



Tamigo Web Services 1.0

Contents

Summary.....	3
Application Services.....	4
Login Service.....	4
Application Login.....	4
Attendance.....	5
Check-in.....	5
Check-out.....	5
Employee Service.....	7
Retrieve Employees.....	7
User Services.....	8
Login Service.....	8
User Login.....	8
Shift service.....	9
My overview.....	9
Authorized roster - today.....	10
Authorized roster – arbitrary day.....	10
Vacant shifts.....	11
Update shift.....	12
Bid service.....	13
Get available shifts.....	13
Lay bid for available shift.....	14
Accept bid on available shift.....	14
Shift Exchange Service.....	15
Create new Shift Exchange.....	15
List of shifts to exchange with.....	15
Get List of pending approvals (Employee and Planner).....	16
Decide shift exchange.....	17
News Service.....	18
Get company and department news.....	18
Absence Service.....	19
Create new absence request / Register absence.....	19
List of future absence.....	19

Decide leave requests	20
List of leave types	20
List of employees from department.....	20
Requests Service.....	21
Pending bids for vacant shifts	21
Decide bid for shift	21
Contact Service	23
Contact List.....	23
Departments Service	24
All Departments.....	24
Set Default department.....	24

Summary

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 other set of services uses a Tamigo user as authentication, and will return data based on the users context. These services can be used to create other user experiences for Tamigo, and we use it ourselves to power the mobile applications for Tamigo.

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

Application Services

Login Service

Application Login

To login as an application first log in to Tamigo as an administrator. Go to Configuration -> Manage Application Keys to create a new application key.

The application key is used for authentication of your application and it is therefore paramount that you 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
Content-Type: application/json
```

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

Response (JSON)

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

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

Attendance

Tamigo offers the attendance service, which allows you to register check-in and check-out for employees directly in Tamigo from a third-party system (e.g. Point-Of-Sale system).

The Attendance Service requires you to set up keys in Tamigo for the departments and employees whom should be able to use the service.

To set up these keys, login to Tamigo as a Planner or Administrator. Go to Configuration -> Departments, to set the Department ID (POS Key ID). Go to Configuration -> POS Keys, to set the corresponding keys for the Employees.

Check-in

Use check-in to register an employee has arrived at work. Tamigo will register the check-in at the time you send, and match it to the employee and department. As time rounding rules can be applied, the service returns the rounded time.

Parameter	Format	Example
departmentKey	Tekst	A123, 1234567, abcdefg
employeeKey	Tekst	A123, 1234567, abcdefg
checkInTime	yyyy-MM-	2011-09-02T12:37:00
sessionToken	as returned by	766c9732-e2d1-46d1-ae3e-

Request (JSON)

POST /attendance/checkin/?token=<sessionToken> HTTP/1.1
Content-Type: application/json

```
{
  "DepartmentKey": "<departmentKey>",
  "EmployeeKey": "<employeeKey>",
  "Time": "<checkInTime>"
}
```

Response (JSON)

HTTP/1.1 200 OK

Check-out

Use check-out if you want to check out from work. Usage as check-in.

Parameter	Format	Eksempel
departmentKey	Tekst	A123, 1234567, abcdefg
employeeKey	Tekst	A123, 1234567, abcdefg
checkInTime	yyyy-MM-	2011-09-02T12:37:00
sessionToken	Som modtaget ved	766c9732-e2d1-46d1-ae3e-

Request (JSON)

POST /attendance/checkout/?securityToken=<sessionToken> HTTP/1.1

Content-Type: application/json

```
{"DepartmentKey": "<departmentKey>",  
  "EmployeeKey": "<employeeKey>",  
  "Time": "<checkInTime>"}
```

Response (JSON)

HTTP/1.1 200 OK

Employee Service

The employee service allows you to retrieve and update employee information for use in a third party system (e.g. Salary System).

Retrieve Employees

If you want a list of employees this is where you go. All employees in the company will be retrieved.

Request (JSON)

```
GET /employees/?securityToken=<securityToken>  
Content-Type: application/json
```

Response (JSON)

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

```
[{"Email":"user1@tamigo.com","EmployeeId":"9c0f571e-cd0e-4e38-ac51-  
ecf965cd3de6","IsPlanner":false,"Name":"Søren","PosKey":null},  
...  
{"Email":"tp1125@tamigo.com","EmployeeId":"d1d0b7db-c774-4e81-b094-  
d1d2acaa1d83","IsPlanner":false,"Name":"Janik","PosKey":null}]
```

User Services

Login Service

User Login

To use user services you need to login as a user of Tamigo, providing their email and password for authentication.

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

```
{"Email": "karina@tamigo.com",  
  "Password": "password"}
```

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
```

```
{"DefaultDepartmentName": "Ny Bistro",  
  "Email": "test@example.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>
```

Shift service

The shift service can be used to retrieve information on rosters and upcoming shifts.

My overview

Returns a list of the logged in employees upcoming shifts. Dates are expressed as milliseconds since EPOCH (1970-1-1 00:00:00). Available for All.

Request (JSON)

```
GET /shifts/future/?securitytoken={sessionToken}
Content-Type: application/json
```

Response (JSON)

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

```
[{"BreakCode": "Q",
  "Comment": "",
  "DepartmentName": "Ny Bistro",
  "EmployeeName": null,
  "EndTime": "\/Date(1318860000000+0200)\/",
  "StartTime": "\/Date(1318831200000+0200)\/"},
  ...
{"BreakCode": "Q",
  "Comment": "",
  "DepartmentName": "Ny Bistro",
  "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>Herning</DepartmentName>
<EmployeeName i:nil="true"/>
<EndTime>2011-12-19T06:00:00</EndTime>
<IsAvailable>>false</IsAvailable>
<IsExchange>>false</IsExchange>
<ShiftId>2f5e8bcc-a419-4c2b-891a-54fa4592bd52</ShiftId>
<StartTime>2011-12-18T23:00:00</StartTime>
</Shift>
</ArrayOfShift>
```

Authorized roster - today

This resource returns a list of the authorized roster for the day including employees that are on leave, signified by an associated activity. Available for all Roles.

Request (JSON)

```
GET /shifts/today/?securitytoken={sessionToken}&departmentId={departmentId}
Content-Type: application/json
```

Response (JSON)

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

```
[{"BreakCode": "Q", "Comment": "",
  "DepartmentName": "Ny Bistro", "EmployeeName": "",
  "EndTime": "\/Date(1316700000000+0200)\/",
  "StartTime": "\/Date(1316671200000+0200)\/"},
  ...
 {"BreakCode": "Q", "Comment": "",
  "DepartmentName": "Ny Bistro", "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>København</DepartmentName>
<EmployeeName/>
<EndTime>2011-12-13T16:00:00</EndTime>
<StartTime>2011-12-13T08:00:00</StartTime>
</Shift>
</ArrayOfShift>"
```

Authorized roster – arbitrary day

This resource returns a list of the authorized roster for the day including employees that are on leave, signified by an associated activity. Available for all Roles.

Request (JSON)

```
GET /shifts/day/{date}?securitytoken={sessionToken}&departmentId={departmentId}
Content-Type: application/json
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Content-Type: application/json; charset=utf-8

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

Vacant shifts

Deprecated, use Bid service instead. This resource is only available for backwards compatibility. Returns list of shifts that are open to take. Available for all Roles.

Request (XML)

GET

/Shifts/available/?securitytoken={sessionToken}(&departmentId={departmentId})

Content-Type: application/json

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>dc1a04be-fb08-443f-aa3f-06b0a7bffb28</DepartmentId>
    <DepartmentName>København</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>d54f1963-bdd4-4929-a053-4efb947f96b2</ShiftId>
    <ShiftProviderId>00000000-0000-0000-0000-000000000000</ShiftProviderId>
    <ShiftProviderName>Ledig</ShiftProviderName>
    <StartTime>2011-12-13T08:00:00</StartTime>
    <Week>50</Week>
  </AvailableShift>
  ...
</ArrayOfAvailableShift>
```

Update shift

Use this to update your shift resources.

Request (JSON)

```
PUT /Shifts/{ShiftId}/?securitytoken={sessionToken}
```

```
{"ShiftId": "<ShiftId>", "PlanId": "<guid>",  
"EmployeeId": "<guid>", "StartDateTime": "\/Date(1316671200000+0200)\/",  
"EndDateTime": "\/Date(1316700000000+0200)\/",  
"StartTime": "\/Date(1316671200000+0200)\/", "ShiftHours": "0", "Comments": "",  
"IsAvailable": "true", "IsExchange": "false", "ShiftActivityId": "<guid>"}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Bid service

The Bid service gives the user a view on shifts that have been offered, and it enables bidding for these shifts.

Get available shifts

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

Request (JSON)

```
GET /bids/Available/?securitytoken={sessionToken}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

```
[{"BreakCode": "Q", "Comment": "",
  "DepartmentName": "Ny Bistro", "EmployeeName": "",
  "EndTime": "\/Date(1316700000000+0200)\/",
  "StartTime": "\/Date(1316671200000+0200)\/"},
  ...
  {"BreakCode": "Q", "Comment": "",
  "DepartmentName": "Ny Bistro", "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>dc1a04be-fb08-443f-aa3f-06b0a7bffb28</DepartmentId>
<DepartmentName>København</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>Ledig</ShiftProviderName>
<StartTime>2011-12-21T08:00:00</StartTime>
<Week>51</Week>
</AvailableShift>
</ArrayOfAvailableShift>"
```

Lay bid for available shift

The logged in employee bids on an available shift

This is only available to Employees. Planners cannot bid on shifts

Request (JSON)

```
PUT /Shifts/Available/{shiftId}/?securitytoken={sessionToken}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Accept bid on available shift

Use this resource to decide a bid for an available shift. Available only to Planners.

Use the "IsAccepted" property to accept or reject the request

Request (JSON)

```
PUT /Requests/{requestId}/?securityToken={sessionToken}
```

```
{
  "RequestId": "dbdb767f-1415-4cdb-8763-1a7c2281f1b8",
  "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>dbdb767f-1415-4cdb-8763-1a7c2281f1b8</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
```


Get List of pending approvals (Employee and Planner)

Get a list of pending Exchange approvals.

An employee will get a list of exchanges that needs to be approved – this means that the employee accepts that he/she would like to exchange the selected shifts.

A Planner will get a list of exchanges that have been approved by the employee and needs to Accept the exchange before it is final.

The Property Violate11HourRule is only available when the planner is accepting the request. If the property is True. There should be an alert.

Request (JSON)

```
GET /Exchanges/Requests/?securityToken={sessionToken}
```

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>38f02810-6ac4-40b4-b238-39236c826118</EmployeeId1>
<EmployeeId2>3203ba4b-5264-4f36-8daa-76e7e69d0f31</EmployeeId2>
<EmployeeName1>Peter</EmployeeName1>
<EmployeeName2>Karina</EmployeeName2>
<EndDateTime1>2011-12-24T16:00:00</EndDateTime1>
<EndDateTime2>2011-12-23T16:00:00</EndDateTime2>
<IsAccepted i:nil="true"/>
<RequestId>9aeb6e1a-d50a-40f0-bd79-0369fcc25ddc</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>
</ShiftExchangeRequest>
</ArrayOfShiftExchangeRequest>
```

Decide shift exchange

Approve or reject a shift exchange. The approval is a two-step process:

1. Approval by the employee who are exchanging his shift (the one that did not create the shift exchange)
2. Approval by the planner.

Available to All Roles.

Request (JSON)

PUT /Exchanges/Requests/{requestId}/ {isAccepted}/?securitytoken={sessionToken}

Response (JSON)

HTTP/1.1 200 OK

News Service

The News Service offer access to the information that is posted on the front page of Tamigo.

Get company and department news

This resource returns the content from the front page in Tamigo. The content is returned as HTML. This resource is available to all Roles.

Request (JSON)

```
GET /News/?securityToken={sessionToken})
```

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>
```

Absence Service

The absence service enables the Employee to ask for absence, and the Planner can register absence (e.g. an employee no-show due to illness), the Planner can also approve or reject the Employee's absence requests.

Create new absence request / Register absence

When an employee wishes to plan absence he submits a absence request. When a Planner wishes to register absence he uses the same service.

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

Request (JSON)

POST /LeaveRequests/?securityToken={sessionToken}

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

Response (JSON)

HTTP/1.1 200 OK

List of future absence

This resource returns a list of the registered absence for the logged in employee.

Request (JSON)

GET /Leave/Future/?securityToken={sessionToken}

Response (JSON)

HTTP/1.1 200 OK

```
[ { "LeaveRequestId": "766c9732-e2d1-46d1-ae3e-a74c560bb8e6",
  "EmployeeId": "ffff9732-e2d1-46d1-ae3e-a74c560bb702",
  "EmployeeName": "Karina",
  "DateFrom": "\/Date(1316700000000+0200)\/",
  "DateTo": "\/Date(1316700000000+0200)\/",
  "LeaveTypeId": "1",
  "LeaveTypeName": "Vacation" }
,
...
{ "LeaveRequestId": "766c9732-e2d1-46d1-ae3e-a74c560bb8e6",
  "EmployeeId": "ffff9732-e2d1-46d1-ae3e-a74c560bb702",
  "EmployeeName": "Karina",
```

```
"DateFrom": "\Date(1316700000000+0200)\/",
"DateTo": "\Date(1316700000000+0200)\/",
"LeaveTypeId": "1"
"LeaveTypeName": "Vacation"}]
```

Decide leave requests

When a planner wishes to approve or deny a leave request he can update the corresponding leave request.
Available to Planner only.

Request (JSON)

```
PUT /LeaveRequests/{requestId}/?securityToken={token}
```

Request (JSON)

```
{ "LeaveRequestId": "<requestId>",
  "IsApproved": "<bool>",
  "MoveShiftsToVacant": "<bool>",
  "SmsComment": "<comment>"}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

List of leave types

This resource returns a list of the currently available leave types for use by the leaverequest service.
Available to all roles, employees can only choose between a subset of leavetypes, as specified in Tamigo.

Request (JSON)

```
GET /Leave/LeaveRequestTypes/?securityToken={token}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

```
[{"LeaveTypeId": "766c9732-e2d1-46d1-ae3e-a74c560bb8e6", "Name": "Ferie"},
...
{"LeaveTypeId": "aaaaa732-e2d1-46d1-ae3e-a74c560bb8e6", "Name": "Syg"}]
```

List of employees from department

Returns the list of employees you can register absence on.

Request (JSON)

```
GET /Leave/Employees/?securityToken={token}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

Requests Service

The request service offers the Planner an option to decide on bids from his Employees.

Pending bids for vacant shifts

This resource returns a list of pending bids. Available for Planners

Request (JSON)

GET /Requests/?securitytoken={sessionToken}

Response (JSON)

```
[{
  "Comment": "",
  "EndDateTime": "\/Date(1324220400000+0100)\/",
  "IsAccepted": null,
  "RequestId": "6a957dc0-bdc8-47bf-9a0e-5f4d936f85da",
  "RequesterId": "38f02810-6ac4-40b4-b238-39236c826118",
  "RequesterName": "Peter",
  "ShiftHours": 7.25,
  "StartDateTime": "\/Date(1324191600000+0100)\/",
  "ProviderId": "00000000-0000-0000-0000-000000000000",
  "ProviderName": "Ledig",
  "Violate11HourRule": "false"
},
...
{
  "Comment": "",
  "EndDateTime": "\/Date(1324220400000+0100)\/",
  "IsAccepted": null,
  "RequestId": "f2a77510-c92b-4543-984b-c7323fc3311b",
  "RequesterId": "38f02810-6ac4-40b4-b238-39236c826118",
  "RequesterName": "Peter",
  "ShiftHours": 7.25,
  "StartDateTime": "\/Date(1324191600000+0100)\/",
  "ProviderId": "00000000-0000-0000-0000-000000000000",
  "ProviderName": "Ledig",
  "Violate11HourRule": "false"
}]
```

Decide bid for shift

Approve or reject a bid or shift exchange. Available to Planner, Administrator

Request (JSON)

PUT /Requests/{requestId}/?securitytoken={sessionToken}

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

Response (JSON)

HTTP/1.1 200 OK

Contact Service

The contact service returns a list of contact information for employees and planners in the current department.

Contact List

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

Request (JSON)

```
GET /Contacts/?securitytoken={sessionToken}
```

Response (JSON)

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
[{"Email":"karina@tamigo.com","FirstName":"Andreas  
T","ImageUrl":"\/images\/small\/Andreas  
T_Untitled.png","LastName":null,"MobilePhone":"+4511111111"},  
...  
{"Email":"peter@tamigo.com","FirstName":"Ulrik","ImageUrl":"","LastName":null,"  
MobilePhone":"+4511111111"}]
```

Departments Service

The department service is used for operations to retrieve a list of departments for the company, and updating of the default department for a User.

All Departments

Get a list of the currently logged in user's available departments. The list is filtered to only show departments where the user has a role.

Request (JSON)

```
GET /Departments/?securitytoken={sessionToken}
Content-Type: application/json
```

Response (JSON)

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

```
[{"DepartmentId": "223a4516-41d5-4327-97ca-65c67d6b20a0",
  "IsDefault": false, "Name": "Afdeling 1"},
  ...
 {"DepartmentId": "c4dcc6f2-ff52-4a22-b29d-aae817b5c8e7",
  "IsDefault": false, "Name": "Slik"}]
```

Set Default department

In v1 of our iPhone application you change the department that your currently looking at by setting the default department.

Request (JSON)

```
PUT /departments/c4dcc6f2-ff52-4a22-b29d-
aae817b5c8e7/?SecurityToken={sessionToken}
Content-Type: application/json
```

```
{"DepartmentId": "c4dcc6f2-ff52-4a22-b29d-aae817b5c8e7", "IsDefault": true}
```

Response (JSON)

```
HTTP/1.1 200 OK
```