DISCOVERY CONSULTING

Integration Documentation – Solution Overview

Time events import from external clock vendor to SAP SuccessFactors Time Tracking integration flow example.

Introduction

This document provides the design principles for integrating a 3rd party clocking system with SAP Time Tracking Clock-In-Clock-Out module (CICO) using SAP Cloud Integration. This example lays the foundation for real-world scenarios, ensuring clarity and functionality in the integration process.

Related Information

- SAP SuccessFactors API Migration from Basic Authentication to OAuth2
- 2653173 - Generating SSH Key pair and uploading on SuccessFactors SFTP servers
- 3206348 - User authorization error when configuring the API connection through the REST service
- APIs for SAP SuccessFactors Clock In Clock Out

Business Requirement

To enable the seamless integration of time entries from 3rd party clocking systems into the SAP Time Tracking Clock-In-Clock-Out functionality.

Design Considerations & Objectives:

1. **File-Based Integration**: Leverage the capabilities of the SAP Integration Suite to accommodate vendors without APIs. This integration is an effective workaround, ensuring the process is not limited to systems with API capabilities.

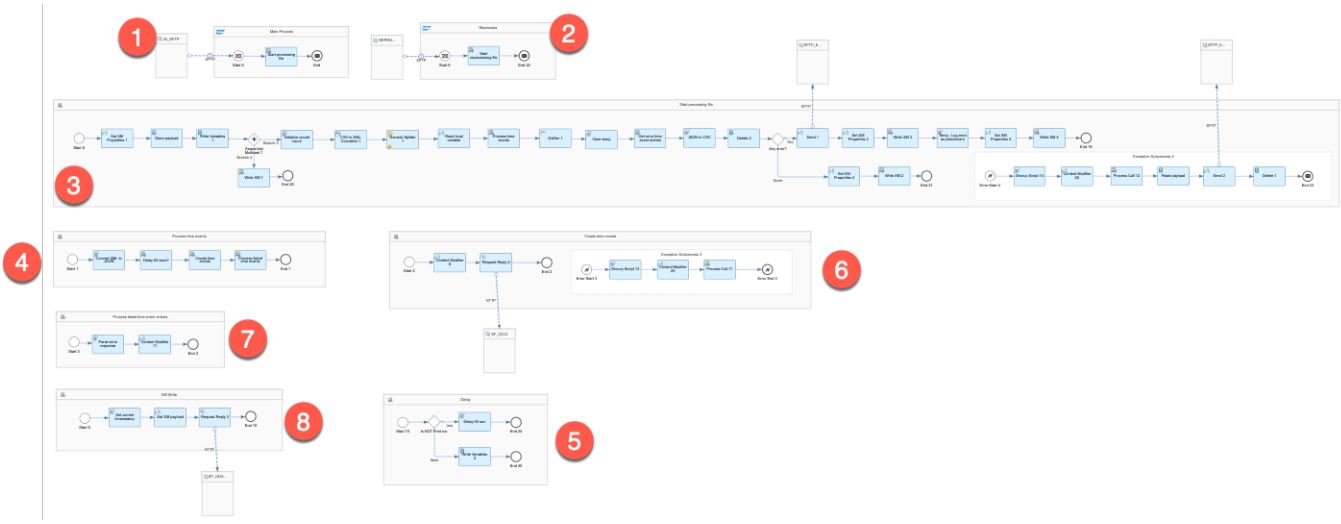
2. **Monitoring**: Despite being a file-based method, this integration incorporates SAP SuccessFactors Execution Manager to offer robust monitoring capabilities. Stakeholders can effectively track and assess data flow between the external Clock In/Clock Out system and SAP SuccessFactors Time Tracking.

3. **Error Handling**: If the file fails to process completely during integration, an error file with the failed time entries along with the error messages will be generated and placed into a designated folder for error files. This process allows for manual intervention and facilitates manual error reprocessing. The error messages return by the API will also be consolidated and posted to SAP SuccessFactors Execution Manager.

4. **Batch Processing**: For optimal efficiency and manageability of the integration, time entries are processed in batches. Each batch contains a maximum of 1,000 entries,

as this is the limit set by the SAP Time Tracking API and also recommended to limit 1 request per minute. Adhering to this constraint is essential for system optimization and compliance with the API's requirements, ensuring a successful and consistent data transfer.

SAP Cloud Integration example - iFlow process step



Step 1: Poll SFTP Server
Local integration process polling SFTP server for time event files as per scheduler.

Step 2: Poll SFTP Server – Reprocess folder.
Local integration process to poll SFTP server for reprocessing of time event files. E.g., corrected time event files after failure.

Step 3: Main Process

a) **Data Staging:** The file payload is first stored in the local datastore before being split for batch processing.
b) **Batch Processing:** The time event data are methodically processed in distinct batches. The splitter element has been configured to segment the payload into groupings, each containing 1,000 entries.
c) **Post-Processing Error Management:** After the entirety of the file payload has been processed, any errors encountered will be logged and managed by the local integration process."

Step 4: Data Transformation & Sequential Batch Processing of Time Events

a) Input File Format:

The integration accepts input data in a CSV format. Below is an illustrative example of the expected CSV structure:

```
id,assignmentId,terminalId,typeCode,timestamp
1,103192,Term1,P10,2023-07-23T08:00:00+0530
2,103192,Term1,P20,2023-07-23T12:00:00+0530
....
```

Table 1: CSV Header Fields Description

Field	Description
id	Unique identifier for the time entry
assignmentId	User Id
terminalId	Clock Terminal Identifier
typeCode	CICO Time Event Type
timestamp	Date and time of the event

b) **Target JSON Structure for REST API**

Upon successful transformation, the data extracted from the CSV file is formatted into a structured JSON payload, aligning with the API's requirements. Presented below is an exemplar of the expected JSON structure:


```json
[
    {
        "id": "1",
        "assignmentId": "103192",
        "terminalId": "Term1",
        "typeCode": "P10",
        "timestamp": "2023-07-23T08:00:00+0530"
    },
    {
        "id": "2",
        "assignmentId": "103192",
        "terminalId": "Term1",
        "typeCode": "P20",
        "timestamp": "2023-07-23T12:00:00+0530"
    }
]
```


The time event batch data is transformed into JSON format and subsequently forwarded to the subsequent local integration process.

Step 5: Recommended limit
For 1000 time events in a request, the recommendation is to send 1 request / minute.

Step 6: Create Time Events
Post the JSON-formatted time event payload to SAP SuccessFactors Time Tracking utilizing the REST API.

Step 7: Process Failed Time Events
Analyse the response obtained from the REST API and address any time events that were not successfully processed.

Step 8: Write to SAP SuccessFactors Execution Manager

Within this local integration process, activity logs are systematically written in the SAP SuccessFactors Execution Manager. This encompasses logs at the initiation of the integration, any encountered errors, and upon the successful completion of the integration run.