key;
        }

        private String getValue() {
            return value;
        }
    }
}

```