Kippy API Integration with JWT

Nauman Khan

16-Feb-2024

Intro

This paper provides details of a proof of concept that invokes a kippy data endpoint using basic authentication and a JWT token. The outcome successfully showed how kippy users could dynamically push data from internal systems to update a KPI actual. More details are provided below.

Background

What is Basic Authentication?

Basic Authentication is a simple and widely used method for authenticating users on the web. It is a part of the HTTP protocol and involves sending a username and password in the request headers. The basic idea is to include a "Authorization" header in the HTTP request, which contains the word "Basic" followed by a space and a base64-encoded string of "username:password".

Here's a basic example of how it works:

1. The client (such as a web browser or a software application) sends an HTTP request to a server.
2. The server responds with a 401 Unauthorized status code, indicating that authentication is required.
3. The client includes an "Authorization" header in the request with the credentials encoded in base64.

For example:

Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=

In this example, "dXNlcm5hbWU6cGFzc3dvcmQ=" is the base64-encoded form of "username:password".

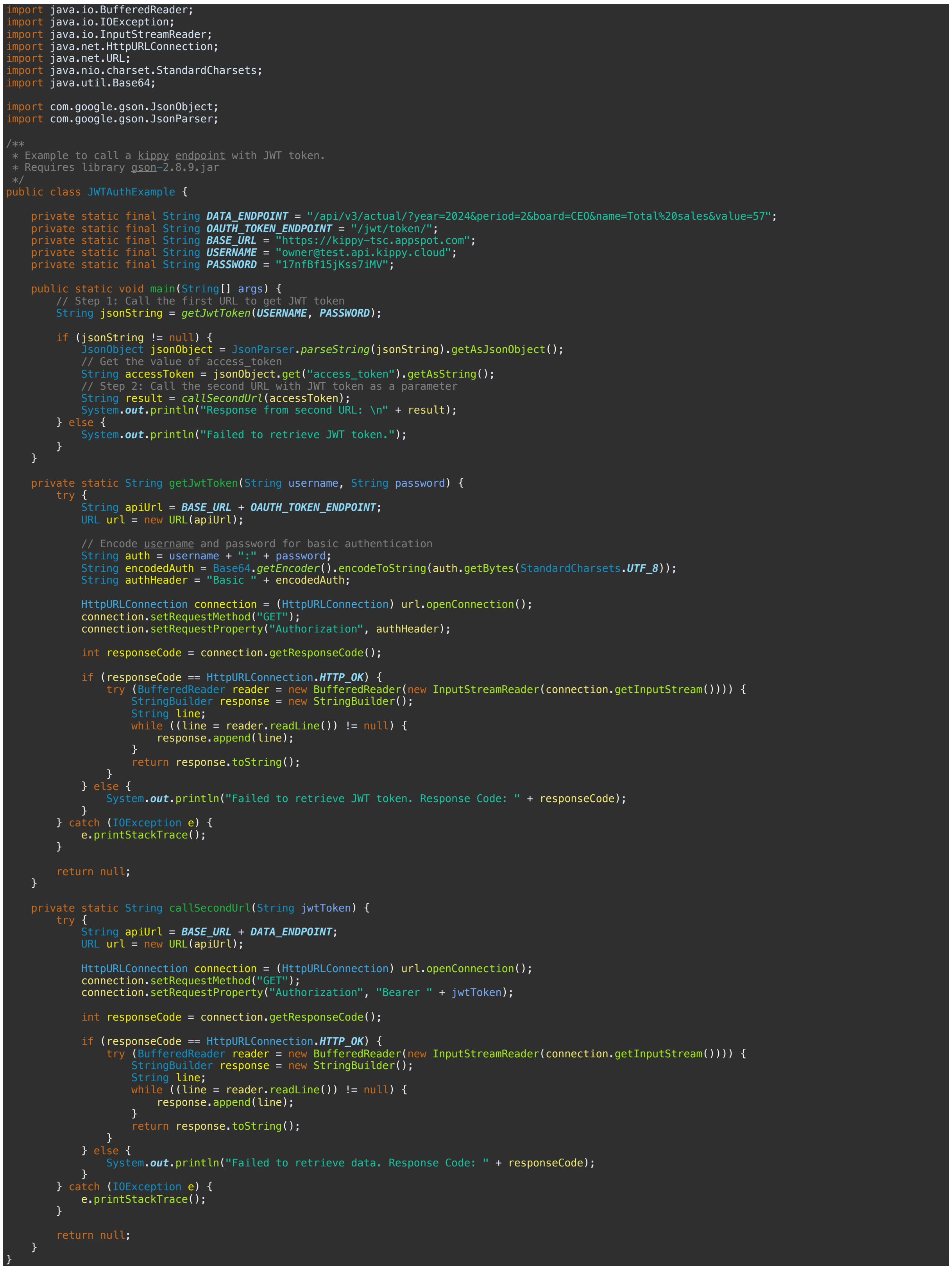
What is a JWT token?

JWT stands for JSON Web Token, and it is a compact, URL-safe means of representing claims to be transferred between two parties. JWTs are often used for authentication and authorization in web applications.

JWTs are commonly used for authentication by generating a token on the server side, sending it to the client upon successful login, and having the client include the token in the headers of subsequent requests. The server can then verify the integrity and authenticity of the token using the stored secret or public key.

Approach

For the POC, the following code was used to call a data endpoint and change a KPI actual value.



```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.Base64;

import com.google.gson.JsonObject;
import com.google.gson.JsonParser;

/**
 * Example to call a kippy endpoint with JWT token.
 * Requires library gson-2.8.9.jar
 */
public class JWTAuthExample {

    private static final String DATA_ENDPOINT = "/api/v3/actual/?year=2024&period=2&board=CEO&name=Total%20sales&value=57";
    private static final String OAUTH_TOKEN_ENDPOINT = "/jwt/token/";
    private static final String BASE_URL = "https://kippy-tsc.appspot.com";
    private static final String USERNAME = "owner@test.api.kippy.cloud";
    private static final String PASSWORD = "17nfBf15jKss7iMV";

    public static void main(String[] args) {
        // Step 1: Call the first URL to get JWT token
        String jsonString = getJwtToken(USERNAME, PASSWORD);

        if (jsonString != null) {
            JsonObject jsonObject = JsonParser.parseString(jsonString).getAsJsonObject();
            // Get the value of access_token
            String accessToken = jsonObject.get("access_token").getAsString();
            // Step 2: Call the second URL with JWT token as a parameter
            String result = callSecondUrl(accessToken);
            System.out.println("Response from second URL: \n" + result);
        } else {
            System.out.println("Failed to retrieve JWT token.");
        }
    }

    private static String getJwtToken(String username, String password) {
        try {
            String apiUrl = BASE_URL + OAUTH_TOKEN_ENDPOINT;
            URL url = new URL(apiUrl);

            // Encode username and password for basic authentication
            String auth = username + ":" + password;
            String encodedAuth = Base64.getEncoder().encodeToString(auth.getBytes(StandardCharsets.UTF_8));
            String authHeader = "Basic " + encodedAuth;

            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            connection.setRequestProperty("Authorization", authHeader);

            int responseCode = connection.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP_OK) {
                try (BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()))) {
                    StringBuilder response = new StringBuilder();
                    String line;
                    while ((line = reader.readLine()) != null) {
                        response.append(line);
                    }
                    return response.toString();
                }
            } else {
                System.out.println("Failed to retrieve JWT token. Response Code: " + responseCode);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }

    private static String callSecondUrl(String jwtToken) {
        try {
            String apiUrl = BASE_URL + DATA_ENDPOINT;
            URL url = new URL(apiUrl);

            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            connection.setRequestProperty("Authorization", "Bearer " + jwtToken);

            int responseCode = connection.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP_OK) {
                try (BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()))) {
                    StringBuilder response = new StringBuilder();
                    String line;
                    while ((line = reader.readLine()) != null) {
                        response.append(line);
                    }
                    return response.toString();
                }
            } else {
                System.out.println("Failed to retrieve data. Response Code: " + responseCode);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }
}
```

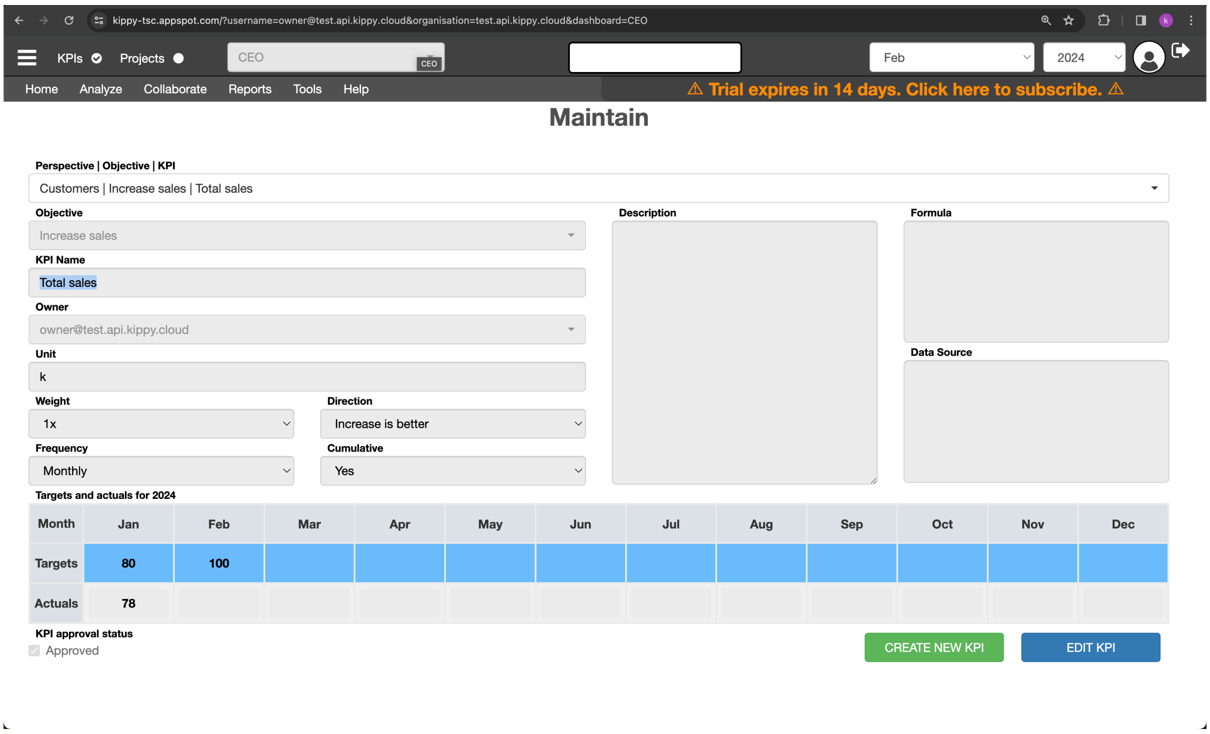
This code logs into the test.api.kippy.cloud instance with the specified username and password.

First an access token is extracted from the retrieved OAUTH_TOKEN_ENDPOINT json object.

Then the access token is passed on the header of a second call to the data endpoint.

The data endpoint will attempt to update a KPI actual named “Total sales” in the kippy team called “CEO” to the value of “57” for the second period (Feb) of the year 2024.

Below you can see the KPI as set up in the kippy instance.


kippy-tsc.appspot.com/?username=owner@test.api.kippy.cloud&organisation=test.api.kippy.cloud&dashboard=CEO
KPIs
Projects
CEO
Feb
2024
Home
Analyze
Collaborate
Reports
Tools
Help
⚠ Trial expires in 14 days. Click here to subscribe. ⚠
CREATE NEW KPI
EDIT KPI

Maintain

Perspective | Objective | KPI: Customers | Increase sales | Total sales

Objective: Increase sales

KPI Name: Total sales

Owner: owner@test.api.kippy.cloud

Unit: k

Weight: 1x

Direction: Increase is better

Frequency: Monthly

Cumulative: Yes

Description

Formula

Data Source

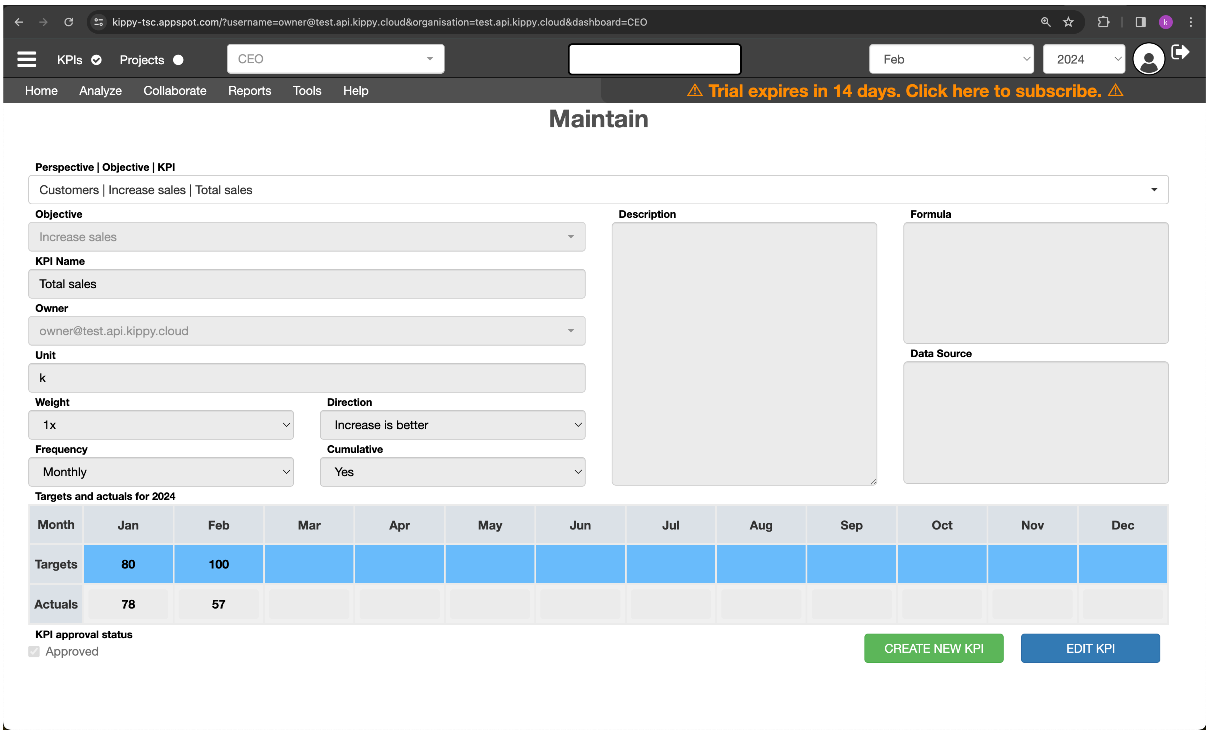
Targets and actuals for 2024

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Targets	80	100										
Actuals	78											

KPI approval status

Approved

After running the code, you can see the Feb KPI actual being updated to the value of “57”


kippy-tsc.appspot.com/?username=owner@test.api.kippy.cloud&organisation=test.api.kippy.cloud&dashboard=CEO

KPIs Projects CEO Feb 2024

Home Analyze Collaborate Reports Tools Help ⚠ Trial expires in 14 days. Click here to subscribe. ⚠

Maintain

Perspective | Objective | KPI: Customers | Increase sales | Total sales

Objective: Increase sales

KPI Name: Total sales

Owner: owner@test.api.kippy.cloud

Unit: k

Weight: 1x

Direction: Increase is better

Frequency: Monthly

Cumulative: Yes

Description:

Formula:

Data Source:

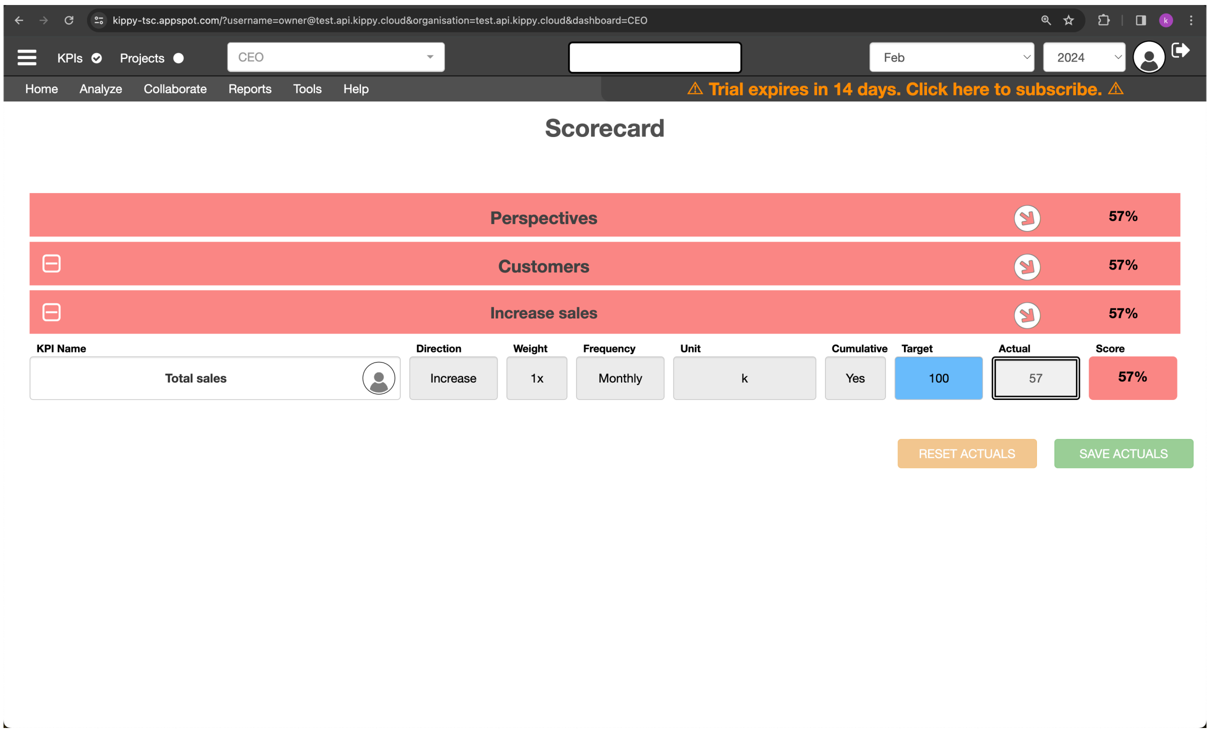
Targets and actuals for 2024

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Targets	80	100										
Actuals	78	57										

KPI approval status: Approved

CREATE NEW KPI EDIT KPI

All scores related to that updated KPI actual are then automically updated.


kippy-tsc.appspot.com/?username=owner@test.api.kippy.cloud&organisation=test.api.kippy.cloud&dashboard=CEO

KPIs Projects CEO Feb 2024

Home Analyze Collaborate Reports Tools Help ⚠ Trial expires in 14 days. Click here to subscribe. ⚠

Scorecard

Perspectives 57%

Customers 57%

Increase sales 57%

KPI Name	Direction	Weight	Frequency	Unit	Cumulative	Target	Actual	Score
Total sales	Increase	1x	Monthly	k	Yes	100	57	57%

RESET ACTUALS SAVE ACTUALS

Conclusion

KPI values can be easily updated across systems.

Other authentication mechanisms are also available. See the Wrike demo for an Oauth2 example.

Data in your internal systems can also be "pulled" from kippy using bespoke and data drive adapters.

Contact us at support@kippy.me for more details.