Application Services - Web Services (API)

- Introduction
- Creating API Keys
- Application Services
- Department Services
- Attendance Services
- Employee Services
- Absence Services
- Importing Opening Balances and Corrections for Absence
- Importing Custom Fields
- KPI Services

Introduction

tamigo offers a set of REST-based services for you to create new ways to interact with tamigo.

The services are divided into two categories:

- The first set of services can be called using an application access and is meant to be used for integrating third-party applications with tamigo.
- The second set of services uses a tamigo user as authentication and will return data based on the user's context. These services can be used to create other user experiences for tamigo, and we use it to power the mobile applications for tamigo.

The services are all capable of returning XML- or JSON-formatted data. The response you get will be based on the type of input and the specified content type in the HTTP headers.

The services URL is https://api.tamigo.com.

Creating API Keys

The tamigo API enables you to collect data directly from tamigo or transmit data directly into tamigo via web services. You can do this at any time and from any other system, for example, from your cash register system.

Using APIs thus provides optimum flexibility for data integration with other systems.

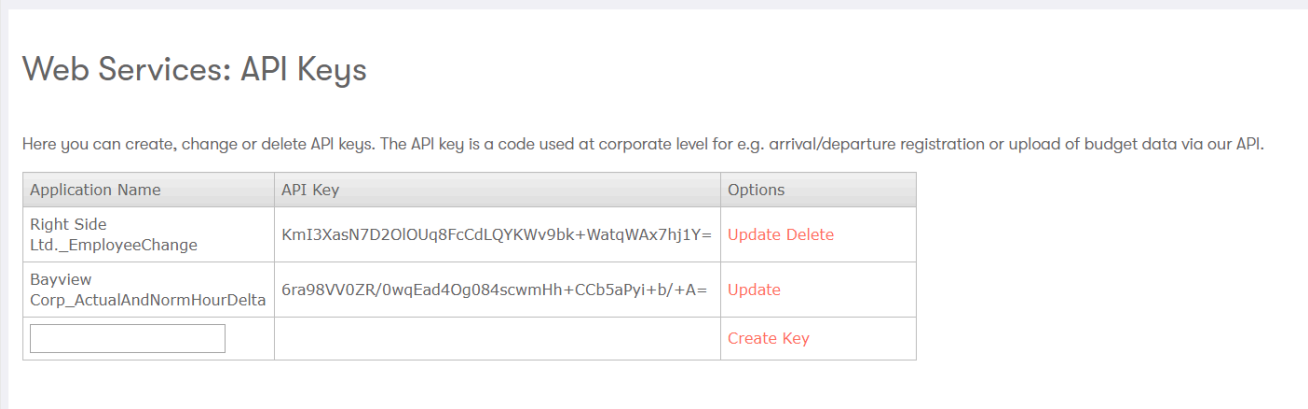
For security reasons, all communication with the tamigo API requires a unique access key which allows you to communicate with tamigo web services. For example, if you want to allow your POS provider to send information directly to tamigo, they will need an access key. You can create an access key for them yourself.

Creating an Access Key

To create an access key, do the following:

1. Go to *Configuration -> Web Services (API)*
2. Enter the name of the application or the provider to state clearly who will be using the key
3. Select *Create Key*

tamigo then generates a key to be used for the tamigo web services.



You can then edit or delete any existing API keys.

Application Services

a. Login Services

b. Application Login

To log in as an application, you:

- Log in to tamigo as an Administrator
- Go to *Configuration -> Web Services (API)*
- Create a new application key by entering an application name and clicking *Create Key*

The application key is used for authentication of your application and you must therefore secure it properly.

If your key has been compromised, it is possible to regenerate the key to ensure the security of your tamigo instance.

The response returned from the service contains the SessionToken. Use this token in subsequent calls to the service to authenticate yourself.

Request (JSON)

```
POST /login/application HTTP/1.1
Content-Type: application/json

{"Name":"micros", "Key":"longstringasyourpasscodegeneratedfromtheapplication"}
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"DefaultDepartmentId":"",
" DefaultDepartmentName ":"",
"DefaultCompanyId":"",
"DefaultCompanyName":"",
"Email":"",
"Password":"",
"Role":"Application",
"MenuId":"5",
"SessionToken":"e2c152f3-49ea-4b50-948b-bc0520261737"}
```

c. Partner Login

To log in as a partner, you must get a login from tamigo by contacting support@tamigo.com.

The partner key is used for your authentication and you must therefore secure it properly. If your key has been compromised, you must contact us immediately, so we can disable it and give you a new key.

The response returned from the service contains the SessionToken. Use this token in subsequent calls to the service to authenticate yourself.

Request (JSON)

```
POST /login/application HTTP/1.1
Content-Type: application/json

{"Name":"micros", "Key":"longstringasyourpasscodegeneratedfromtheapplication"}
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"DefaultDepartmentId":"",
" DefaultDepartmentName ":"",
"DefaultCompanyId":"",
"DefaultCompanyName":"",
"Email":"",
"Password":"",
"Role":"Application",
"MenuId":"5",
"SessionToken":"e2c152f3-49ea-4b50-948b-bc0520261737"}
```

Department Services

a. Get All Departments

You can get a list of all departments of the company.

Request (JSON)

```
GET /departments/application/?includeParents=includeParentDepartments}/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"DepartmentId": "00000000-0000-0000-0000-23e1b49fcdb4",
"IsDefault": false,
"DepartmentKey": "30601",
"Name": "Department A",
"ParentDepartmentId": "00000000-0000-0000-0000-000000000001"
},
```

Department import:

1. Method: POST
2. URL: /Import/
3. Parameters:
   Name (string)
   StoreId (string)

ParentStoreId (string)
ResponsibleEmployeeId (string)

4. Request body example:

```
[{
"Name": "department name",
 "StoreId": "100",
 "ParentStoreId": "200",
 "ResponsibleEmployeeId": "1500"
},
{
 "Name": "parent department name",
 "StoreId": "200",
 "ParentStoreId": "",
 "ResponsibleEmployeeId": "1600"
}]
```

5. Response example:

```
[{
"StoreId": "100",
 "Success": true,
 "Message": "Department imported",
},
{
 "StoreId": "200",
 "Success": true,
 "Message": "Department imported",
}]
```

6. Validation upon request:
   - StoreID must not be empty
   - Parent department must exist (in tamigo or among departments being sent in the request)
   - Cannot have the same name as an existing department
   - If the department is the parent department, it cannot contain employees
   - Responsible employee must exist in tamigo
   - Request cannot contain circular dependent parent - whole request will be rejected
7. Additional notes:
   - Invalid departments in a request will not be imported. Others will be imported (except in the case of circular dependency, where the whole request is rejected).

Department update:

1. Method: POST
2. URL: /Update/

3. Parameters:
   Name (string)
   StoreId (string)
   ParentStoreId (string)
   ResponsibleEmployeeId (string)
4. Request body example:

```
[{
 "Name": "new department name",
  "StoreId": "100",
  "ParentStoreId": "200",
  "ResponsibleEmployeeId": "2500"
 },
 {
  "Name": "new parent department name",
  "StoreId": "200",
  "ParentStoreId": "",
  "ResponsibleEmployeeId": "2600"
 }]
```

5. Response example:

```
[{
 "StoreId": "100",
  "Success": true,
  "Message": "Department updated",
 },
 {
  "StoreId": "200",
  "Success": true,
  "Message": "Department updated",
 }]
```

6. Validation upon request:
   - StoreID must not be empty
   - Department must exist in tamigo
   - Parent department must exist in tamigo
   - If the department is a parent department, it cannot contain employees
   - Responsible employee must exist in tamigo
7. Additional notes:
   - Invalid departments in a request will not be updated. Others will be updated.
   - If the department hierarchy is updated and since the logic is one department at a time, it is necessary to send a list of department updates ordered in a way that goes from top to bottom of the new hierarchy (from root to leafs).

Department delete:

1. Method: POST
2. URL: /Delete/
3. Parameters:
   StoreId (string)
4. Request body example:

```
[{
 "StoreId": "100"
 },
 {
  "StoreId": "200"
 }]
```

5. Response example:

```
[{
 "StoreId": "100",
  "Success": true,
  "Message": "Department deleted",
 },
 {
  "StoreId": "200",
  "Success": true,
  "Message": "Department deleted",
 }]
```

6. Validation upon request:
   - StoreID must not be empty
   - Department must exist in tamigo
   - Department must not have child departments
   - If department is last department existing in company
7. Additional notes:
   - If multiple departments in same the hierarchy branch are sent and since the validation will not allow deleting departments with child departments, it is necessary to order a list of departments in a way that goes from the bottom to the top of the hierarchy (from leafs to root).

Attendance Services

tamigo offers the attendance services, which allow you to register check-in and check-out times for employees directly in tamigo from a third-party system (e.g. Point-of-Sale (POS) system).

The attendance services require you to set up keys in tamigo for the departments and employees who will use the services.

To set up these keys, you:

- Log in to tamigo as a Planner or Administrator
- Go to Configuration -> Departments, to set the Department ID
- Go to Configuration -> POS Keys, to set the corresponding keys for the employees

a) Opening Hours

tamigo offers a way to get a list of opening hours.

Request (JSON)

```
GET /OpeningHours/2018-01-01/2018-01-02/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
     {
        "DepartmentName": "Shop A",
        "NormalOpeningHours": [],
       "SpecialOpeningHours": [],
        "StoreId": null
    },
    {
        "DepartmentName": "Shop A",
        "NormalOpeningHours": [],
        "SpecialOpeningHours": [],
       "StoreId": null
]
```

b) Check-In

Use the check-in service to register an employee who has arrived at work. tamigo will register the check-in at the time you send and match it to the employee and department. If time rounding rules have been applied, the service returns the rounded time.

When using optional parameter `ActivityKey`, the employee will be checked in to the current shift, and the check-out time will not be registered in worked hours.

For a complete example of usage, see the *Batch Update* section.

If `EmployeeId` and `EmployeeKey` are both provided, then `EmployeeId` is used.

Parameter	Format	Example
departmentKey	text	A123, 1234567, abcdefg
employeeKey	text	A123, 1234567, abcdefg
checkInTime	yyyy-MM-ddThh:mm:ss	2011-09-02T12:37:00
sessionToken	as returned by login	766c9732-e2d1-46d1-ae3e-a74c560bb8e6
activityKey (optional)	text	Lunch, SmallBreak, (activity key)
EmployeeId	GUID	766c9732-e2d1-46d1-ae3e-a74c560bb8e6

Request (JSON)

```
POST /attendance/checkin/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

```
{
"DepartmentKey":"<departmentKey>",
"EmployeeKey":"<employeeKey>",
"CheckInTime":"<checkInTime>",
"ActivityKey":"<activityKey>",
"EmployeeId":"<employeeId>",
}
```

Response (JSON)

```
HTTP/1.1 200 OK
<actual checkin date time>
```

c) Get Check-in/Check-out Overview

You can get an overview of check ins and check outs in a selected period.

- Start and End dates are in format yyyy-MM-dd
- departmentId can relate to a specific department Id (Guid) or «all» can be sent to fetch information for all departments

Request (JSON)

```
GET /attendance/CheckInOutOverview/?startDate=2020-01-01&endDate=2020-01-
31&departmentId=all HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
    {
        "ActualShiftComment": "",
        "ActualShiftHours": 0.02,
        "ActualShiftText": "13:45-13:46",
        "CheckInOutShiftComment": null,
        "CheckInOutShiftHours": 0.06,
        "CheckInOutShiftText": "13:41-13:45",
        "Date": "/Date(1600898400000+0200)/",
        "DepartmentId": "00000000-0000-0000-0000-000000000000",
        "DepartmentName": "Department 1",
        "DepartmentShiftedTimeZone": "",
        "EmployeeId": "00000000-0000-0000-0000-000000000000",
        "EmployeeName": "Employee Name",
        "PlannedShiftComment": "",
        "PlannedShiftHours": 0.50,
        "PlannedShiftText": "14:00-14:30"
        "WageSystemKey": "12345"
    }
]
```

d) Batch Update

The batch update service is similar to the check-in and check-out services, only that it gets a multiple array of check-in and -out times, in chronological order. This import can also be

used for multiple employees at the same time. The method will return the same array as was sent, only that the `CheckInTime` and `CheckOutTime` will be adjusted to actual time registered.

For example, the below check-in and -out times will be registered in the system:

Shifts visible only on check-out:

- 09:00-10:15
- 10:15-10:33 `SmallBreak`
- 10:33-12:03
- 12:03-12:44 `Lunch`
- 12:44-17:02

Shifts visible in the Timesheet:

- 09:00-10:15
- 10:33-12:03
- 12:44-17:02

Request (JSON)

```
POST /attendance/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

```
[
{"DepartmentKey":"<departmentKey>","EmployeeKey":"
<employeeKey>","CheckInTime":"2017-04-07T09:00:11"},
{"DepartmentKey":"<departmentKey>","EmployeeKey":"
<employeeKey>","CheckInTime":"2017-04-07T10:15:11",
"ActivityKey":"SmallBreak"},
{"DepartmentKey":"<departmentKey>","EmployeeKey":"
<employeeKey>","CheckOutTime":"2017-04-07T10:33:11",
"ActivityKey":"SmallBreak"},
{"DepartmentKey":"<departmentKey>","EmployeeKey":"
<employeeKey>","CheckInTime":"2017-04-07T12:03:11",
"ActivityKey":"Lunch"},
{"DepartmentKey":"<departmentKey>","EmployeeKey":"
<employeeKey>","CheckOutTime":"2017-04-07T12:44:11",
"ActivityKey":"Lunch"},
{"DepartmentKey":"<departmentKey>","EmployeeKey":"
<employeeKey>","CheckOutTime":"2017-04-07T17:02:11"}
]
```

Response (JSON)

```
HTTP/1.1 200 OK
```

e) Export Absence

Use this service to export absence from one or more departments.

Use the following parameters:

`StartDate` : Date formatted as MM-DD-YYYY.

`EndDate` : Date formatted as MM-DD-YYYY.

`DepartmentId` : Optional. If you do not use it, the result returned is for all departments.

Request (JSON)

```
GET /Leave/ByDate/startDate=<startDate>&endDate=<endDate>
&departmentId=<departmentId>/ HTTP/1.1
Content-Type application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

[{"AbsenceType":"Vacation",
"Date":"\/Date(1324854000000+0100)\/",
"Name":"John Doe",
"WageSystemKey":"00044"},
...
{"AbsenceType":"Vacation",
"Date":"\/Date(1324940400000+0100)\/",
"Name":" John Doe ",
"WageSystemKey":"00044"}]
```

f) Import Revenue

Use this service to import the daily revenue.

Use the following parameters:

`ApplicationName` : Same name as used when logging in

`Key` : Same key as used when logging in (I)

`Content` : A CSV file formatted as `DepartmentId`, date (YYYY-MM-DD), revenue.

`Format` : Use `Standard`

`DateTimeReceived` : The time you send the request at

`Type: NULL`

Request (JSON)

```
POST /Revenues/ HTTP/1.1
Content-Type application/json
x-tamigo-token: <token>
```

```
 {
"ApplicationName": "<ApplicationName>",
"Key": "<ApplicationKey>”,
"Content": [
"100,2011-01-01,1000",
"100,2011-01-02,1200"
],
"Format": "Standard",
"DateTimeReceived": "\/Date(1327996809388+0100)\/",
"Type": null
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

g) Import Revenue Over Day/Footfall

Use this service to import the revenue over the day compared to footfall.

Use the following parameters:

`ApplicationName` : Same name as used when logging in

`Key` : Same key as used when logging in

`Content` : A CSV file formatted as `amount;amountType;startDateTime;endDateTime;posId;statusType;employeePosKey`

`Format` : Use `Standard`

`DateTimeReceived` : the time you send the request at

`Type: NULL`

`Amount`: The number of either revenue or footfall

`Amounttype`: 1 = Revenue, 2 = Footfall (customers)

`StartDateTime`: Start of data to import (should be whole quarters, ie start and end 01-01-2014 13:00 to 01-01-2014 13:15)

`EndDateTime`: End of data to import

`posId`: Department ID

`statusType`: 1 = Actual, 2 = forecast

`employeePosKey`: Should only be filled for employee-specific revenue

Request (JSON)

```
POST /Revenues/UploadRevenueOverDay/ HTTP/1.1
Content-Type application/json
x-tamigo-token: <token>
```

```
"ApplicationName": "<ApplicationName>",
"Key": "<ApplicationKey>”,
"Content": [
 "12;2;2014-01-01 13:00;201-01-01 13:15;999;1;NULL",
 "6;2;2014-01-01 13:15;201-01-01 13:30;999;1;NULL",
 ],
"Format": "Standard",
"DateTimeReceived": "\/Date(1327996809388+0100)\/",
"Type": null
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

h) Import Transactions

Use this service to import the daily revenue, including related transactions.

Use the following parameters:

`DepartmentKey`: ID of the department

`EmployeePosKey: NULL`

`Time` : Time of the transaction

`Amount` : Total amount of lines in the array

`Lines` : Array using the following parameters:

`Count` : Number of items

`Price` : Item price

`ProductKey` : Product key of product

`ProductName` : Name of product

Request (JSON)

```
POST /Revenues/UploadTransactions/ HTTP/1.1
Content-Type application/json
x-tamigo-token: <token>

[
 {
   "DepartmentKey": "100",
   "EmployeePosKey": null,
   "Time": "\/Date(1412589803535+0200)\/",
   "Amount": 1000.5,
   "Lines": [
      {
        "ProductKey": "Some-product-key",
        "ProductName": "Some-product-name",
        "Count": 2,
        "Price": 500.25
      }
    ]
  }
]
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Employee Services

The employee services allow you to get and update employee information for use in a third-party system (ie payroll system). The employee services also allow you to create new Employees, Planners, or Administrators from third-party applications.

a) Get Employees

Use this service to get a list of all employees in the company.

Request (JSON)

```
GET /employees/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

[{"Email":"user@company.com","EmployeeId":"9c0f571e-cd0e-4e38-ac51-
ecf965cd3de6","IsPlanner":false,"Name":"Jane","PosKey":null,
"DepartmentRoles":null },
...
{"Email":"user2@company.com","EmployeeId":"d1d0b7db-c774-4e81-b094-
d1d2acaa1d83","IsPlanner":false,"Name":"John","PosKey":null, ","DepartmentRoles
":null }]
```

b) Get Detailed Employees

Five hundred employees of a company will be retrieved, including related data, such as notes, salary distribution, and skills.

Employees are displayed in alphabetical order and can be paged. You will get the first five hundred employees if the page is excluded from the query or is set to 0.

If the set home department of an employee starts before the start date of the employee, the API will set the home department to begin the same date as the employee. In cases where an employee has a start date in the future, the API will still return a valid HomeDepartment.

Full request (JSON)

```
GET /employees/getemployeedetails/?page=<pagenumber>
&includedeleted=true,false>/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Minimal request (JSON)

```
GET /employees/getemployeedetails/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
    "About": "",
    "Address1": "",
    "Address2": "",
    "BankAccount": "",
    "BankRegistration": "",
    "Birthdate": "1984-01-01",
    "CanAddOwnShiftsOnSmartphone": false,
    "CanApproveEmployeeChanges": false,
    "CanEditActualHours": false,
    "CanEditPlan": false,
    "CanSendSms": false,
    "CarRegistration": "",
    "City": "",
    "Competencies": [],
    "ContractHours": [
          {
             "Comment": "",
             "ContractHours": 25.0000,
             "ContractHoursHistoryId": "aaafac41-a4e0-4217-bf31-294ae0f9175b",
             "ContractHoursHistoryPerDays": [
             {
                  "ContractHoursHistoryPerDayId": "8eaefe30-b3bf-42fb-a3eb-24650da0
                  "DayIndex": 1,
                  "Hours": 5.0000
              },
              {
                  "ContractHoursHistoryPerDayId": "9a84ca65-c962-4a7b-9854-e56b68dd
                  "DayIndex": 2,
                  "Hours": 5.0000
               },
               {
                    "ContractHoursHistoryPerDayId": "b63403cb-00cd-45b9-b2e2-4d38f7
                    "DayIndex": 3,
                    "Hours": 5.0000
                },
                {
                     "ContractHoursHistoryPerDayId": "8d71378e-4b99-4aad-9df9-434393
                     "DayIndex": 4,
                     "Hours": 5.0000
```

```
            },
            {
                "ContractHoursHistoryPerDayId": "9413ce96-2103-40a4-b846-d5dc3
                "DayIndex": 5,
                "Hours": 5.0000
            }
          ],
          "ContractHoursMonthly": null,
          "EndDate": "2022-04-17T00:00:00",
          "EntryInformation": "Changed by Tamigo Support (20/04/2022 10:05:17)"
          "StartDate": "2022-04-14T00:00:00",
          "Temporary": false,
           "WorkDaysPerWeek": 5,
           "ApprovedBy": "00000000-1112-0000-0000-000000000000",
           "ApprovedInfo": "Added by:",
           "ApprovedOn": "2022-04-20T10:04:08",
            "CanBeRemovedByCurrentUser": true,
            "CreatedBy": "00000000-1112-0000-0000-000000000000",
            "CreatedOn": "2022-04-20T10:04:08",
           "IsApproved": true,
          "ModifiedBy": "00000000-1112-0000-0000-000000000000",
          "ModifiedOn": "2022-04-20T10:05:17"
     }
"CurrentContractType": "Monthly",
"CurrentPaymentModel": "PaidByMonthComplex",
"CurrentWageRateType": "",
"CustomColumns": [
    {
      "Description": "",
      "Name": "",
      "Type": "CheckBox",
      "Value": "False"
   }
 ],
"CustomRoles": []
"DeletedOn": null,
"Email": "",
"EmployeeId": "00000000-0000-0000-0000-000000000000",
"EmployeeIsAdmin": false,
"EmployeeType": "",
"EmployeeTypeHistories": [
     {
      "CreatedBy": "00000000-0000-0000-0000-000000000000",
      "CreatedOn": "/Date(0000000000000+0100)/",
      "EndDate": null,
      "ModifiedBy": null,
      "ModifiedOn": null,
      "Name": "",
      "StartDate": "/Date(0000000000000+0100)/"
```

```
      }
    ],
  "EmployerNumber": "",
  "EnforceMaxWeeklyWorkingHours": false,
  "From": "2000-01-01",
  "Gender": null,
  "HeadCount": null,
  "HistoricalWages": [
     {
       "StartDate": "2020-12-01",
       "Wage": 11231.11
      }
   ],
  "HomeDepartment": "00000000-0000-0000-0000-000000000000",
  "HomeDepartments": [
      {
       "CreatedBy": "00000000-0000-0000-0000-000000000000",
       "CreatedOn": "/Date(000000000000+0100)/",
       "DepartmentId": "00000000-0000-0000-0000-000000000000",
       "DepartmentName": "",
       "EndDate": null,
       "ModifiedBy": null,
       "ModifiedOn": null,
       "StartDate": "/Date(000000000000+0100)/"
      }
    ],
  "IdNumber": "",
  "IsActive": true,
  "IsUserEnabled": true,
  "ManualRegistrations": [],
  "MaximumHours": 40,
  "MaximumWageRate": 10,
  "MediaTypes": [
      {
       "Name": "Sms",
       "Selected": false
      },
      {
       "Name": "Email",
       "Selected": true
      }
  ],
  "MinimumHours": 30,
  "MinimumWageRate": null,
  "Name": "John",
  "NotAvailable": [],
  "Notes": [],
  "Phone": "123456",
  "Phone2":"123456",
```

```
    "PhoneCountryCode": "+32",
    "ReasonForTermination": "",
    "Roles": [{
        "Admin": false,
        "DepartmentId": "00000000-0000-0000-0000-000000000000",
        "Employee": true,
        "Planner": false
      }],
    "SendShiftReminder": false,
    "StandardHours": null,
    "To": "2010-01-01",
    "WageNumber": "1234",
    "WageRateTypes": [
         {
        "ContractType": "",
        "CreatedOn": "2000-01-30T00:00:00",
        "EndDate": null,
        "EntryInformation": "Added by XXXXX xxxxx  (2000-01-30T00:00:00)",
        "ModifiedOn": null,
        "PaymentModel": "PaidByMonthComplex",
        "StartDate": "2000-01-03T00:00:00",
        "WageRateTypeName": "01 XXX"
         }
    ],
    "WorkTravelTime": 30,
    "Zip": ""
  }
]
```

You can also get employee details for only one specific user by using the following request:

```
GET /employees/getemployeedetails/?page=<pagenumber>&includedeleted<true,false>
&employeeID=<GUID>/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

c) Get Login Overview

You can get an overview of employee logins.

- Date is in format yyyy-MM-dd
- departmentId can relate to a specific department Id (Guid) or «all» can be sent to fetch information for all departments

Request (JSON)

```
GET /employees/LoginOverview/?fromDate=2020-01-01&departmentId=all HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
    {
        "EmailId": "email@tamigo.com",
        "EmployeeId": "00000000-0000-0000-0000-000000000000",
        "FailureLoginCount": 0,
        "FirstName": "Employee Name",
        "ICalLastAccessedOn": null,
        "LastLogin": "/Date(1601458052860+0200)/",
        "SmartphoneLoginCount": 0,
        "TouchLoginCount": 0,
        "WageSystemKey": "12345",
        "WebLoginCount": 0
    }
]
```

d) Create Employees

You can create a new Employee, Planner, or Administrator. If the password is left blank, a random password will be generated.

Request (JSON)

```
POST /Employees/?departmentId={departmentId} HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

```
{"Email": "user@company.com",
 "Name": "Jane Doe",
 "Phone": "88888888",
"EmployeNumber": "1234",
"WageSystemKey": "4321",
"Role": "name of the role ex. Employee, Planner, Administrator",
"IsEnabled": "true"
"Password": "password"
}
```

Response (JSON)

```
HTTP/1.1 200 OK
{ Success":true}
```

e) Create or Update Employees

You can create and update employees using the following endpoint.

Requirements:

- DepartmentKey

  DepartmentKey represents the store ID and must be valid for the employee to be imported or updated.

- Employee identifier

  There are 4 possible identifiers you can use when creating/updating employees:

1. Email address
2. WageSystemKey (displayed as Employee ID in tamigo)
3. POS key (visible under *Configuration -> POS Keys* in tamigo)
4. EmployeeId – can be used when updating employees (guid specific to every employee)

   If the employee's identifier is already available in the database, the employee record will be updated automatically.

Employee information notes:

- All dates must be in Microsoft JSON date format, which consists of epoch timestamp in milliseconds, wrapped in "/Date()/".
- When assigning a contract type, you can have one of 3 possible fields: "WageModel", "WageModelId", or "WageModelDescription". If the request contains none of them, then a random contract type, or default contract type, which is setup in the company settings will be assigned.
- The Role key accepts both standard tamigo role names, as well as custom roles. If no matching name is found, the employee will be created with the Employee role.
- To make sure wage is set up correctly, you must send the MonthlySalary or HourlyRate keys, depending on whether the employee's contract type is monthly or hourly paid.
- Gender: 0 for male; 1 for female.
- SkillsToAdd has the alternative name CompetenciesToAdd, and SkillsToRemove has the alternative name CompetenciesToRemove. The values should be the Skill ID defined in the Skills page in tamigo.

Import settings notes:

- ShouldActivateEmployee can be set to true, if you wish to activate all employees upon create/update.
- ShouldSendActivationEmail can be set to true, if you wish to activate employees by email upon create/update.
- ForceUpdateEmployeeHistories will allow overwrite of employee histories like home department, contract type, or position.
- ShouldUpdateEmptyValuesOnFields will allows deletion of data. If this setting is set to false, an empty value will be ignored on import. If it is set to true, then empty values will overwrite existing values in tamigo. To remove dates, send in null value with this setting enabled.
- ContractTypeStartDateEqualsEmployeeStartDate; will make Contract Type start date equal to employee start date. Only relevant when adding a new employee.
- HomeDepartmentStartDateEqualsEmployeeStartDate; will make Home Department start date equal to employee start date. Only relevant when adding a new employee.
- PositionStartDateEqualsEmployeeStartDate; will make Position start date equal to employee start date. Only relevant when adding a new employee.

Request (JSON)

```
POST /Employees/CreateUpdateEmployees/ HTTP/1.1
Accept: application/json
Content-Type: application/json
x-tamigo-token: <token>

{
    "Employees": [
        {
            "EmployeeId": null,
            "FirstName": "Test user 1",
             "ContractHour": {
                "ContractHoursEntry": {
                     "Comment": null,
                     "StartDate": "2022-05-20T00:00:00",
                     "Temporary": true,
                     "ContractHoursHistoryPerDays": [
                          {
                              "DayIndex": 1,
                              "Hours": 5
                          },
                          {
                               "DayIndex": 2,
                               "Hours": 4
                          }
                      ]
                 }
             }
             "Email": "create.user@tamigo.com",
```

```
"SocialSecurityNumber": "9999999",
"WageSystemKey": "12345",
"DateOfBirth": "/Date(631152000000)/",
"StartDate": "/Date(1579910400000)/",
"SkillsToAdd": ["SK2"],
"SkillsToRemove": ["SK1"],
"EndDate": "/Date(1609372800000)/",
"TerminationReason":"FIN",
"Gender": 0,
"ContractHours": 37.5,
"ContractHoursAreTemporary": true,
"ContractHoursComment": "Contract H. Comment",
"ContractHourStartDate": "/Date(1579910400000)/",
"ContractHourEndDate": "/Date(1609372800000)/",
"MinimumHours": 30.5,
"WageModel": "Contract type 1",
"WageModelId": "CT1",
"WageModelDescription": "Contract type 1 description",
"ContractTypeStartDate": "/Date(1579910400000)/",
"Role": "Planner",
"AddressLine1": "Address 1",
"AddressLine2": "Street 2",
"PostCode": "1000",
"City": "Copenhagen",
"CountryCode": "+45",
"MobilePhoneNumber": "1111111",
"HomePhoneNumber": "2222222",
"DepartmentKey": "DEP1",
"DepartmentStartDate": "/Date(1579910400000)/",
"DepartmentEndDate": "/Date(1609372800000)/",
"BankRegistrationNumber": "333333",
"BankAccountNumber": "0000 1111 2222 3333",
"HourlyRate": 14.5,
"MonthlySalary": 2000.0,
"WageRate": 2000.0,
"WageRateStartDate": "/Date(1579910400000)/",
"PosKey": "1",
"Position": "Supervisor",
"PositionStartDate": "/Date(1579910400000)/",
"PositionEndDate": "/Date(1609372800000)/",
"WorkDaysPerWeek": 5,
"TravelTimeToWork": 15,
"CustomColumns":
[
    {
        "Name": "CustomDate",
        "Value": "/Date(1579910400000)/"
    },
    {
```

```
                    "Name": "CustomFieldBool",
                    "Value": false
                },
                {
                    "Name": "CustomFieldText",
                    "Value": "Text custom value"
                }
            ],
            "ActivateEmployee": true,
            "EmployerNumber": "REF NUMBER 1",
            "Password": "tamigo"
        }
    ],
    "ImportSettings":    {
        "ShouldActivateEmployee": true,
        "ForceUpdateEmployeeHistories": true,
        "ContractTypeStartDateEqualsEmployeeStartDate": true,
        "HomeDepartmentStartDateEqualsEmployeeStartDate": true,
        "PositionStartDateEqualsEmployeeStartDate": true,
        "ShouldUpdateEmptyValuesOnFields": true,
        "ShouldSendActivationEmail": true
    }
}}
```

Response (JSON)

```
HTTP/1.1 200 OK

      {
          "Message": "Employees are created/updated successfully",
        "Success": true,
        "Data": [
          {
              "EmployeeId": "9508e424-da92-4f5a-9aab-d2ca95e71123",
            "ImportStatus": 0,
            "ImportText": null,
            "WageSystemKey": "12345"
          }
        ]
      }
```

Note: If the data import was successful, the code 200 will be returned from the API. If there are errors, they will be returned as part of the response.

f) Delete Employees

Requirements:

- The Employee POS key is required to delete a particular employee's record.

Request (JSON)

```
POST /employees/DeleteEmployee/?posKey=<POS-key>/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

g) Delete roles

You can delete all of the roles on a employee, except the employee role in the current home department:

Request (JSON)

```
DELETE /roles/?employeeId={employeeId}&resetRoles={resetRoles}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
{"Message":null,"Success":true}
```

Note that the call is not successful if there is no employee role in the current home department.

You can delete specific roles on an employee, except the last role in the employee's current home department:

Request (JSON)

```
DELETE /roles/?employeeId={employeeId}&roleId={roleId}&customRoleId=
{customRoleId}&departmentId={departmentId}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
{"Message":null,"Success":true}
```

h) Anonymise Employees

Use this service to anonymise an employee. Anonymising will remove all personal information from the employee. This cannot be undone. This can only be done by a user that has the DPO role.
Request (JSON)

```
POST Employees/AnonymizeEmployee/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>

{ "FirstName": "EmployeeName", "EmployeeId":
"00000000-0000-0000-0000-000000000000" }
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"Success": true, "Message": "Possible error"}
```

i) Check if employees need to approve their own shifts

Use this endpoint for mobile apps to retrieve information about whether the employee needs to approve their own shifts or not.
Request (JSON)

```
GET /employees/<employeeId>/needshiftapprovalbyemployee/?date=<date> HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP 1/1 200 OK
Content-Type: application/json; charset=utf-8

{ "Message": "", "Success": true }
```

j) Get employee unavailabilities

This service returns a list of all employee unavailibilities for a selected period. Start and end dates are required, department and employee Ids are optional.

DepartmentId and EmployeeId represent tamigo internal identifiers (GUID), which can be retrieved via other endpoints.

Request (JSON)

```
GET /employees/GetEmployeeUnavailabilities/?from={startDate}&to={endDate}&employeeId
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
Host: api.tamigo.com
Content-Type: application/json
Accept: application/json

[
    {
        "EmployeeDailyUnavailabilities": [
            {
                "DailyUnavailabilities": [],
                "DateTime": "/Date(1640991600000+0100)/"
            },
            {
                "DailyUnavailabilities": [],
                "DateTime": "/Date(1641078000000+0100)/"
            },
            {
                "DailyUnavailabilities": [],
                "DateTime": "/Date(1641164400000+0100)/"
            },
            {
                "DailyUnavailabilities": [],
                "DateTime": "/Date(1641250800000+0100)/"
            },
            {
                "DailyUnavailabilities": [
                    {
                        "EndTime": "/Date(1641351600000+0100)/",
                        "StartTime": "/Date(1641337200000+0100)/"
                    },
                    {
                        "EndTime": "/Date(1641423600000+0100)/",
                        "StartTime": "/Date(1641394800000+0100)/"
                    }
                ],
                "DateTime": "/Date(1641337200000+0100)/"
            },
            {
                "DailyUnavailabilities": [],
                "DateTime": "/Date(1641423600000+0100)/"
```

```
            },
            {
                "DailyUnavailabilies": [],
                "DateTime": "/Date(1641510000000+0100)/"
            }
        ],
        "EmployeeId": "53472a00-b92a-40fb-8a02-09b1582ad695"
    }
]
```

Absence Services

The absence services allow you to create, get, and delete employee's absences from third-party applications.

a) Create Absence

Create one or more employee absences by passing the Employee POS key or employee ID and the absence type.

Requirements:

- Employee POS key or employee ID
- Start date of absence
- End date of absence
- Absence type
- Absence

Request (JSON)

```
POST /LeaveRequests/Create/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>

   [{
     "LeaveRequestType":6,
     "PosKey":<POS Key> | null,
     "EmployeeId": <00000000-0000-0000-0000-000000000000> | null,
     "FromDate":"\/Date(1467370454000)\/",
     "ToDate":"\/Date(1468148054000)\/",
     "Comment":"Comment",
     "RequestedDate":"\/Date(1464778454000)\/",
     "Approved":null | true | false,
     "Rejected":null | true | false,
```

```
    "ApprovedDate":null,
    "NumberOfHours":null,
    "DepartmentId":<00000000-0000-0000-0000-000000000000> | null,
    "EmployeeLeaveRequestId":null
    }]
```

Response (JSON)

```
HTTP/1.1 200 OK
```

b) Create/Update Sick Leave Absence Type

Create an employee sick leave by passing the `EmployeePOSKey`, sick leave date, `leaveRequestTypeID`, and a flag to indicate whether the employee is sick or not sick.

Requirements:

- When creating an employee sick leave, the `IsSick` flag is required to be true, and the `LeaveDate` passed will be considered as the date when the employee's sick leave starts. When the `IsSick flag` is passed as false, the sick leave date provided will be the date when the employee's sick leave ends.

Request (JSON)

```
POST /LeaveRequests/CreateUpdateSickLeave/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>

{
    "EmployeePosKey":"1000",
    "LeaveDate":"\/Date(1467370454000)\/",
    "IsSick":true,
    "LeaveRequestTypeId":6
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

c) Delete an Absence

Requirements:

- Employee POS key
- Sick leave start date

- Sick leave end date
- Absence type

Request (JSON)

```
POST /LeaveRequests/Delete/?posKey=0002&dateFrom=2016-07-01&dateTo=2016-07-10
&leaveType=6/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

d) Delete an Absence with Employee ID

Requirements:

- Employee ID
- Sick leave start date
- Sick leave end date
- Absence type

Request (JSON)

```
DELETE LeaveRequests/DeleteByEmployeeId/?EmployeeId=18f92461-c53b-460b-bef9-5b47d307
&leaveType=6&dateFrom=2021-07-01&dateTo=2021-07-16 HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
{
    "Message": null,
    "Success": true
}
```

e) Delete All Absences

Deletes all employee absences that are related to the Employee POS key and are within a given start and end date range.

Requirements:

- Start date and end date of absences
- EmployeePOSKey is an optional parameter.

Request (JSON)

```
POST /LeaveRequests/Delete/All/?posKey=0002&dateFrom=2016-07-01
&dateTo=2016-07-10/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

f) Get All Absences by Date Range

Get all absences that are within a given start and end date range, including overlapping absences.

Absence shifts are included in the result, as well as the 'Sum' property that shows the sum of hours on an individual shift level. A 'Sum' property on absence level is the sum of hours for the particular absence.

Requirements:

- Start date and end date of absences

Request (JSON)

```
GET /LeaveRequests/All/?startdate=2016-06-09&enddate=2016-06-10/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
[{
    "AbsenceShifts": [
        {
            "ActivityName": "Vacation",
            "Comments": "",
            "CompetencyId": null,
            "CompetencyName": null,
            "IsDraft": null,
            "IsProductive": true,
```

```
            "Shift": "08:00 - 16:00",
            "ShiftActivityId": "00000000-0000-0000-0000-000000000000",
            "Sum": 8.0000
        }
    ],
    "DepartmentId": "00000000-0000-0000-0000-000000000000",
    "DepartmentName": "Department Name",
    "EmployeeId": "00000000-0000-0000-0000-000000000000",
    "EmployeeName": "EmployeeA",
    "LeaveApproved": true,
    "LeaveEndDate": "\/Date(0000000000000+0100)\/",
    "LeaveRejected": false,
    "LeaveStartDate": "\/Date(0000000000000+0100)\/",
    "LeaveType": "Vacation",
    "Sum": 32.0000
}]
```

g) Get Absence Overview

You can get an overview of absences in a selected period.

- Start and End dates are in format yyyy-MM-dd
- departmentId can relate to a specific department Id (Guid) or «all» can be sent to fetch information for all departments
- employeeId (optional) Wage system key of employee to filter by
- absenceId (optional filter) from Leave/LeaveRequestTypes endpoint
- absenceGroupId (optional) from /Leave/Groups endpoint

Request (JSON)

```
GET /Absences/AbsenceOverview/?startDate=2020-01-01&endDate=2020-01-
31&departmentId=all/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
    {
        "AbsenceName": "Vacation",
        "AbsenceTypeId": 1,
        "AcceptDocument": 0,
        "Actual": [
            {
```

```
        "AbsenceName": "Vacation",
                "AbsenceTypeId": 1,
                "AcceptDocument": null,
                "Comment": "",
                "Deleted": false,
                "DepartmentId": "00000000-0000-0000-0000-000000000000",
                "DepartmentName": "Department 1",
                "EmployeeId": "00000000-0000-0000-0000-000000000000",
                "EndDateTime": "/Date(1578677760000+0100)/",
                "GroupId": null,
                "IsActual": true,
                "Name": "Employee Name",
                "RequireComment": false,
                "ShiftHours": 7.6000,
                "StartDateTime": "/Date(1578646800000+0100)/",
                "StoreId": "123",
                "WageSystemKey": "00001234"
            }
        ],
        "Date": "/Date(1578610800000+0100)/",
        "Deleted": false,
        "DepartmentName": "Department 1",
        "EmployeeId": "00000000-0000-0000-0000-000000000000",
        "GroupId": null,
        "Name": "Employee Name",
        "Planned": [
            {
        "AbsenceName": "Vacation",
                "AbsenceTypeId": 1,
                "AcceptDocument": null,
                "Comment": "",
                "CreatedOn": "/Date(1618902212070+0200)/",
                "Deleted": false,
                "DepartmentId": "00000000-0000-0000-0000-000000000000",
                "DepartmentName": "Department 1",
                "EmployeeId": "00000000-0000-0000-0000-000000000000",
                "EndDateTime": "/Date(1578677760000+0100)/",
                "GroupId": null,
                "IsActual": false,
                "ModifiedOn": null,
                "Name": "Employee Name",
                "RequireComment": false,
                "ShiftHours": 7.6000,
                "StartDateTime": "/Date(1578646800000+0100)/",
                "StoreId": "123",
                "WageSystemKey": "00001234"
            }
        ],
        "RequireComment": false,
```

```
        "StoreId": "123",
        "WageSystemKey": "00001234"
    }
]
```

h) Get Absence Groups

This service makes it possible to download absence groups. This endpoint has no addition, it will always return all defined absence groups in tamigo.

Request (JSON)

```
GET /Leave/Groups HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK

[
    {
        "Id": "GUID",
        "Name": "Group name"
    },
…
]
```

i) Get Absence Types

This service makes it possible to download absence groups. This endpoint has no addition, it will always return all defined absence groups in tamigo.

Request (JSON)

```
GET /Leave/LeaveRequestTypes HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK

[
    {
        "CanBeOngoing": true/false,
        "CustomPercentage": true/false,
        "CustomPercentages": true/false,
```

```
        "IsFullDay": true/false,
        "IsTimeSpan": true/false,
        "LeaveTypeId": 1,
        "Name": "Name of absence"
    },
…
]
```

j) Import Absences

This endpoint works in correlation with the general absence plugin, where it can be defined how the importer will work. There the mappings between absences in tamigo and absence IDs which will be sent into tamigo need to be defined. When the response comes back, the absence import function will run in the background. For now, the results from the importer can be requested on demand from tamigo.

Request (JSON)

```
POST Absences/Import HTTP/1.1
x-tamigo-token: <token>

{
    "IntegrationName" : "GeneralAbsence",
    "AbsenceImports" : [
     {"EmployeeId" : "wagesystemKey", "AbsenceId" : "test1", "StartDate" :
      "/Date(1619827200000)/", "EndDate" : "/Date(1619827200000)/", "IsApproved" : t

    ]
}
```

Response (JSON)

```
HTTP/1.1 200 OK

{
    "Message": "Absence import Function started.",
    "Success": true
}
```

k) Retrieve absence balances

This endpoint retrieves absence balances for all employees in a given department in a given year.

Request (JSON)

```
GET /AbsenceBalance/{leaveTypeId}/department/{departmentId}/?year={year} HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
Response(JSON)
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
    {
        "Balance": 321341249.0000,
        "BalanceExpirationDate": null,
        "CurrentTotalLastYearBalance": 321341239.0000,
        "Details": [
            {
                "Balance": 321341250.0000,
                "Comment": null,
                "Correction": 0,
                "Date": "/Date(1609455600000+0100)/",
                "IsDistribution": false,
                "LeaveRequestCorretionId": null,
                "ManualWageRegistrationId": null,
                "NotAddedToBalance": false,
                "PlannedUnits": 0,
                "RemovedUnitsLastYear": 0,
                "RequestedUnits": 0,
                "SpentUnits": 0.0000,
                "SpentUnitsExtra": 0,
                "SpentUnitsLastYear": 1.0000,
                "WageCorrectionType": null
            },
            {
                "Balance": 321341249.0000,
                "Comment": null,
                "Correction": 0,
                "Date": "/Date(1609628400000+0100)/",
                "IsDistribution": false,
                "LeaveRequestCorretionId": null,
                "ManualWageRegistrationId": null,
                "NotAddedToBalance": false,
                "PlannedUnits": 0,
                "RemovedUnitsLastYear": 0,
                "RequestedUnits": 0,
                "SpentUnits": 0.0000,
```

```
                "SpentUnitsExtra": 0,
                "SpentUnitsLastYear": 1.0000,
                "WageCorrectionType": null
            }
        ],
        "DetailsTotalRow": {
            "Balance": 321341249.0000,
            "Comment": null,
            "Correction": 0,
            "Date": "/Date(1609455600000+0100)/",
            "IsDistribution": false,
            "LeaveRequestCorretionId": null,
            "ManualWageRegistrationId": null,
            "NotAddedToBalance": false,
            "PlannedUnits": 0,
            "RemovedUnitsLastYear": 0,
            "RequestedUnits": 0,
            "SpentUnits": 10.0000,
            "SpentUnitsExtra": 0,
            "SpentUnitsLastYear": 321341239.0000,
            "WageCorrectionType": null
        },
        "InitialBalance": 10.00,
        "InitialExtraBalance": 0,
        "InitialLastYearBalance": 321341241.00,
        "IsEditable": true,
        "IsLastYearBalanceCalculated": false,
        "ManualCorrections": 0,
        "PlannedUnits": 0,
        "RequestedUnits": 0,
        "SpentUnits": 2.0000,
        "SpentUnitsExtra": 0,
        "Tooltip": null,
        "TotalInitialBalance": 321341251.00
    }
]
```


l) Retrieve a list of absence types with absence balances

This endpoint retrieves a list of absence types with absence balances.

Request (JSON)

```
GET /Leave/LeaveRequestTypesWithAbsenceBalance
HTTP/1.1
```

```
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

```
Response(JSON)
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
    {
        "BlockRegisteringAbsenceOnSelf": false,
        "CanBeOngoing": false,
        "CustomPercentage": false,
        "CustomPercentages": null,
        "IsFullDay": true,
        "IsTimeSpan": true,
        "LeaveTypeId": 1,
        "Name": "Vacation"
    }
]
```

m) Create New Absence Request/Register Absence

When an employee wants to plan an absence, they must submit an absence request. When a planner wants to register absence, they use the same service. If a planner registers absence, there is the option to make shifts in the absence period available.

Parameter	Format	Example
employeeId	Guid	766c9732-e2d1-46d1-ae3e-a74c560bb8e6
DateFrom	DateTime	2011-09-02T12:37:00
DateTo	DateTime	2011-09-02T12:37:00
leaveTypeId	as specified by the leavetype service	1

Request (JSON)

```
POST /LeaveRequests/?makeShiftsVacant={makeShiftsVacant}/ HTTP/1.1
x-tamigo-token: <token>

{"EmployeeId":"<employeeId>", "DateFrom":"<DateFrom>","DateTo":"<dateTo>",
"LeaveTypeId":"<leaveTypeId>, "Comment":"<comment>"}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

n) List of Future Absences

This service returns a list of the registered absences for the logged-in employee.

Request (JSON)

```
GET /Leave/Future/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK

[{"LeaveRequestId":"00000000-0000-0000-0000-000000000000", "EmployeeId":
"00000000-0000-0000-0000-000000000000",
“EmployeeName”:”Jane Doe”,
"DateFrom":"\/Date(1316700000000+0200)\/",
"DateTo":"\/Date(1316700000000+0200)\/",
"LeaveTypeId":"1",
”LeaveTypeName”:”Vacation”}

...
{"LeaveRequestId":"00000000-0000-0000-0000-000000000000", "EmployeeId":
"00000000-0000-0000-0000-000000000000",
“EmployeeName”:”Jane Doe”,
"DateFrom":"\/Date(1316700000000+0200)\/",
"DateTo":"\/Date(1316700000000+0200)\/",
"LeaveTypeId":"1"
”LeaveTypeName”:”Vacation”}]
```

o) Approve or Deny Absence Requests

When a planner wishes to approve or deny an absence request, they are updating the corresponding absence request. This is available for planners only.

Request (JSON)

```
PUT /LeaveRequests/{requestId}/ HTTP/1.1
x-tamigo-token: <token>

Request (JSON)
{  "LeaveRequestId": "<requestId>",
"IsApproved": “<bool>”,
"MoveShiftsToVacant":” <bool>”,
"SmsComment": "<comment>"}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

p) List of Absence Categories

This service returns a list of the currently available absence categories for use by the leave request service. This is available for all roles.

If `employeeId` and `datefrom` are provided, the categories are limited for that employee.

Request (JSON)

```
GET /Leave/Groups/?employeeId={employeeId}&dateFrom={dateFrom}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK

[{"Id":"00000000-0000-0000-0000-000000000000", "Name":"Vacation"},
...
{"Id":"00000000-0000-0000-0000-000000000000", "Name":"Sick"}]
```

q) List of Absence Types

This service returns a list of the currently available absence types for use by the leave request service. This is available for all roles. Employees can only choose between a subset of absence types, as specified in tamigo.

If the `absenseGroupId` is supplied, the list will be filtered by the categories if all possible `leavetypes` are returned.

Request (JSON)

```
GET /Leave/LeaveRequestTypes/?employeeId={employeeId}&absenceGroupId=
{absenceGroupdId}")/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK

[{"LeaveTypeId":"00000000-0000-0000-0000-000000000000", "Name":"Vacation"},

{"LeaveTypeId":"00000000-0000-0000-0000-000000000000", "Name":"Sick"}]
```

r) List of Employees from Department

This service returns the list of employees you can register absence for.

Request (JSON)

```
GET /Leave/Employees/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Importing Opening Balances and Corrections for Absence

Absence opening balances import endpoint

Method: POST

URL:

```
"/AbsenceBalance/ImportStartingBalances/"
```

Parameters:

```
WageSystemKey (string)
AbsenceTypeShortName (string)
Year (int)
StartingBalance (decimal)
```

Request body example:

```
[{
  "WageSystemKey": "123",
  "AbsenceTypeShortName": "HOL",
  "Year": 2019,
  "StartingBalance": 9.0
},
{
  "WageSystemKey": "124",
  "AbsenceTypeShortName": "HOL",
  "Year": 2019,
  "StartingBalance": 10.0
 }]
```

Validation:

- WageSystemKey and AbsenceTypeShortName must not be empty or null
- WageSystemKey must exist in tamigo
- AbsenceTypeShortName must exist in tamigo

Additional notes:

Invalid elements will not be imported, and a response error message will appear for each of them. Others will be imported.

Absence corrections import endpoint

Method: POST

URL:

```
 "/AbsenceBalance/ImportAbsenceBalanceCorrections/"
```

Parameters:

```
WageSystemKey (string)
AbsenceTypeShortName (string)
CorrectionAmount (decimal)
```

```
DateOfCorrection (string)
Comment (string)
```

Request body example:

```
[{
  "WageSystemKey": "123",
  "AbsenceTypeShortName": "HOL",
  "CorrectionAmount": 1,
  "DateOfCorrection": "2019-01-15",
  "Comment": "test"
},
{
  "WageSystemKey": "124",
  "AbsenceTypeShortName": "HOL",
  "CorrectionAmount": 2,
  "DateOfCorrection": "2019-01-16",
  "Comment": ""
 }]
```

Validation:

- WageSystemKey, AbsenceTypeShortName, and DateOfCorrection must not be empty or null
- WageSystemKey must exist in tamigo
- AbsenceTypeShortName must exist in tamigo
- DateOfCorrection must be in the correct format (YYYY-MM-DD)
- Comment length must not be greater than 500 characters

Additional notes:

Invalid elements will not be imported, and a response error message will appear for each of them. Others will be imported. The endpoint uses the same existing logic in tamigo:

- if there is no correction on a given date, a new correction will be added
- if there is an existing correction on a given date, a correction will be updated
- if there is an existing correction on a given date and the correction amount is set to 0, the correction will be deleted

CSV File Format

The correct format of the CSV file is:

◦ Header row:
WageSystemKey;AbsenceTypeShortName;StartingBalance;Year;CorrectionAmount;DateOfCorre
◦ Absence balance rows should be after the header row.

◦ Delimiter: semicolon (";")

◦ One row can be used to import either absence opening balance, or absence balance correction, or both
◦ Example of absence balance row containing opening balance and correction:
1159319;HOL;15;2019;1;2019-01-20;test correction

◦ Example of absence balance row containing opening balance only:
1159319;HOL;15;2019;;;

◦ Example of absence balance row containing balance correction only:
1159319;HOL;;;2;2019-02-21;test correction

◦ Example with multiple rows is as follows:

	A	B	C	D	E	F	G	H
1	WageSystemKey	AbsenceTypeShortName	StartingBalance	Year	CorrectionAmount	DateOfCorrection	Comment	
2	1159319	HOL	15	2019				
3	1159319	HOL			2	2019-02-21	test correction 2	
4	1159319	HOL			3	2019-02-04		
5	1159319	HOL			1	2019-01-23	test correction 3	
6	1159945	HOL	12	2019	2	2019-01-21	test correction 1	
7	1159945	HOL			3	2019-01-25	test correction 2	
8								
9								
10								
11								

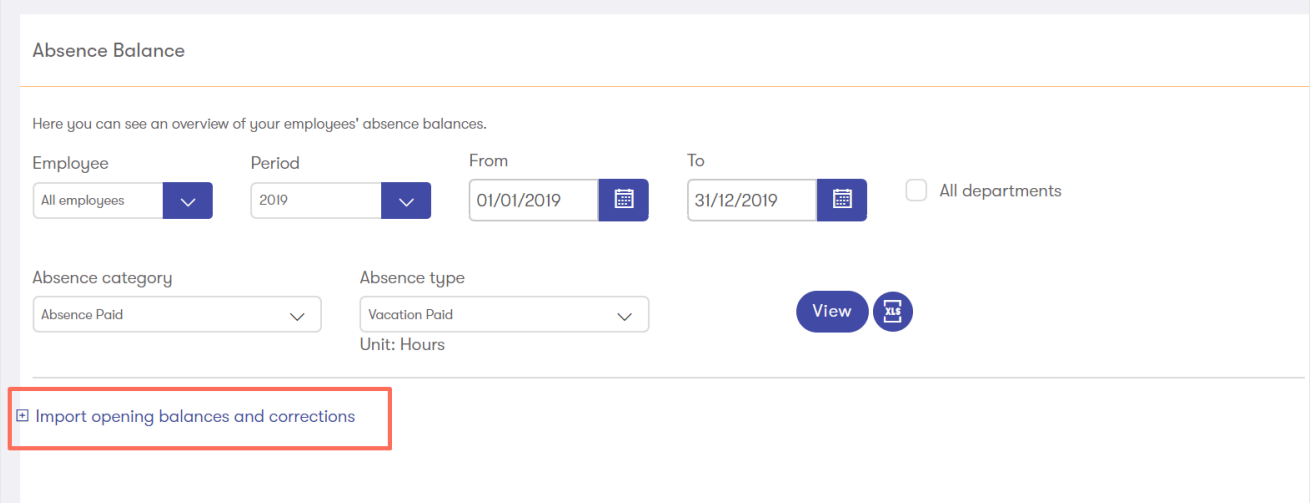
Yellow rows represent skipped data.

Example of the CSV import file:

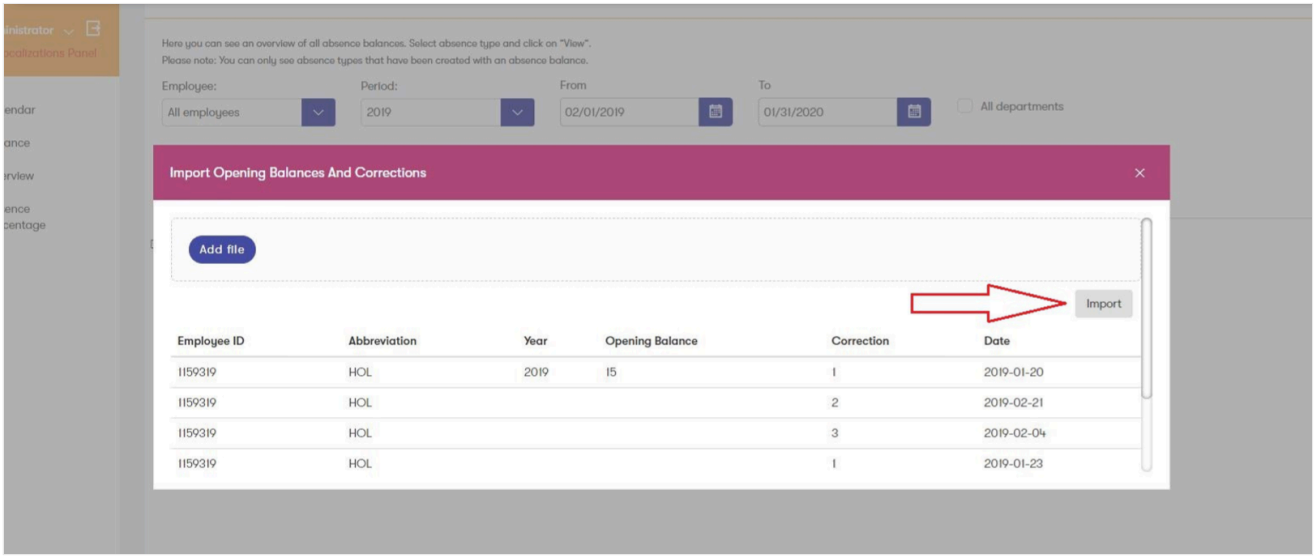
	A	B	C	D	E	F	G	H	I	J	K
1	WageSystemKey;AbsenceTypeShortName;StartingBalance;Year;CorrectionAmount;DateOfCorrection;Comment										
2	1159319;HOL;15;2019;;;										
3	1159319;HOL;;;2;2019-02-21;test correction 2										
4	1159319;HOL;;;3;2019-02-04;										
5	1159319;HOL;;;1;2019-01-23;test correction 3										
6	1159945;HOL;12;2019;2;2019-01-21;test correction 1										
7	1159945;HOL;;;3;2019-01-25;test correction 2										
8											
9											
10											

Importing Absence Balance

Importing absence opening balances and corrections can also be done on the *Absence* -> *Balance* page, without going through the API:



Clicking on the link will open the following popup window:

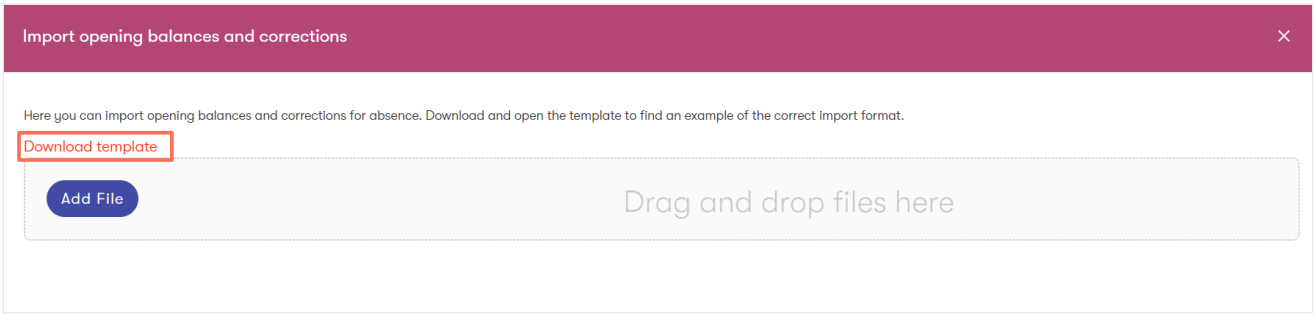


You can either:

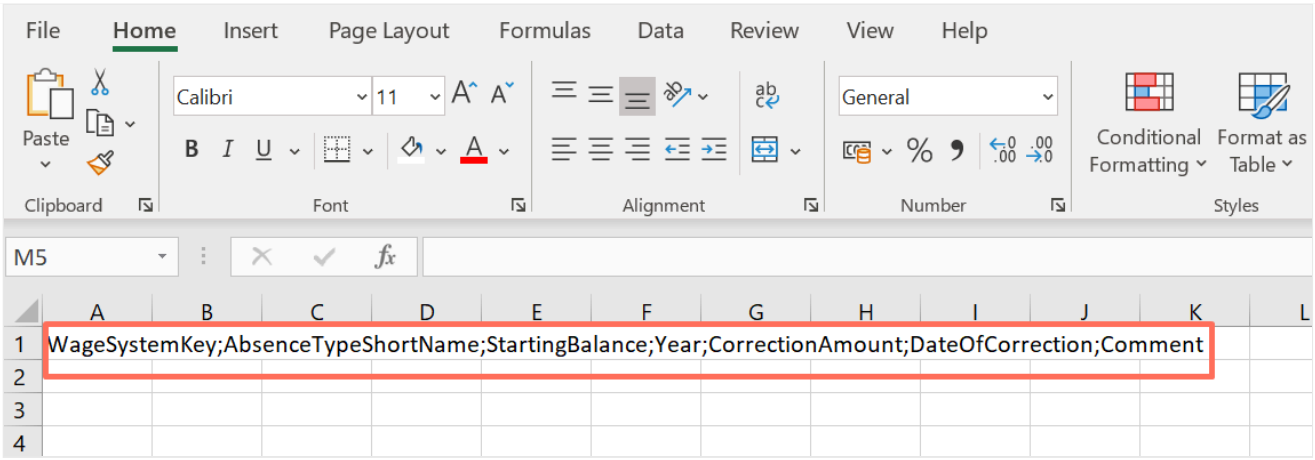
1. Prepare a CSV file beforehand as shown in the above section *CSV File Format*, then drag and drop that CSV file into this popup. When the CSV file is dragged and dropped in, all rows for import are shown in the table as above. To trigger the import, click on the *Import* button.

If there are any invalid rows, error messages are shown for each of them.

2. Click on the *Download template* link to download an Excel file template to fill out:



Inside the template will be the correct format you need to use to enter your data:



- Note that the WageSystemKey should be filled out as the employee ID.
- All lines must have at least the WageSystemKey and AbsenceTypeShortName columns filled in.
- If you are importing balances, lines must also have StartingBalance and Year filled in.
- If you are importing corrections, lines must have CorrectionAmount and DateOfCorrection filled in.

As above, fill in the template and save it, then drag and drop it into the popup window when you are finished. Click on *Import* to import the data. If any columns have not been filled out correctly, you will receive error messages at this point, informing you what to fix.

Importing Custom Fields

a) Get Custom Field History

```
{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/
```

Headers:

x-tamigo-token {{UserToken}}

Accept application/json

Request (JSON)

```
var settings = {
"url": "{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/",
"method": "GET",
"timeout": 0,
"headers": {
"x-tamigo-token": "{{UserToken}}",
"Accept": "application/json"
},
};
```

Response (JSON)

```
[
{
"CustomColumnId": 4,
"History": [
{
"CustomValue": "true",
"EndDate": null,
"Id": 9,
"StartDate": "/Date(1620338400000+0200)/"
}
]
```

b) Get Specific Custom Field History

```
{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/3
```

Headers:

x-tamigo-token {{UserToken}}

Accept application/json

Request (JSON)

```
var settings = {
"url": "{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/4",
"method": "GET",
"timeout": 0,
"headers": {
"x-tamigo-token": "{{UserToken}}",
"Accept": "application/json"
},
};
```

Response (JSON)

```
[
{
"CustomValue": "true",
"EndDate": null,
"Id": 9,
"StartDate": "/Date(1620338400000+0200)/"
```

```
}
]
```

c) Add to Specific Custom Field History

```
{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/3
```

Headers:

x-tamigo-token {{UserToken}}

Accept application/json

Body

```
{
"CustomValue": "abc",
"StartDate": "/Date({{date}})/",
"EndDate": null
}
```

Request (JSON)

```
var settings = {
"url": "{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/4",
"method": "POST",
"timeout": 0,
"headers": {
"x-tamigo-token": "{{UserToken}}",
"Accept": "application/json"
},
"data": "{\r\n \"CustomValue\": \"false\",\r\n \"StartDate\": \"
/Date({{date}})/\",\r\n \"EndDate\": null\r\n}",
};
```

Response (JSON)

```
{
"Message": null,
"Success": true
}
```

d) Delete Specific Custom Field History Value

```
{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/111/Value/10
```

Headers:

x-tamigo-token {{UserToken}}

Accept application/json

Request (JSON)

```
var settings = {
"url": "{{url}}/Employees/{{EmployeeId}}/CustomFieldHistory/4/Value/10",
"method": "DELETE",
"timeout": 0,
"headers": {
"x-tamigo-token": "{{UserToken}}",
"Accept": "application/json"
},
};
```

Response (JSON)

```
{
"Message": null,
"Success": true
}
```

KPI Services

Upload KPI Data

a) Standard and Custom KPIs

This service makes it possible to upload KPI data to tamigo for displaying throughout tamigo. This service requires you to log in as an application.

Description of options

KpiStatusType:

Actual = 1,

Forecast = 2,

Budget = 3

KpiType:

Revenue = 1,

Customers = 2,

Vacation = 3,

Hours = 4,

ProductiveHours = 100,

DailyRevenue = 101,

DailyHours = 102,

Wage = 103,

WageIndex = 104,

HeadCount = 105,

DailyRevenueIncludingVat = 106

For other types of KPI data, contact us directly. Note that DailyRevenue is 'Daily Revenue Excluding VAT' in the UI, and DailyRevenueIncludingVat is 'Daily Revenue Including VAT'.

KpiPeriodType:

Quarter = 1,

Half hour = 2,

Hour = 3,

Day = 4,

Week = 5,

Month = 6

Amount: Amount

WageSystemKey: Employee ID in tamigo

DepartmentId: Store ID in tamigo

Start: Date of KPI - please note: if for example importing monthly KPI (KpiPeriodType =6) then date of KPI must be on first day of month. Same rule applies for weekly and daily.

CompetencyId: This is internal id of a skill

CompetencyExternalId: This is the external skill id that can be set by the user in the configuration of a skill as external skill id

Request (JSON)

```
POST /KPI/Upload/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>

Body
[
    {
        "DepartmentId": "56989851",
        "WageSystemKey": "123",
        "Amount": 1500.13,
        "Start": "/Date(1607324400000)/",
        "KpiType": 1,
        "KpiStatusType": 3,
        "KpiPeriodType": 3,
        "CompetencyId": "98cb2210-a1a0-4024-b2d8-8abd673732da",
        "CompetencyExternalId": "SkillID",
    },
...
]
```

b) Budget Revenue, Actual Revenue, Footfall

This service makes it possible to upload KPI data to tamigo for displaying throughout tamigo. For example, whether hourly budget can be shown in *Edit Day* on an hourly basis. This service requires you to log in as an application.

For other types of KPI data, contact us directly.

Request (JSON)

```
POST /KPI/Upload/{status}/{type}/{period}/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>

 Status: Actual = 1 Forecast = 2 Budget = 3
 Type: Revenue = 1 Customers = 2
 Period (Time interval): Quarter = 1 Half hour = 2 Hour = 3 Day = 4
 Week = 5 Month = 6
```

```
Body
[
 "DepartmentId": "10",
  "Amount": 560.44,
  "Start": "/Date(1412589803535+0200)/"
 },
{
 "DepartmentId": "12",
  "Amount": 410.24,
  "Start": "/Date(1412589803535+0200)/"
 },
   ...
]
```

Previous article

KPI Reporting Data Flow

Next article

How to create API keys and use claims

Related articles

Web Services (API)

How to Fill in Your Templates

KPI Importer plugin

Calendar Events Importer

HR Diff Importer

Recently viewed articles

User Services - Web Services (API)

Web Services (API)

Setting Up Automatic Shifts

Setting Up Absence Balance

Absences