
User Services - Web Services (API)

- User Services
- Menu Services
- Bid Services
- Shift Exchange Services
- News Services
- Shift Activity Services
- Absence Services
- Request Services
- Contact Services
- Department Services
- Revenue Over Day Services
- iCal Services
- Calendar Comments Services
- Role Services
- Company Services
- Break Code Services
- Shift Services
- Actual Shift Services
- Services for Partner Login
- Touch Services for Smartphones
- Wage Percentage Services
- Document Services
- Financials Services

User Services

a) Login Services

b) User Login

For user services, you need to log in as a user of tamigo, providing the email and password for authentication. The response returned from the service contains the `SessionToken` key. Use this token in subsequent calls to the service to authenticate yourself.

Request (JSON)

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

```
{"Email": "user@company.com",
  "Password": "password",
  "DefaultCompanyId": "00000000-0000-0000-0000-000000000000"}
```

Note: The DefaultCompanyId is an optional parameter. If you leave the DefaultCompanyId blank, the service log ins to the default department.

Request (XML)

```
<Login xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Email>[Email]</Email>
  <Password>[Password]</Password>
</Login>
```

Response (JSON)

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

```
{"DefaultDepartmentId": "00000000-0000-0000-0000-000000000000",
  "DefaultDepartmentName": "Department A",
  "DefaultCompanyId": "00000000-0000-0000-0000-000000000000",
  "DefaultCompanyName": "Company A",
  "Email": "user@company.com",
  "Password": "",
  "Role": "Planner",
  "SessionToken": "1fcbace9-28f4-4e76-9917-6f0ebc98da64"}
```

Response (XML)

```
<Login xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <DefaultDepartmentName>[DepartmentName]</DefaultDepartmentName>
  <Email>[Email]</Email>
  <Password/><Role>[Role]</Role><SessionToken>[SessionToken]</SessionToken>
</Login>
```

c) Get Token

This service returns the token information. This is available for all roles.

Request (JSON)

```
GET /Login/ 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

{"DefaultDepartmentId":"00000000-0000-0000-0000-000000000000",
"DefaultDepartmentName":"Department A",
"DefaultCompanyId":"00000000-0000-0000-0000-000000000000",
"DefaultCompanyName":"Company A",
"Email":"user@company.com",
"Password":"",
"Role":"Planner",
"SessionToken":"00000000-0000-0000-0000-000000000000"}
```

Menu Services

Mobile Menu

This service returns a list of menu items on the app. The menu that will be returned is determined by the `CompanyId` key and role from the token.

The menu contains headers (`ParentId = null`) and an item (`ParentId` is equal to header ID).

The list is automatically sorted in the correct order by the service.

The `Version` key determines what the menu will include. Currently, the highest version of a menu is **2**.

Request (JSON)

```
GET /Menu/mobile/?Version={version}/ 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
```

```
"[{"DefaultChildId":null,"Description":"","Enabled":true,"Id":1024,
"OrderIndex":1,
```

Bid Services

The bid services give the user a view on shifts that have been made available, and they enable bidding for these shifts.

a) Get Available Shifts

This service returns a list of shifts that are available to bid on. This service is available for all roles.

Request (JSON)

```
GET /bids/Available/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK

[{"BreakCode":"Q","Comment":"","
"DepartmentName":"Department A","EmployeeName":"","
"EndTime":"\\/Date(1316700000000+0200)\\/",
"StartTime":"\\/Date(1316671200000+0200)\\/"},
...
{"BreakCode":"Q","Comment":"","
"DepartmentName":"Department A","EmployeeName":"","
"EndTime":"\\/Date(1316700000000+0200)\\/",
"StartTime":"\\/Date(1316671200000+0200)\\/"}]
```

Response (XML)

```
"<ArrayOfAvailableShift
xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<AvailableShift>
<BidPlaced>>false</BidPlaced>
<BreakCode>Q</BreakCode>
<Comment/>
<DepartmentId>00000000-0000-0000-0000-000000000000</DepartmentId>
```

```
<DepartmentName>Department A</DepartmentName>
<EmployeeHaveShiftSameDay>>false</EmployeeHaveShiftSameDay>
<EndTime>2011-12-21T16:00:00</EndTime>
<ExtendedShiftEndTime>0001-01-01T00:00:00</ExtendedShiftEndTime>
<ExtendedShiftStartTime>0001-01-01T00:00:00</ExtendedShiftStartTime>
<IsExtended>>false</IsExtended>
<OtherEmployeesHaveBidOnShift>>false</OtherEmployeesHaveBidOnShift>
<ShiftId>dbdb767f-1415-4cdb-8763-1a7c2281f1b8</ShiftId>
<ShiftProviderId>00000000-0000-0000-0000-000000000000</ShiftProviderId>
<ShiftProviderName>Available</ShiftProviderName>
<StartTime>2011-12-21T08:00:00</StartTime>
<Week>51</Week>
</AvailableShift>
</ArrayOfAvailableShift>"
```

b) Bid for Available Shifts

Use this for when a logged-in employee bids on an available shift.

This is only available for employees. Planners cannot bid on shifts.

Request (JSON)

```
PUT /Shifts/Available/{shiftId}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
<HTTP/1.1 200 OK
```

c) Accept Bids on Available Shifts

Use this service to decide on a bid for an available shift. This is only available to Planners.

Use the `IsAccepted` property to approve or deny the request.

Request (JSON)

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

```
{
  "RequestId": "00000000-0000-0000-0000-000000000000",
```

```
"IsAccepted": true
}
```

Request (XML)

```
<ShiftRequest
xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <ProviderId>00000000-0000-0000-0000-000000000000</ProviderId>
  <RequestId>00000000-0000-0000-0000-000000000000</RequestId>
  <StartDateTime>01-01-0001 00:00:00</StartDateTime>
  <EndDateTime>01-01-0001 00:00:00</EndDateTime>
  <ShiftHours>0</ShiftHours>
  <RequesterId>00000000-0000-0000-0000-000000000000</RequesterId>
  <IsAccepted>True</IsAccepted>
</ShiftRequest>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

d) Withdraw Available Shifts

Use this service to withdraw a shift if it has been made available. The bid cannot be withdrawn if other employees have bid on the shift.

Request (JSON)

```
PUT /bids/withdraw/{ShiftId}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

Shift Exchange Services

The shift exchange services can be used to make exchanges between two shifts, as opposed to the bid services, where there is only one shift involved.

a) Create New Shift Exchange

This creates a new shift exchange. This is available for employees only. Planners cannot exchange shifts.

Request (JSON)

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

```
{"ProposerShiftId":"<shiftId1>","AcceptorShiftId":"<shiftId2>"}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

b) List of Shifts to Exchange With

This service returns a list of shifts that the current shift can be exchanged with. This is available for employees only.

Request (JSON)

```
GET /Exchanges/Available/?shiftId={shiftIdToExchange}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
[{"EmployeeId":"00000000-0000-0000-0000-000000000000",
  "Name":"Jane Doe",
  "HaveShifts":"false",
  "BreakCode":"Q",
  "EndDateTime":"\\/Date(1316700000000+0200)\\/\"",
  "StartDateTime":"\\/Date(1316671200000+0200)\\/\"",
  "Comment":"Close",
  "CanExchangeShift":"false"},
  ...
  {"EmployeeId":"00000000-0000-0000-0000-000000000000",
  "Name":"Jane Doe",
  "HaveShifts":"false",
  "BreakCode":"Q",
  "EndDateTime":"\\/Date(1316700000000+0200)\\/\"",
  "StartDateTime":"\\/Date(1316671200000+0200)\\/\"",
  "Comment":"Close",
  "CanExchangeShift":"true"}]
```

c) Get List of Pending Approvals (Employee and Planner)

You can get a list of pending shift exchange approvals.

An employee will get a list of shift exchanges that they need to approve or deny – this means that the employee approves or denies that they would like to exchange the selected shifts with another employee.

A planner will get a list of shift exchanges that have been approved by the employee, and they need to approve the shift exchange to finalize it.

The `Violate11HourRule` property is only available when the planner has approved the request. If the property is `True`, there will be an alert.

Request (JSON)

```
GET /Exchanges/Requests/ HTTP/1.1
x-tamigo-token: <token>
```

Response (XML)

```
<ArrayOfShiftExchangeRequest
xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <ShiftExchangeRequest>
    <Comment i:nil="true"/>
    <EmployeeId1>00000000-0000-0000-0000-0000000000008</EmployeeId1>
    <EmployeeId2>00000000-0000-0000-0000-000000000000</EmployeeId2>
    <EmployeeName1>Jane Doe</EmployeeName1>
    <EmployeeName2>John Doe</EmployeeName2>
    <EndDateTime1>2011-12-24T16:00:00</EndDateTime1>
    <EndDateTime2>2011-12-23T16:00:00</EndDateTime2>
    <IsAccepted i:nil="true"/>
    <RequestId>00000000-0000-0000-0000-000000000000</RequestId>
    <ShiftHours1>7.25</ShiftHours1>
    <ShiftHours2>7.25</ShiftHours2>
    <StartDateTime1>2011-12-24T08:00:00</StartDateTime1>
    <StartDateTime2>2011-12-23T08:00:00</StartDateTime2>
    <Violate11HourRule>False</Violate11HourRule>
    <ShiftComment1>Opening</ ShiftComment1>
    < ShiftComment2>Closing</ ShiftComment2>
    < ShiftBreakCode1>P</ ShiftBreakCode1>
    < ShiftBreakCode2>Q</ ShiftBreakCode2>
  </ShiftExchangeRequest>
</ArrayOfShiftExchangeRequest>
```

d) Approve or Deny a Shift Exchange

Approve or deny a shift exchange.

The approval is a two-step process:

- Approval by the employee who is receiving the shift exchange request
- Approval by the planner

This is available for all roles.

Request (JSON)

```
PUT /Exchanges/Requests/{requestId}/{isAccepted}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

e) Withdraw a Shift Exchange

Use this service to withdraw a shift exchange if the other employee has not approved it yet.

Request (JSON)

```
PUT /Exchanges/withdraw/{shiftId}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

News Services

Get Company and Department News

This service returns the content from the front page in tamigo. The content is returned as HTML. This service is available to all roles.

Request (JSON)

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

Response (JSON)

```
HTTP/1.1 200 OK
Content-Type: text/html
```

```
<html><body>
<h1>{departmentNews}</h1>
<div id="departmentNews">
{content}
</div>
<h1>{companyNews}</h1>
<div id="companyNews">
{content}
</div>
```

Shift Activity Services

Get Shift Activities

The shift activity service returns a list of all the company's activities. It requires user login, so the list is for the logged-in user and the default department of that user.

Request (JSON)

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

Response (JSON)

```
HTTP/1.1 200 OK "true"

[{"ShiftActivityId":"00000000-0000-0000-0000-000000000000", "Code":" 1",
"Name":"Evening shift", "Description":"", "Enabled":"True", "OrderIndex":"2"},
...
{" ShiftActivityId ":"00000000-0000-0000-0000-000000000000", " Code ":"0",
"Name":"Inventory", "Description":"", "Enabled":"
```

Request Services

The request services offer the planner an option to approve or deny bids from their employees.

a) Pending Bids for Available Shifts

This service returns a list of pending bids. This is only available for planners.

Request (JSON)

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

Response (JSON)

```
[{
  "Comment": "",
  "EndDateTime": "\/Date(1324220400000+0100)\/",
  "IsAccepted": null,
  "RequestId": "00000000-0000-0000-0000-000000000000",
  "RequesterId": "00000000-0000-0000-0000-000000000000",
  "RequesterName": "John Doe",
  "ShiftHours": 7.25,
  "StartDateTime": "\/Date(1324191600000+0100)\/",
  "ProviderId": "00000000-0000-0000-0000-000000000000",
  "ProviderName": "Available",
  "Violate11HourRule": "false"
},
...
{
  "Comment": "",
  "EndDateTime": "\/Date(1324220400000+0100)\/",
  "IsAccepted": null,
  "RequestId": "00000000-0000-0000-0000-000000000000",
  "RequesterId": "00000000-0000-0000-0000-000000000000",
  "RequesterName": "John Doe",
  "ShiftHours": 7.25,
  "StartDateTime": "\/Date(1324191600000+0100)\/",
  "ProviderId": "00000000-0000-0000-0000-000000000000",
  "ProviderName": "Available",
  "Violate11HourRule": "false"
}
```

```
}]
```

b) Approve or Deny Bids for Shifts

This approves or denies a bid or shift exchange. This is only available for planners and administrators.

Request (JSON)

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

```
{"RequestId": "<requestID>", "Approved": <true/false>}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Contact Services

The contact services return a list of contact information for employees and planners in the current department.

Contact List

This service returns a list of available contacts, including email addresses and phone numbers.

Request (JSON)

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

Response (JSON)

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

```
[{"Email": "user@company.com", "FirstName": "Jane", "ImageUrl": "\\images\\small\\Jane_Untitled.png", "LastName": null, "MobilePhone": "+4511111111"},  
...]
```

```
{"Email":"user2@company.com","FirstName":"John","ImageUrl":"","LastName":null,"MobilePhone":"+4511111111"}]
```

Department Services

The department services are used for operations to get a list of departments for the company and to update the default department for a user.

a) All Departments

Get a list of the currently logged-in user's departments.

Request (JSON)

```
GET /Departments/ 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":"00000000-0000-0000-0000-000000000000",
  "IsDefault":false,"Name":"Department A"},
  ...
{"DepartmentId":"00000000-0000-0000-0000-000000000000",
  "IsDefault":false,"Name":"Clothing"}]
```

b) Set Default Department

In our iPhone app, you change the department that you are currently looking at by setting the default department.

Request (JSON)

```
PUT /departments/00000000-0000-0000-0000-000000000000/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
{"DepartmentId":"00000000-0000-0000-0000-000000000000","IsDefault":true}
```

Response (JSON)

HTTP/1.1 200 OK

Revenue Over Day Services

The revenue over day services are used to upload data for forecasting purposes.

AmountTypeId 1 is Revenue and 2 is Customers.

StatusTypeId 1 is Actual data and 2 is Forecast data. 2 can be used to upload your own forecast data.

Note that the types have to exist for the upload to be successful.

Note that the time span for a single unit has to be less than 24 hours, and more than 15 minutes. The upload will fail if the span cannot be divided into 15-minute spans.

a) Upload Single Data Unit

Upload actual or forecast data for a single department in a period.

Request (JSON)

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

```
{"Amount": "<float>", "AmountTypeId": "[1|2]", "End": "<DateTime>", "PosStoreId": "<string>", "Start": "<DateTime>", "StatusTypeId": "[1|2]"}
```

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

b) Upload List of Data Units

Upload actual or forecast data for multiple departments for multiple periods.

Request (JSON)

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

```
[{"Amount": "<float>", "AmountTypeId": "[1|2]", "End": "<DateTime>", "PosStoreId": "<string>", "Start": "<DateTime>", "StatusTypeId": "[1|2]"},
```

```
...
{"Amount": "<float>", "AmountTypeId": "[1|2]", "End": "<DateTime>", "PosStoreId": "<string>", "Start": "<DateTime>", "StatusTypeId": "[1|2]"}
```

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

iCal Services

The iCal services return the user's iCal link. This is their personal calendar in iCal format.

a) Get iCal Link

Request (JSON)

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

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

b) Upload List of Data Units

Use this service to upload actual or forecast data for multiple departments for multiple periods.

Request (JSON)

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

```
Result: "https://services.tamigo.com/Calendar/00000000-0000-0000-0000-000000000000/Calendar.ics"
```

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

Calendar Comments Services

This service returns the comments that are attached to a day in the rota.

a) Get Calendar Comments

Request (JSON)

```
GET /comments/day/{date}/ HTTP/1.1
Content-Type:application/json
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

b) Upload List of Data Units

Use this service to upload actual or forecast data for multiple departments for multiple periods.

Request (JSON)

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

```
Result: "https://services.tamigo.com/Calendar\
/00000000-0000-0000-0000-000000000000/Calendar.ics"
```

Response (JSON)

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

```
{
  "CompanyCommentShort":"Short message",
  "DepartmentCommentLong":"Long message!",
  "DepartmentCommentShort":"Short message"
}
```

c) Import Calendar Comments

You can create or update existing calendar comments in tamigo.

- Existing comments will be overwritten (not added)
- AllSubDepartments indicates if you wish to create comments for subdepartments of the department

Request (JSON)

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

```
[
  {
    "Date": "/Date(1605139200000)/",
    "DepartmentKey": "StoreId",
    "DepartmentComment": "Department comment",
    "DepartmentCommentColor": "#FFB969",
    "AllSubDepartments": true,
    "CompanyComment": "Company comment",
    "CompanyCommentColor": "#FFB969"
  }
]
```

Response (JSON)

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

```
[
  {
    "Message": "Comment added for department id: 3 on date 12/11/2020",
    "Success": true
  },
  {
    "Message": "Comment added for company on date 12/11/2020",
    "Success": true
  }
]
```

Note that 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.

d) Remove Calendar Comments

You can create or update existing calendar comments in tamigo.

- Existing comments will be deleted
- The allDepartments field indicates if comments are being removed for the department or entire company on the selected dates
- AllSubDepartments indicates if you wish to remove comments for subdepartments of the department

Request (JSON)

```
DELETE /Comments/RemoveCalendarComments/?allDepartments={allDepartments}
HTTP/1.1
Accept: application/json
Content-Type: application/json
x-tamigo-token: <token>
```

```
[
  {
    "Date": "/Date(1605139200000)/",
    "AllSubDepartments": true,
    "DepartmentKey": "StoreId"
  }
]
```

Response (JSON)

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

```
[
  {
    "Message": "Comment deleted for department id: 3 on date 12/11/2020",
    "Success": true
  }
]
```

Note that 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.

Role Services

The role services returns the roles that the user has in the current department.

a) Get Roles

Request (JSON)

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

Response (JSON)

```
HTTP/1.1 200 OK "true"
[{"Role 1"}, {"Role 2"}, {"Role 3"}]
```

b) Set Role

The set role service is used for setting in the token the new role selected by the user.

Request (JSON)

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

Body
"planner"

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

Company Services

The company service returns all the user's companies.

Get Employee Companies

Request (JSON)

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

Response (JSON)

```
HTTP/1.1 200 OK "true"
```

```
[{"CompanyId":"00000000-0000-0000-0000-000000000000", "Name":"Company Name"},  
...  
{"CompanyId ":"00000000-0000-0000-0000-000000000000", "Name":"Company Name "}]
```

Break Code Services

The break code service returns all the company's break codes (including custom and standard).

Get Break Codes

Request (JSON)

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

Response (JSON)

```
HTTP/1.1 200 OK "true"  
  
[{"BreakCodeId":"00000000-0000-0000-0000-000000000000", "Code":"Q", "Name":"Q",  
"Description":"BreakCode1", "Hours":"5", "MinimumHours":"5"}],  
...  
{"BreakCodeId":"00000000-0000-0000-0000-000000000000", "Code":"P", "Name":"P",  
"Description":"Break
```

Shift Services

The shift services can be used to get information on rotas and upcoming shifts.

a) My Overview

This service returns a list of the logged in employees' upcoming shifts. Dates are expressed as milliseconds since EPOCH (1970-1-1 00:00:00). This is available for all roles.

Request (JSON)

```
GET /shifts/future/ 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
```

```
[{"BreakCode":"Q",
"Comment":"","
"DepartmentName":"Department A",
"EmployeeName":null,
"EndTime":"\\/Date(1318860000000+0200)\\/\"",
"StartTime":"\\/Date(1318831200000+0200)\\/\"},
...
{"BreakCode":"Q",
"Comment":"","
"DepartmentName":"Department A",
"EmployeeName":null,
"EndTime":"\\/Date(1318946400000+0200)\\/\"",
"StartTime":"\\/Date(1318917600000+0200)\\/\"}]
```

Response (XML)

```
<ArrayOfShift xmlns="http://schemas.datacontract.org/2004/07
/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Shift>
<BreakCode>Q</BreakCode>
<Comment/>
<DepartmentName>Department A</DepartmentName>
<EmployeeName i:nil="true"/>
<EndTime>2011-12-19T06:00:00</EndTime>
<IsAvailable>>false</IsAvailable>
<IsExchange>>false</IsExchange>
<ShiftId>00000000-0000-0000-0000-000000000000</ShiftId>
<StartTime>2011-12-18T23:00:00</StartTime>
</Shift>
</ArrayOfShift>
```

b) Approved Rota – Current Day

This service returns a list of the shifts in the approved rota for the day, including employees who are absent, as indicated by the activity on the day. This is available for all roles. If the department is equal to all , shifts from all rotas in all departments are returned.

Request (JSON)

```
GET /shifts/today/(?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

[{"BreakCode":"Q","Comment":"","
  "DepartmentName":"Department A","EmployeeName":"","
  "EndTime":"\\/Date(1316700000000+0200)\\/","
  "StartTime":"\\/Date(1316671200000+0200)\\/"},
  ...
{"BreakCode":"Q","Comment":"","
  "DepartmentName":"Department A","EmployeeName":"","
  "EndTime":"\\/Date(1316700000000+0200)\\/","
  "StartTime":"\\/Date(1316671200000+0200)\\/"}]
```

Response (XML)

```
"<ArrayOfShift xmlns="http://schemas.datacontract.org/2004/07
/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Shift>
    <BreakCode>Q</BreakCode>
    <Comment/>
    <DepartmentName>Department A</DepartmentName>
    <EmployeeName/>
    <EndTime>2011-12-13T16:00:00</EndTime>
    <StartTime>2011-12-13T08:00:00</StartTime>
  </Shift>
</ArrayOfShift>"
```

c) Approved Rota – Day Other Than Current Day

This service returns a list of shifts in the approved rota for the day, including employees that are absent, as indicated by the activity on the day. This is available for all roles. If the department is equal to `all`, shifts from all rotas in all departments are returned.

Request (JSON)

```
GET /shifts/day/{date}/(?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

[{"BreakCode":"Q","Comment":"","
  "DepartmentName":"Department A","EmployeeName":"","
  "EndTime":"\\/Date(1316700000000+0200)\\/","
  "StartTime":"\\/Date(1316671200000+0200)\\/"},
  ...
{"BreakCode":"Q","Comment":"","
  "DepartmentName":"Department A","EmployeeName":"","
  "EndTime":"\\/Date(1316700000000+0200)\\/","
  "StartTime":"\\/Date(1316671200000+0200)\\/"}]
```

d) Approved Rota – Period of Dates per Employee

This service returns a list of the shifts in the approved rota for the selected period of dates for the selected employee. This is available for all roles. If the department is equal to `all`, shifts from all rotas in all departments are returned.

Request (JSON)

```
GET /shifts/period/{startDate}/{endDate}/(?departmentId={departmentId})
&employeeId=employeeId/ 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

[{"BreakCode":"Q","Comment":"","
  "DepartmentName":"Department A","EmployeeName":"","
  "EndTime":"\\/Date(1316700000000+0200)\\/","
  "StartTime":"\\/Date(1316671200000+0200)\\/"},
  ...
{"BreakCode":"Q","Comment":"","
```

```
"DepartmentName":"Department A","EmployeeName":"","  
"EndTime":"\\/Date(1316700000000+0200)\\/","  
"StartTime":"\\/Date(1316671200000+0200)\\/"}]
```

Response (XML)

```
"<ArrayOfShift xmlns="http://schemas.datacontract.org/2004/07  
/Tamigo.Services.Entities"  
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
<Shift>  
<BreakCode>Q</BreakCode>  
<Comment/>  
<DepartmentName>Department A</DepartmentName>  
<EmployeeName/>  
<EndTime>2011-12-13T16:00:00</EndTime>  
<StartTime>2011-12-13T08:00:00</StartTime>  
</Shift>  
</ArrayOfShift>"
```

e) Get All Types of Shifts – Original, Planned, and Worked Shifts

This service returns a list of all types of shifts for the selected period of dates. This is available only when you are logged in as an application. If the department is equal to all, shifts from all rotas in all departments are returned.

If the includeDraft query parameter is set to true, draft shifts will be included in the response. Draft shift will have the 'IsDraft' parameter set to true inside the 'PlannedShift' object.

Request (JSON)

```
GET /shifts/all/{startDate}/{endDate}/(?departmentId={departmentId}  
&includeDraft={true|false})  
/ HTTP/1.1  
Content-Type: application/json  
x-tamigo-token: <token>
```

Response (JSON)

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

```
[{
```

```

    "Date": "/Date(1453071600000+0100)/",
    "DepartmentId": "00000000-0000-0000-0000-000000000000",
    "DepartmentKey": "835",
    "DepartmentName": "Department A",
    "EmployeeId": "00000000-0000-0000-0000-000000000000",
    "FirstName": null,
    "EmployeePosKey": "E100",
    "Email": "user@company.com",
    "OriginalShift": {
    "ActivityName": null,
    "Comments": "",
    "Shift": 7.25,
    "ShiftActivityId": null,
    "Sum": null
    "IsDraft": null,
    },
    "PlannedShift": {
    "ActivityName": null,
    "Comments": "",
    "Shift": 7.25,
    "ShiftActivityId": null,
    "Sum": null
    "IsDraft": true,
    },
    "ActualShift": {
    "ActivityName": null,
    "Comments": null,
    "Shift": null,
    "ShiftActivityId": null,
    "Sum": null
    "IsDraft": null,
    },
    "WageNumber": null
  }
]

```

Response (XML)

```

"<ArrayOfEmployeeShift xmlns="http://schemas.datacontract.org/2004/07
/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <EmployeeShift>
    <Date>2016-01-05T00:00:00</Date>
    <DepartmentId>00000000-0000-0000-0000-000000000000</DepartmentId>
    <DepartmentKey>835</DepartmentKey>
    <DepartmentName>Department A</DepartmentName>

```

```

<EmployeeId>00000000-0000-0000-0000-000000000000</EmployeeId>
<FirstName i:nil="true"/>
<OriginalShift>
  <ActivityName i:nil="true"/>
  <Comments/>
  <Shift>9.00</Shift>
  <ShiftActivityId i:nil="true"/>
  <Sum i:nil="true"/>
  <IsDraft i:nil="true"/>
</OriginalShift>
<PlannedShift>
  <ActivityName i:nil="true"/>
  <Comments/>
  <Shift>9.00</Shift>
  <ShiftActivityId i:nil="true"/>
  <Sum i:nil="true"/>
  <IsDraft>true</IsDraft>
</PlannedShift>
<ActualShift>
  <ActivityName i:nil="true"/>
  <Comments i:nil="true"/>
  <Shift i:nil="true"/>
  <ShiftActivityId i:nil="true"/>
  <Sum i:nil="true"/>
  <IsDraft i:nil="true"/>
</ActualShift>

  <WageNumber i:nil="true"/>
</EmployeeShift>

```

f) Get Relevant Planned Shifts

This service returns the most relevant planned shift for the employee, meaning the shift that has not ended yet.

Request (JSON)

```

GET /shifts/planned/relevant/ 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

{"BreakCode":"Q","Comment":"","

```

```
"DepartmentName":"Department A","EmployeeName":"","  
"EndTime":"\\/Date(1316700000000+0200)\\/",  
"StartTime":"\\/Date(1316671200000+0200)\\/"},  
...  
{"BreakCode":"Q","Comment":"","  
"DepartmentName":"Department A","EmployeeName":"","  
"EndTime":"\\/Date(1316700000000+0200)\\/",  
"StartTime":"\\/Date(1316671200000+0200)\\/"}
```

Response (XML)

```
<Shift xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Services.Entities"  
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
<BreakCode>Q</BreakCode>  
<Comment/>  
<DepartmentName>Department A</DepartmentName>  
<EmployeeName/>  
<EndTime>2011-12-13T16:00:00</EndTime>  
<StartTime>2011-12-13T08:00:00</StartTime>  
</Shift>
```

g) Get shifts for a period and a department

This service returns a list of all shifts for the selected period and department. This is available only when you are logged in as an application. If the department is left empty, shifts from all departments are returned. The timespan limitation of this endpoint is 60 days.

Request (JSON)

```
GET /shifts?from={startDate}&to={endDate}&departmentId={departmentId}/ HTTP/1.1  
Content-Type: application/json  
x-tamigo-token: <token>
```

Response (JSON)

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

```
[  
  {  
    "AbsencePercentage": null,  
    "ActivityId": null,
```

```

    "BreakCodeId": null,
    "BreakStartTime": null,
    "Comments": null,
    "Comments2": null,
    "CompetencyId": "2fb82130-8487-472c-b3f8-e07b6d47335e",
    "CreatedOn": "/Date(1654081740353+0200)/",
    "EmployeeId": null,
    "EndDateTime": "/Date(1654073100000+0200)/",
    "ForceSave": false,
    "IsApprovedByEmployee": false,
    "IsAvailable": true,
    "IsCalculated": false,
    "IsExchange": false,
    "OrganizationalUnitId": "883dacb4-6bd3-4a0d-9faf-dadd79c8be48",
    "PlanId": "c28aa35c-3e5c-47fd-9d87-fb713e2af67c",
    "PlanStatusCodeId": 1,
    "ShiftActivityId": null,
    "ShiftHours": 5.7500,
    "ShiftId": "5af61d0d-f504-4665-b6e8-d49a598dcd69",
    "SkillExternalId": "SK1",
    "StartDateTime": "/Date(1654052400000+0200)/",
    "BreakCode": "",
    "BreakCodeHours": 0.000000,
    "DepartmentId": "883dacb4-6bd3-4a0d-9faf-dadd79c8be48",
    "DepartmentKey": "C1",
    "DepartmentName": "Child 1",
    "EmployeeName": "",
    "IsProductive": true,
    "ShiftActivityName": "",
    "ShiftActivityShortName": "",
    "WageSystemKey": ""
  }
]

```

The PlanStatusCode can have these values:

Draft = 1,
 Authorized = 2,
 Actual = 3,
 Original = 4,
 Closed = 5,
 Attendance = 6

h) Available Shifts

This service is deprecated—use the *Bid* service instead. This service is only available for backwards compatibility. It returns a list of shifts that are available. This service is available for all roles.

Request (XML)

```
GET /Shifts/available/(?departmentId={departmentId})/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response(XML)

```
<ArrayOfAvailableShift xmlns="http://schemas.datacontract.org/2004/07
/Tamigo.Services.Entities"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<AvailableShift>
<BidPlaced>false</BidPlaced>
<BreakCode>Q</BreakCode>
<Comment/><DepartmentId>00000000-0000-0000-0000-000000000000</DepartmentId>
<DepartmentName>Department A</DepartmentName>
<EmployeeHaveShiftSameDay>false</EmployeeHaveShiftSameDay>
<EndTime>2011-12-13T16:00:00</EndTime>
<ExtendedShiftEndTime>0001-01-01T00:00:00</ExtendedShiftEndTime>
<ExtendedShiftStartTime>0001-01-01T00:00:00</ExtendedShiftStartTime>
<IsExtended>false</IsExtended>
<OtherEmployeesHaveBidOnShift>false</OtherEmployeesHaveBidOnShift>
<ShiftId>00000000-0000-0000-0000-000000000000</ShiftId>
<ShiftProviderId>00000000-0000-0000-0000-000000000000</ShiftProviderId>
<ShiftProviderName>Available</ShiftProviderName>
<StartTime>2011-12-13T08:00:00</StartTime>
<Week>50</Week>
</AvailableShift>
...
</ArrayOfAvailableShift>
```

i) Update Shifts

Use this service to update your shift resources. It is important to send all information about the shift back, even if it is not changed. All changes between the original shift and the new shifts will be updates. So, if an original shift comment is omitted the comment will be deleted.

This service is also used for employees to set their shifts as available by setting `IsAvailable` to true. If an employee uses this service, only the `IsAvailable` property will be used. If a planner changes the employee on a shift, exchanges and bids will be removed if the shift can be changed. (if the shift can be changed it will be removed). If `EmployeeId` is changed to null, `IsAvailable` will be set as true on the server.

This service returns a Response Message. If `Success` is true, the response is successful. If `Success` is false a message will be supplied with a description of the error. To update a shift with a missing end time, set the end time to a date before 01-01-1900.

Request (JSON)

```
PUT /Shifts/{ShiftId}/ HTTP/1.1
x-tamigo-token: <token>

{
  "ShiftId": "00000000-0000-0000-0000-000000000000",
  "StartTime": "\/Date(1355986847000+0100)\/",
  "EndTime": "\/Date(1356008447000+0100)\/",
  "Comment": "Early",
  "BreakCode": "P",
  "IsAvailable": false,
  "IsExchange": false,
  "EmployeeId": "00000000-0000-0000-0000-000000000000",
  "ShiftActivityId": "00000000-0000-0000-0000-000000000000"
}
```

Response (JSON)

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

j) Add Shifts

A Planner can use everything on a shift besides the `IsAvailable` and the `IsExchange` properties. If the `EmployeeId` is sent as `null`, `IsAvailable` will be set as `true` on the server.

This service returns a Response Message. If `Success` is `true`, the response is successful. If `Success` is `false`, a message will be supplied with a description of the error. `type` should be either `planned` or `actual` to define if you are adding a planned or a worked shift.

Request (JSON)

```
POST /Shifts/{type}/ HTTP/1.1
x-tamigo-token: <token>

{
  "StartTime": "\/Date(1355986847000+0100)\/",
  "EndTime": "\/Date(1356008447000+0100)\/",
  "Comment": "Early",
```

```
"BreakCode": "P",
"IsAvailable": false,
"IsExchange": false,
"EmployeeId": "00000000-0000-0000-0000-000000000000",
"ShiftActivityId": "00000000-0000-0000-0000-000000000000"
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

k) Delete Shifts

A Planner or Administrator can delete a shift if they have a role in the specific department.

Request (JSON)

```
DELETE /Shifts/{shiftId}/ HTTP/1.1
x-tamigo-token: <token>
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

l) Import shifts

This service allows you to import shifts into the rota or a timesheet. It is only available when you are logged in as an application.

All dates must be in Microsoft date format, with timestamp in milliseconds. All records must include PlanStatus which can be *actual* for timesheet or *planned* for rota.

The endpoint allows import with *EmployeeId* (GUID), or *WageSystemKey* (string). If both are present, *EmployeeId* is used. It also allows import with *DepartmentId* (GUID) and *DepartmentKey* (string). If both are present, *DepartmentId* will be used.

Planned shifts without employee information can be imported as available. Actual shifts without employee information cannot be imported.

If *ShiftId* is provided, the importer will attempt to update an existing shift with the provided *ShiftId*. If no *ShiftId* is provided, then importer will create a new shift.

ShiftActivityId can be provided (optional), in order to mark shift with a certain activity. *CompetencyId* (Skill) can be added only to planned shifts. Break codes can be imported and must use the Code column of specified *Break*.

Request (JSON)

```
POST /shifts/Import HTTP/1.1
```

```
Content-Type: application/json
```

```
x-tamigo-token: <token>
```

```
[
  {
    "StartTime": "/Date(1624953600000)/",
    "EndTime": "/Date(1624982400000)/",
    "DepartmentId": "2a76d8be-63bb-45ef-9235-0d95c27a0c68",
    "DepartmentKey": " StoreID ",
    "EmployeeId": "4cc81fef-4151-43d5-8bc3-811cba743a62",
    "WageSystemKey": "BBBBBBBB",
    "PlanStatus": "actual",
    "SkillId": "SKI1",
    "ShiftId": "42f4da4d-ad2b-409b-8300-2258491849e9",
    "ShiftActivityId": "c23bb0c2-89ad-4222-ab64-b9f3a746339e",
    "Comment": "Some shift comment",
    "CompetencyId": "c23bb0c2-89ad-4222-ab64-b9f3a746339e",
    "BreakCode": "2"
  }
]
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

Actual Shift Services

a) Actual Shifts – Day Other Than Current Day

This service returns a list of the actual shifts (worked shifts from the timesheet) for the day, including employees that are absent, indicated by the activity. This is available for all roles.

If the department is equal to `all`, shifts from all timesheets in all departments are returned.
If the `endTime` is equal to `1/1/1900`, it means the end time is blank.

Request (JSON)

```
GET /actualshifts/day/{date}/(?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

[{"BreakCode": "Q", "Comment": "",
  "DepartmentName": "Department A", "EmployeeName": "",
  "EndTime": "\\Date(1316700000000+0200)\\",
  "StartTime": "\\Date(1316671200000+0200)\\",
  ...
  {"BreakCode": "Q", "Comment": "",
    "DepartmentName": "Department A", "EmployeeName": "",
    "EndTime": "\\Date(1316700000000+0200)\\",
    "StartTime": "\\Date(1316671200000+0200)\\",
    "ShiftActivityId": "00000000-0000-0000-0000-000000000000",
    "ShiftId": "00000000-0000-0000-0000-000000000000"}]
```

b) Actual Plan Day Status

This service returns a true or false if the day in the timesheet is closed or open.

If `closed = true`. If `open = false`.

Request (XML)

```
GET /actualshifts/status/day/{date}/(?departmentId={departmentId})/ HTTP/1.1
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

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

c) Actual Shifts – By Employee

This service returns a list of the actual shifts (worked shifts from the timesheet) for the past 60 days for a selected employee.

If the department is equal to all, actual shifts from all the employee's departments is returned. If the `Departmentid` is left empty, `defaultdepartment` will be chosen. If the `endTime` is equal to `1/1/1900`, it means the end time is blank

Request (JSON)

```
GET /actualshifts/past/{date}/(?departmentId={departmentId}
&employeeId={employeeId})/ 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
```

```
[{"BreakCode":"Q","Comment":"","
"DepartmentName":"Department A","EmployeeName":"","
"EndTime":"\\/Date(1316700000000+0200)\\/","
"StartTime":"\\/Date(1316671200000+0200)\\/"},
...
{"BreakCode":"Q","Comment":"","
"DepartmentName":"Department A","EmployeeName":"","
"EndTime":"\\/Date(1316700000000+0200)\\/","
"StartTime":"\\/Date(1316671200000+0200)\\/"}]
```

d) Close Actual Day

Use this service to close a day of your timesheet. Supply the day and the department. This is available for planners and administrators.

Request (JSON)

```
PUT /actualshifts/close/day/{date}/(?departmentId={departmentId})/ HTTP/1.1
x-tamigo-token: <token>
}

{
```

```
"Departmentid": "00000000-0000-0000-0000-000000000000",
>Date": "\/Date(1355986847000+0100)\/"
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

e) Open Actual Day

Use this service to reopen a day of your timesheet. Supply the day and the department. This is available for administrators.

Request (JSON)

```
PUT /actualshifts/open/day/{date}/(?departmentId={departmentId})/ HTTP/1.1
x-tamigo-token: <token>
```

```
{
  "Departmentid": "00000000-0000-0000-0000-000000000000",
  "Date": "\/Date(1355986847000+0100)\/"
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

f) Can Planner Reopen Actual Day

This service returns if it is possible for planners to reopen closed days in the timesheet, in the current wage period. The setting *Planners can reopen closed days* under *Configuration -> Company Settings -> Timesheet* must also be enabled in tamigo for planners to reopen a day in the timesheet.

Request (JSON)

```
GET /actualshifts/reopen/day/{date}/ HTTP/1.1
x-tamigo-token: <token>
```

```
{
  "Date": "\/Date(1355986847000+0100)\/"
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

```
{Success":true}
```

g) Copy Approved Rota to Timesheet – For a Day

This service copies an approved rota to the timesheet. This is only available for planners or administrators.

Request (JSON)

```
POST /actualshifts/copy/day/{date}/(?departmentId={departmentId})/ HTTP/1.1
x-tamigo-token: <token>
```

```
{
  "Departmentid": "00000000-0000-0000-0000-000000000000",
  "Date": "\/Date(1355986847000+0100)\/"
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

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

Services for Partner Login

tamigo offers special services for partners. The services are listed below.

Only a partner login has access to these services.

Create New Company

This service creates a new company (with one administrator). Fill out the different parameters to set culture and package type for the company, and name, email, etc. for the administrator.

Request (JSON)

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

Body:

Company

```
{
  "Name": "Company Name",
  "ContactPerson": "Administrator Name",
  "Address": "Street and Number",
  "City": "City Name",
  "Email": "administratorname@company.com",
  "Password": "login password",
  "Phone": "Administrator phone number",
  "zipCode": "post code",
  "Culture": "language of the company",
  "PackageType": "package type is ranged from 1-5",
}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

```
{Message:"Company Created!" Success:true}
```

Touch Services for Smartphones

Touch services for smartphones are used to check in and out.

a) TouchCheckIn

The TouchCheckIn method allows the user to check in from a smartphone.

Request (JSON)

```
POST /Attendance/TouchCheckIn/ HTTP/1.1
x-tamigo-token: <token>
```

Body:

Null

Response (JSON)

```
{
  "CheckInTime": "12:37",
  "CheckOutTime": "null",
  "CurrentTime": "12:48:32",
  "CurrentStatus": "AllowCheckOut",
  "EnableShiftBreakCodePanel": "true",
  "IsCheckedInElsewhere": "false",
}
```

b) TouchCheckOut

Similar to the `TouchCheckIn` method, the `TouchCheckOut` method calls the `CheckOut` service and returns an `AttendanceInformation` object, updated with the `CheckOutTime`. If the break time is empty, insert `null`.

Request (JSON)

```
POST /Attendance/TouchCheckOut/{breakcodeid}/ HTTP/1.1
x-tamigo-token: <token>
```

Body:

Null

Response (JSON)

```
{
  "CheckInTime": "12:37",
  "CheckOutTime": "15:45",
  "CurrentTime": "15:45:00",
  "CurrentStatus": "AllowCheckIn",
  "EnableShiftBreakCodePanel": "false",
  "IsCheckedInElsewhere": "false",
}
```

c) GetTouchStatus

This method returns an `AttendanceInformation` type of object, updated with the latest information.

Request (JSON)

```
GET /Attendance/GetTouchStatus/ HTTP/1.1
x-tamigo-token: <token>
```

Body:

```
Null
```

Response (JSON)

```
{
  "CheckInTime": "12:37",
  "CheckOutTime": "null",
  "CurrentTime": "12:48:32",
  "CurrentStatus": "AllowCheckOut",
  "EnableShiftBreakCodePanel": "true",
  "IsCheckedInElsewhere": "false",
}
```

d) GetBreakCodes

This method returns a list of `AttendanceBreakCode` types of objects, with the break codes corresponding to a certain employee. `Breakcodeid` is optional. However, if sent, the service will return a list with the `breakcodeid`'s.

Request (JSON)

```
GET /Attendance/GetBreakCodes/?breakcodeid={breakcodeid}/ HTTP/1.1
x-tamigo-token: <token>
```

Body:

```
Null
```

Response (JSON)

```
[{"BreakCodeID": "1",
  "Code": "12",
  "Enabled": "false",
  "Hours": "0.12",
  "Name": "Q",
  "Selected": "true"},
```

```
...
[{"BreakCodeID": "4",
"Code": "30",
"Enabled": "false",
"Hours": "1.5",
"Name": "R",
"Selected": "true"}]
```

e) UpdateBreakCode

This method is used in the application when a user is selecting a certain break code. The purpose is to update the property of the selected break code. The method returns a `ResponseMessage`, which has the `Success = true` in the case that the break code was successfully updated, and `Success = false` otherwise.

Request (JSON)

```
POST /AttendanceService/UpdateBreakCode/{breakcodeid}/ HTTP/1.1
x-tamigo-token: <token>
```

Body:

```
Null
```

Response (JSON)

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

f) GetTouchAccess

The `GetTouchAccess` method is used to decide whether a device has access to Touch or not.

In the case that the company does not have the option `AlwaysCheckIPOnTouchLogin` enabled, the device using Touch will always be able to use it. Otherwise, a set of checks on the IP address is made – this is used when components are first initialised in the smartphone versions.

Request (JSON)

```
GET /AttendanceService/GetTouchAccess/ HTTP/1.1
x-tamigo-token: <token>
```

Body:

Null

Response (JSON)

```
{Message:"" Success:"false"}
```

Wage Percentage Services

The service returns a wage percentage report for a selected period and for a desired interval. This is available only when you are logged in as an application. Parameter *departmentId* is an optional parameter, if not specified then all departments will be included in the report. Parameter *interval* is an integer representing a certain interval of the report. The meaning of the values are shown in the table below.

Interval	Value
Day	1
Week	2
Month	3

Request (JSON)

```
GET /wagepercentage/report/?startdate={startDate}&enddate={endDate}&interval={interval}&departmentId={departmentId}/ HTTP/1.1  
Content-Type: application/json  
x-tamigo-token: <token>
```

Response (XML)

```
<ArrayOfWagePercentageRowModel  
xmlns="http://schemas.datacontract.org/2004/07/Tamigo.Models.Wage"  
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
  <WagePercentageRowModel>  
    <ActualCommission i:nil="true"/>  
    <ActualHourlyBudget>0</ActualHourlyBudget>  
    <ActualRevenue>0.0</ActualRevenue>
```

```
<ActualWageCost i:nil="true"/>
<ActualWagePercentage>0</ActualWagePercentage>
<BudgetRevenue>0.0</BudgetRevenue>
<BudgetWageCost i:nil="true"/>
<BudgetWagePercentage>0</BudgetWagePercentage>
<Date>0001-01-01T00:00:00</Date>
<DepartmentName i:nil="true"/>
<ExpectedCommission i:nil="true"/>
<ExpectedRevenue>0.0</ExpectedRevenue>
<PlanWageCost i:nil="true"/>
<PlanWagePercentage>0</PlanWagePercentage>
<PlannedHourlyBudget>0</PlannedHourlyBudget>
<SubRows/>
  <WageIndex i:nil="true"/>
</WagePercentageRowModel>
</ArrayOfWagePercentageRowModel>
```

Document Services

The document services allow you to manage employee's documents.

a) Upload a new document

Use this service to upload a document to the employee's documents folder.

List of parameters:

- **employeeId**: a unique identifier of the employee
- **fileName**: a name of the file to upload
- **documentType** (optional): document type, if not specified it will default to 'Proof of Identity'

Request (JSON)

```
POST /Documents/?employeeId=00000000-0000-0000-0000-
000000000000&fileName=document.pdf&documentType=Proof of Identity HTTP/1.1
x-tamigo-token: <token>
Content-Type: <file content type>
```

```
"<file contents here>"
```

Response (JSON)

```
HTTP/1.1 200 OK
Content-Length: 0
```

Financials Services

The financials services allow you to interact with the Financials functionality.

a) Get Available Hours (Hourly budget report)

This service returns a report of available hours for the selected period and department. This is available only when you are logged in as an application. If the department is left empty, shifts from all departments are returned. The timespan limitation of this endpoint is 60 days.

Request (JSON)

```
GET /Financials/HourlyBudgetReport?periodType=monthly&from=2021-09-01&to=2021-09-30&
Content-Type: application/json
x-tamigo-token: <token>
```

Response (JSON)

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

```
[
  {
    "ActualHours": 0,
    "BudgetHours": 42.8571428550,
    "Date": "/Date(1630447200000+0200)/",
    "DepartmentName": "Department 1",
    "OrganizationalUnitId": "15ec58f8-864c-4e7f-ace4-002c1d674b41"
  },
  {
    "ActualHours": 0,
    "BudgetHours": 85.7142857130,
    "Date": "/Date(1630447200000+0200)/",
    "DepartmentName": "Department 2",
    "OrganizationalUnitId": "58c3f06b-2cec-41cf-9cc1-01dd682433eb"
  },
  {
    "ActualHours": 0,
    "BudgetHours": 100.7142857130,
    "Date": "/Date(1630447200000+0200)/",
    "DepartmentName": "Department 3",
    "OrganizationalUnitId": "bebaddb1-0855-4ed4-9a26-07bc4d7aa9"
  }
]
```

Previous article

[PowerBI integration](#)

Next article

[The New Rota](#)

Related articles

[Application Services - Web Services \(API\)](#)

[The EU Working Time Directive: Requirements for time registration](#)

[Absences](#)

[Setting Up Breaks](#)

[Sprint 30 - June 12, 2024](#)

Recently viewed articles

[Application Services - Web Services \(API\)](#)

Articles in this section

- [Application Services - Web Services \(API\)](#)
- [How to create API keys and use claims](#)
- [Using the CSV Importer \(HR\) Plugin](#)
- [Setting Up Workload Planner](#)
- [PowerBI integration](#)
- **[User Services - Web Services \(API\)](#)**

Can't find what you're looking for?

Our team of experts is here to help

Contact Support
