



SoftOne GO: API Endpoint

EmployeeActiveSchedules (Post)



Date Version: 2026-05-13

Contents

1. EmployeeActiveSchedules (Post)	3
1.1 Endpoint description	3
1.2 API Access Information	3
1.3 Read more	3
2. Request parameters	3
2.1  Authentication	3
2.2  Main Structure	4
2.3 Request sample	5
2.4 Response parameters	6
2.5 Response sample	6
3. Request examples	7
3.1 Example request X	7
3.2 Example request Y	7
3.3 [...]	7
4. Application Reference - SoftOne GO	8
5. Additional information	9
5.1 Pre-conditions / How to get started	9
5.2 Use cases	9
5.3 Best practices	9
5.4 Limitations	9
5.5 Post conditions / Expected outcome	9
5.6 Tips for the integrator	9
6. Revision history	10

1. EmployeeActiveSchedules (Post)

API Endpoint: EmployeeActiveSchedules (Post) | Import employee schedules into SoftOne

1.1 Endpoint description

The **EmployeeActiveSchedules** endpoint is used to import or update active work schedules for employees in SoftOne GO. It supports full replacement of schedules for a given date range. This is typically used to synchronize schedules from external systems into SoftOne, ensuring employees have up-to-date working hours and break periods. Existing schedule entries within the given date ranges are fully replaced by the new entries.

1.2 API Access Information

- Endpoint URL: <https://apiexternal.softone.se/Personell/Schedule/EmployeeActiveSchedules>
- Method: Post
- Authentication required: Yes

1.3 Read more

To read the full technical documentation in *Swagger*, please click this link: [EmployeeActiveSchedules](#)

2. Request parameters

2.1 Authentication

The following authentication parameters must be included in each request to securely identify the client and authorize access to the SoftOne GO API.

Name	Data type	Mandatory	Description
companyApiKey	string	Yes	Unique company identifier
connectApiKey	string	Yes	Unique integration partner identifier
Token	string	Yes	The unique authentication token for the user request

2.2 Main Structure

The table below defines the structure of the request body, including the required employee identifiers and schedule intervals.

Name	Data type	Mandatory	Description
EmployeeNr	string	Yes	Use this or EmployeeExternalCode to identify the employee.
EmployeeExternalCode	string	Yes	Use this or EmployeeNr to identify the employee.
ActiveScheduleIntervals	array of objects	Yes	List of intervals with start/stop and day schedules.
StartDate	date-time	Yes	Start date for the schedule period.
StopDate	date-time	Yes	End date for the schedule period.
ActiveScheduleDays	array of objects	Yes	List of days and their respective scheduled blocks and breaks.

2.3 Request sample

The following request JSON example shows how to register a 5-day schedule with work and break periods for one employee:

```
[
  {
    "EmployeeNr": "EMP001",
    "EmployeeExternalCode": "EXT001",
    "ActiveScheduleIntervals": [
      {
        "StartDate": "2025-06-24T00:00:00Z",
        "StopDate": "2025-06-28T00:00:00Z",
        "ActiveScheduleDays": [
          {
            "Date": "2025-06-24T00:00:00Z",
            "ActiveScheduleBlocks": [
              {
                "StartTime": "2025-06-24T08:00:00Z",
                "StopTime": "2025-06-24T12:00:00Z",
                "ShiftTypeExternalCode": "DAY",
                "HierarchicalAccountExternalCode": "DEP001",
                "AccountExternalCodeSIECostCenter": "1000",
                "AccountExternalCodeSIEProject": "PRJ001",
                "AccountExternalCodeSIEDepartment": "SALES",
                "AccountExternalCodeSIECostUnit": "UNIT01",
                "AccountExternalCodeSIEShop": "SHOP01"
              }
            ],
            "ActiveScheduleBreakBlocks": [
              {
                "StartTime": "2025-06-24T10:00:00Z",
                "StopTime": "2025-06-24T10:15:00Z",
                "BreakTypeExternalCode": "COFFEE",
                "AccountExternalCode": "ACC001"
              }
            ],
            "DayExternalCode": "TUESDAY"
          }
        ]
      }
    ]
  }
]
```

2.4 Response parameters

[Lacks documentation]

No specific response parameters or schema are documented for this endpoint. A successful response is typically indicated by an HTTP 200 status code.

2.5 Response sample

Sample of data returned in JSON format:

[Lacks documentation]

No official response example is available. Typically, a successful import does not return a detailed body.

3. Request examples

3.1 Example request X

[Provide a short explanation/description of the example + add the actual post, body and response]

POST

<https://apiexternal.softone.se/...>

BODY

[...]

3.2 Example request Y

[Provide a short explanation/description of the example + add the actual post, body and response]

POST

<https://apiexternal.softone.se/...>

BODY

3.3 [...]

4. Application Reference - SoftOne GO

This section explains how the API data relates to the UI/logic in SoftOne GO. It also describes the purpose of the properties in context of SoftOne GO. It helps new integrators connect what they see in the system with what they get via the API.

Navigate to <https://go.softone.se>

5. Additional information

[This chapter is optional. If the chapter is included, then use only those sub-chapters that is adequate for this endpoint and remove the rest]

5.1 Pre-conditions / How to get started

[What needs to be in place in SoftOne GO before being able to run a data exchange: technical prerequisites, data in GO, etc.]

5.2 Use cases

[Describe the common practices when used]

5.3 Best practices

[Describe the common practices SoftOne recommend]

5.4 Limitations

[Describe known limitations]

5.5 Post conditions / Expected outcome

[Describe the expected outcome]

5.6 Tips for the integrator

[Provide good-to-know information here]

6. Revision history

Version	Date	Author	Description
0.1 (beta edition)	2025-05-13	Erik Arvidsson	First draft. Not completed. Not reviewed.