



SoftOne GO: API Endpoint

TimeCodeTransactionSimple (Post)



Date Version: 2025-05-13

Contents

1. TimeCodeTransactionSimple (Post)	3
1.1 Endpoint description	3
2. API Access Information	3
2.1 Read more	3
3. Request parameters	3
3.1  Authentication	3
3.2  Main Structure	3
3.3 Request sample	5
3.4 Response parameters	6
3.5 Response sample	7
4. Request examples	8
4.1 Example request X	8
4.2 Example request Y	8
5. Application Reference - SoftOne GO	9
6. Additional information	10
6.1 Pre-conditions / How to get started	10
6.2 Use cases	10
6.3 Best practices	10
6.4 Limitations	10
6.5 Post conditions / Expected outcome	10
6.6 Tips for the integrator	10
7. Revision history	11

1. TimeCodeTransactionSimple (Post)

API Endpoint: TimeCodeTransactionSimple (Post) | Retrieve time transactions for employees

1.1 Endpoint description

The **TimeCodeTransactionSimple** endpoint is used to register or import time-based transactions for employees in SoftOne GO. Each transaction may represent a shift, deviation, or specific payroll-related time entry. The request supports submitting detailed information such as the time span, minutes worked, a registration code, and optional comments. This endpoint is commonly used in integrations with external time-tracking systems, terminals, or custom-built scheduling tools.

2. API Access Information

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

2.1 Read more

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

3. Request parameters

3.1 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

3.2 Main Structure

The table below lists the available parameters for submitting a time code transaction to the endpoint.

Name	Data type	Mandatory	Description
TimeCodeCode	string	Yes	The code and key of the TimeCode
TimeCodeName	string	No	Name of connected TimeCode (readonly)
ProjectNr	string	No	Connected Project if any
EmployeeNr	string	Yes	Connected EmployeeNr (key)

Type	string	Yes	TimeCode type (readonly enum value)
Amount	double	No	Amount
AmountCurrency	double	No	Amount in the currency
AmountEntCurrency	double	No	Recalculated to company main currency
AmountLedgerCurrency	double	No	Amount for ledger currency
Vat	double	No	VAT amount
VatCurrency	double	No	VAT amount in the currency
VatEntCurrency	double	No	VAT recalculated to company main currency
VatLedgerCurrency	double	No	VAT amount for ledger currency
Quantity	double	Yes	Quantity (time in minutes)
InvoiceQuantity	double	No	Quantity for billing
Start	date-time	Yes	Time when the transaction starts
Stop	date-time	Yes	Time when the transaction ends
Comment	string	No	Internal comment
ExternalComment	string	No	Customer-visible comment
AccountNr	string	No	Main account number
AccountNrSieDim1	string	No	Cost center
AccountNrSieDim2	string	No	Cost unit
AccountNrSieDim6	string	No	Project
AccountNrSieDim7	string	No	Employee
AccountNrSieDim8	string	No	Customer
AccountNrSieDim9	string	No	Supplier
AccountNrSieDim10	string	No	Invoice
AccountNrSieDim30	string	No	Region
AccountNrSieDim40	string	No	Shop
AccountNrSieDim50	string	No	Department

3.3 Request sample

The following JSON example shows a sample request with fictional values for a work transaction:

```
[
  {
    "TimeCodeCode": "WORK01",
    "TimeCodeName": "Work time",
    "ProjectNr": "PRJ001",
    "EmployeeNr": "EMP001",
    "Type": "Work",
    "Amount": 1500,
    "AmountCurrency": 1500,
    "AmountEntCurrency": 1400,
    "AmountLedgerCurrency": 1450,
    "Vat": 375,
    "VatCurrency": 375,
    "VatEntCurrency": 350,
    "VatLedgerCurrency": 360,
    "Quantity": 480,
    "InvoiceQuantity": 480,
    "Start": "2025-06-20T08:00:00Z",
    "Stop": "2025-06-20T16:00:00Z",
    "Comment": "Worked full shift",
    "ExternalComment": "Development project",
    "AccountNr": "7010",
    "AccountNrSieDim1": "COST01",
    "AccountNrSieDim2": "UNIT01",
    "AccountNrSieDim6": "PRJ001",
    "AccountNrSieDim7": "EMP001",
    "AccountNrSieDim8": "CUST01",
    "AccountNrSieDim9": "SUP01",
    "AccountNrSieDim10": "INV01",
    "AccountNrSieDim30": "REG01",
    "AccountNrSieDim40": "SHOP01",
    "AccountNrSieDim50": "DEPT01"
  }
]
```

3.4 Response parameters

The table below describes the fields returned in the response, which reflect the stored values from the submitted transaction.

Name	Data type	Description
TimeCodeCode	string	The code and key of the TimeCode
TimeCodeName	string	Name of connected TimeCode (readonly)
ProjectNr	string	Connected Project if any
EmployeeNr	string	Connected EmployeeNr (key)
Type	string	TimeCode type (readonly enum value)
Amount	double	Amount
AmountCurrency	double	Amount in the currency
AmountEntCurrency	double	Recalculated to company main currency
AmountLedgerCurrency	double	Amount for ledger currency
Vat	double	VAT amount
VatCurrency	double	VAT amount in the currency
VatEntCurrency	double	VAT recalculated to company main currency
VatLedgerCurrency	double	VAT amount for ledger currency
Quantity	double	Quantity (time in minutes)
InvoiceQuantity	double	Quantity for billing
Start	date-time	Time when the transaction starts
Stop	date-time	Time when the transaction ends
Comment	string	Internal comment
ExternalComment	string	Customer-visible comment
AccountNr	string	Main account number
AccountNrSieDim1	string	Cost centre
AccountNrSieDim2	string	Cost unit
AccountNrSieDim6	string	Project
AccountNrSieDim7	string	Employee
AccountNrSieDim8	string	Customer
AccountNrSieDim9	string	Supplier
AccountNrSieDim10	string	Invoice
AccountNrSieDim30	string	Region
AccountNrSieDim40	string	Shop
AccountNrSieDim50	string	Department

3.5 Response sample

Below is a sample JSON response representing a successfully retrieved or stored transaction:

```
[
  {
    "TimeCodeCode": "WORK01",
    "TimeCodeName": "Work time",
    "ProjectNr": "PRJ001",
    "EmployeeNr": "EMP001",
    "Type": "Work",
    "Amount": 1500,
    "AmountCurrency": 1500,
    "AmountEntCurrency": 1400,
    "AmountLedgerCurrency": 1450,
    "Vat": 375,
    "VatCurrency": 375,
    "VatEntCurrency": 350,
    "VatLedgerCurrency": 360,
    "Quantity": 480,
    "InvoiceQuantity": 480,
    "Start": "2025-06-20T08:00:00Z",
    "Stop": "2025-06-20T16:00:00Z",
    "Comment": "Worked full shift",
    "ExternalComment": "Development project",
    "AccountNr": "7010",
    "AccountNrSieDim1": "COST01",
    "AccountNrSieDim2": "UNIT01",
    "AccountNrSieDim6": "PRJ001",
    "AccountNrSieDim7": "EMP001",
    "AccountNrSieDim8": "CUST01",
    "AccountNrSieDim9": "SUP01",
    "AccountNrSieDim10": "INV01",
    "AccountNrSieDim30": "REG01",
    "AccountNrSieDim40": "SHOP01",
    "AccountNrSieDim50": "DEPT01"
  }
]
```

4. Request examples

4.1 Example request X

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

POST

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

BODY

[...]

4.2 Example request Y

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

POST

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

BODY

[...]

5. 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>

6. 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]

6.1 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.]

6.2 Use cases

[Describe the common practices when used]

6.3 Best practices

[Describe the common practices SoftOne recommend]

6.4 Limitations

[Describe known limitations]

6.5 Post conditions / Expected outcome

[Describe the expected outcome]

6.6 Tips for the integrator

[Provide good-to-know information here]

7. Revision history

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