

OroTimesheet.com

API

(Application Programming Interface)

Copyright © 1996-2017
OroLogic Inc.
<http://www.oroLogic.com>

Revision 8.33.0 (July 7, 2017)

Contents

- OroTimesheet API..... 1**
- Introduction to OroTimesheet API..... 1**
- Target Audience 1**
- Internationalization and Character Sets 1**
- Prerequisites..... 1**
- API password..... 1**
- API requests..... 1**
 - Note about strings 2
 - Note about date formats 2
 - Note about number format 3
 - Note about user defined fields 3
- API URL..... 3**
- How to send a request and get the response..... 3**
 - PHP example..... 3
 - Visual Basic example..... 4
- List of API request types..... 4**
 - employee_list (Get entire list of employees in a summary format) 4
 - employee_get (Get an employee record) 5
 - employee_edit (Edit an existing employee) 6
 - employee_group_list (Get entire list of employee groups in a summary format) 8
 - employee_group_get (Get an employee group record) 8
 - employee_group_edit (Edit an existing employee group) 9
 - activity_type_list (Get entire list of activity types in a summary format) 10
 - activity_type_get (Get an activity type record) 11
 - activity_type_add (Add a new activity type) 11
 - activity_type_edit (Edit an existing activity type) 12
 - activity_type_delete (Delete an activity type) 12
 - customer_list (Get entire list of customers in a summary format) 13
 - customer_get (Get a customer record) 13
 - customer_add (Add a new customer) 15
 - customer_edit (Edit an existing customer) 16
 - customer_delete (Delete a customer) 16
 - customer_special_rate_add (Add a special rate to an existing customer) 17
 - customer_special_rate_delete (Delete a special rate from an existing customer) 17
 - project_list (Get entire list of projects and sub-projects in a summary format) 18
 - project_get (Get a project record) 19
 - project_add (Add a new project) 21
 - project_edit (Edit an existing project) 22
 - project_delete (Delete a project) 23
 - project_member_add (Add a member (employee or employee group) to a project) 23
 - project_member_delete (Delete a member (employee or employee group) from a project) 24
 - project_additional_manager_add (Add an additional manager (employee or employee group) to a project) 24
 - project_additional_manager_delete (Delete a manager (employee or employee group) from a project) 25
 - project_activity_type_add (Add an activity type or activity type category to a project) 26
 - project_activity_type_delete (Delete an activity type or activity type category from a project) 26

project_special_rate_add (Add a special rate to an existing project).....	27
project_special_rate_delete (Delete a special rate from an existing project).....	28
sub_project_add (Add a sub-project to a project).....	28
sub_project_edit (Edit an existing sub-project).....	29
sub_project_delete (Delete a sub-project).....	29
sub_project_activity_type_hide (Hide an activity type or activity type category for a sub-project).....	30
sub_project_activity_type_unhide (Unhide an activity type or activity type category from a sub-project).....	31
sub_project_budget_distribution_edit (Set budget distribution for a sub-project and an activity type, an employee or an expense type).....	31
timesheet_tx_list (Get list of timesheet transactions in a summary format).....	32
timesheet_tx_get (Get a timesheet transaction record).....	33
timesheet_tx_add (Add a timesheet transaction).....	35
timesheet_tx_edit (Edit an existing timesheet transaction).....	36
timesheet_tx_delete (Delete a timesheet transaction).....	36
expense_get (Get an expense transaction record).....	37
expense_add (Add an expense to an existing timesheet transaction).....	38
expense_edit (Edit an existing expense transaction).....	38
expense_delete (Delete an expense transaction).....	39
timesheet_week_state_get (Get the state of the timesheet of an employee/week).....	39
timesheet_week_state_set (Set the state of the timesheet of an employee/week).....	40
time_bank_update_list (Get entire list of time banks updates in a summary format).....	41
time_bank_update_get (Get detailed information about a time bank update).....	42
invoice_list (Get entire list of invoices in a summary format).....	43
invoice_get (Get detailed information about an invoice).....	44
employee_category_list (Get entire list of employee categories).....	46
activity_type_category_list (Get entire list of activity_type categories).....	46
customer_category_list (Get entire list of customer categories).....	47
project_category_list (Get entire list of project categories).....	47
expense_type_list (Get entire list of expense types).....	47
fiscal_period_list (Get entire list of fiscal periods).....	48
tax_rate_list (Get entire list of tax rates).....	49
Appendix A. List of error codes.....	50
Appendix B. Shift and invoice status descriptions.....	51
Shift descriptions.....	51
Invoice status descriptions.....	51
Copyright.....	52

OroTimesheet API

Introduction to OroTimesheet API

The OroTimesheet API (Application Programming Interface) is an interface that allow you to exchange data between OroTimesheet and other software. This API allows you to read and write data in most of options of OroTimesheet such as Projects, Employees, Timesheets, etc. This API is based on industry standard components HTTPS (Secure Hypertext Transfer Protocol) and XML (Extensible Markup Language).

Target Audience

This document is intended for developers of applications that will connect to the OroTimesheet.com web site.

Internationalization and Character Sets

OroTimesheet uses UTF-8 encoding to store and display characters in the application.

Prerequisites

The API works only with OroTimesheet.com. To activate the API, from OroTimesheet, select the tab **Plus** then select the menu **Configuration | Application configuration**. Check the box **Allow access to data via API** from the section named **API (Application Programming Interface)**. Note that the API can be fully used even if you are in trial mode.

API password

Each OroTimesheet account has a unique API password which can be found in the option **Application configuration** (section **API (Application Programming Interface)**) via the tab **Plus**. A new API password can be generated just by clicking the button **Change API password**. Note that if you generate a new API password for your account, the previous password will not work anymore to perform API requests.

This API password is used for two things:

- 1) To prevent unauthorized access to your OroTimesheet data.
- 2) To know which OroTimesheet account data you want to access. Indeed, each OroTimesheet password is unique. Two different accounts will never have the same password.

API requests

All API requests must be made via SSL over HTTPS. Requests must be submitted via HTTPS POST in the following format:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
```

```
Request_details
</request>
```

Replace abcdef-123456 by your API password then replace Request_details by valid request instructions. See section **API request types** below for more information about valid requests instructions.

Here is an example API request that will add a new sub-project in a project record:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_add>
    <project_id>5271</project_id>
    <description>Phase 3</description>
    <position>3</position>
    <time_bank>accumulated_time</time_bank>
  </sub_project_add>
</request>
```

If the example API request above is successful, the API will return an XML as the following:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>3834</sub_project_id>
</response>
```

On the other side, if the example API request above fails, the API will return an XML as the following:

```
<response app="orotimesheet" version="8" status="0">
  <error code="1102">
    <![CDATA[Record duplicate error. A record with the supplied
      value already exists. There is already a sub-project
      with this description for this project.]]>
  </error>
</response>
```

In short, you have to verify the **status** attribute of the **response** tag to know if the API request was successful or not. A value of **1** means successful while a value of **0** means that an error occurred. Additional information will be also included in the XML result according to the kind of result and/or the kind of request you made. Refer to the section **API request types** below for more information about results each request type can produce.

Note about strings

Note that all strings contained in responses are enclosed in a CDATA section. When you send a request, you can as well enclose strings in CDATA sections for example when these strings contain special characters. However, if your strings does not include special characters, it is not mandatory to use CDATA section in your XML requests.

Note about date formats

Date are always returned in the following format: YYYY-MM-DD
Ex.: 2013-04-25

DateTime are always returned in the following format YYYY-MM-DD HH:MM:SS
Ex.: 2013-04-25 23:15:00

Time are always returned in the following format: HH:MM:SS

When you have to pass Date, Time or DateTime to an API request, you must also respect these formats as well else your request could be rejected or return an error message.

Note about number format

Decimal numbers are always returned with a dot as the decimal separator and nothing as the thousands separator. Ex.: 23941.60

When you have to pass decimal numbers to an API request, numbers must be passed in this format as well else your request could be rejected or return an error message.

Note about user defined fields

If you defined user defined fields, when applicable, these fields will be returned in the response as well. In such case, the tag for these fields will be named udfXXXX where XXXX is the user defined field number generated by the system at user defined field creation time. An attribute named **description** will be added to the tag as well to let you know the description of the user defined field.

You can insert and update data in user defined fields as well. In such case, just use the tag name that correspond to the user defined field to update. You do not have to specify the **description** attribute when adding or updating data in user defined fields. See examples in sections **customer_get** and **customer_add** below for more information about user defined fields.

API URL

Always use the following URL to send your XML requests: <https://app.orotimesheet.com/api.php>

This url requires only one parameter (POST only) named **xml_request**.

How to send a request and get the response

You can use any programming language you want to send the request to the API.

PHP example

Here is a PHP example that is using CURL to send a request and retrieve the response. This example generates an HTML page and prints the result in a textarea object:

```
<?php
$xml = '<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_add>
    <project_id>5271</project_id>
    <description>Phase 3</description>
    <position>3</position>
    <time_bank>accumulated_time</time_bank>
```

```

        </sub_project_add>
    </request>';

$postfields = array('xml_request'=>$xml);
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://app.orotimesheet.com/api.php');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
$result = curl_exec($ch);

?>

<!DOCTYPE html>
<html>
<head>
    <title>test</title>
</head>
<body>
    <textarea rows="200" cols="200"><?php echo $result; ?></textarea>
</body>
</html>

```

Visual Basic example

Here is a Visual Basic example that retrieves the customer list. The variable API_PW should be set to your API password. This example prints the result in the debugger window:

```

Sub TestHTTPPost()
    Set objHTTP = CreateObject("MSXML2.ServerXMLHTTP")
    URL = "https://app.orotimesheet.com/api.php"
    objHTTP.Open "POST", URL, False
    objHTTP.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
    objHTTP.send ("xml_request=<request app=""orotimesheet"" version=""5""
password="" & API_PW & ""><customer_list></customer_list></request>")
    Debug.Print objHTTP.responseText
End Sub

```

List of API request types

employee_list (Get entire list of employees in a summary format)

Once the list of employees retrieved, you can use the **employee_get** request type to retrieve the complete record of a specific employee using the **employee_id** retrieved.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
```

```
<employee_list>
</employee_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
<employee>
  <employee_id>1</employee_id>
  <name><![CDATA[John Doe]]></name>
  <email><![CDATA[jdoe@orotimesheet.com]]></email>
  <status>active</status>
</employee>
<employee>
  <employee_id>14</employee_id>
  <name><![CDATA[Janet White]]></name>
  <email><![CDATA[janet@orotimesheet.com]]></email>
  <status>active</status>
</employee>
</response>
```

employee_get (Get an employee record)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <employee_get>
    <employee_id>14</employee_id>
  </employee_get>
</request>
```

employee_id: integer – Required - Id of the employee to retrieve. If the employee id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <employee>
    <employee_id>14</employee_id>
    <name><![CDATA[Janet White]]></name>
    <email><![CDATA[janet@orotimesheet.com]]></email>
    <status>active</status>
    <type>0</type>

    <employee_group_description><![CDATA[Everybody]]></employee_group_description>
    <employee_group_id>0</employee_group_id>
    <category_description><![CDATA[Default category]]></category_description>
    <category_id>0</category_id>
    <notes><![CDATA[This is a good employee]]></notes>
    <sin><![CDATA[125-147-874]]></sin>
    <hire_date>2005-08-25</hire_date>
    <max_minutes_enterable>480</max_minutes_enterable>
```



```

<max_minutes_enterable_type>1</max_minutes_enterable_type>

<max_minutes_enterable_use_group_settings>no</max_minutes_enterable_use_group_s
ettings>
  <minutes_usually_worked_per_week>2400</minutes_usually_worked_per_week>

<minutes_usually_worked_per_week_use_group_settings>yes</minutes_usually_worked
_per_week_use_group_settings>
  <billing_rate>50.0000</billing_rate>
  <billing_rate_type>1</billing_rate_type>
  <billing_rate_billable>yes</billing_rate_billable>
  <billing_use_group_settings>no</billing_use_group_settings>
  <cost_rate>26.7500</cost_rate>
  <cost_affect_cost>yes</cost_affect_cost>
  <cost_use_group_settings>no</cost_use_group_settings>
  <time_bank_allow_negative>no</time_bank_allow_negative>

<time_bank_allow_negative_use_group_settings>yes</time_bank_allow_negative_use_
group_settings>

<time_bank_overtime_calc_use_group_settings>yes</time_bank_overtime_calc_use_gr
oup_settings>
  <time_bank_overtime_calc_type>2</time_bank_overtime_calc_type>

<time_bank_overtime_calc_after_nbr_minutes>2400</time_bank_overtime_calc_after_
nbr_minutes>

<time_bank_overtime_calc_multiplier_factor>1.5000</time_bank_overtime_calc_mult
iplier_factor>
  <time_bank_balance_accumulated>600</time_bank_balance_accumulated>
  <time_bank_balance_vacations>4800</time_bank_balance_vacations>
  <time_bank_balance_sick>480</time_bank_balance_sick>
  <udf327 description="Position"><![CDATA[Marketing]]></udf327>
  <udf348 description="Part time">no</udf348>
  <employee_group>
    <employee_group_id>0</employee_group_id>
  </employee_group>
  <employee_group>
    <employee_group_id>5</employee_group_id>
  </employee_group>
</employee>
</response>

```

See request type **employee_edit** for information about fields.

employee_edit (Edit an existing employee)

XML request example:

```

<request app="orotimesheet" version="8" password="abcdef-123456">
  <employee_edit>
    <employee_id>14</employee_id>
    <notes><![CDATA[This employee is still a very good employee]]></notes>
  </employee_edit>
</request>

```

```
<email><![CDATA[Janet.White@orotimesheet.com]]></email>
<category_id>3</category_id>
<billing_rate>54.25</billing_rate>
<billing_rate_billable>yes</billing_rate_billable>
<billing_use_group_settings>no</billing_use_group_settings>
<hire_date>2006-09-14</hire_date>
<max_minutes_enterable>450</max_minutes_enterable>
<max_minutes_enterable_type>1</max_minutes_enterable_type>
</employee_edit>
</request>
```

employee_id: integer – Required - Id of the employee to edit. If the employee id specified does not exist, the request will fail.

email: string (60) - If you provide an e-mail address, the e-mail address must be in a valid e-mail format.

name: string(50) - Employee name. If the employee name already exists, the request will fail.

notes: text

category_id: integer – Id of the employee category.

category_description: string (35) – It allows you to specify the category using the category description instead of its id. The API will retrieve the category_id for you. If category_id is specified this field will be ignored.

sin: string (30)

hire_date: date

max_minutes_enterable: integer – Maximum minutes enterable per day or per week. See **max_minutes_enterable_type** below.

max_minutes_enterable_type: integer – 1=per day, 2=per week.

max_minutes_enterable_use_group_settings: string - You must specify yes or no.

minutes_usually_worked_per_week: integer – Number of minutes usually worked week.

minutes_usually_worked_per_week_use_group_settings: string - You must specify yes or no.

billing_rate: decimal

billing_rate_billable: string - You must specify yes or no.

billing_rate_type: integer – 0=Billing rate based on activity type rate, 1=Billing rate based on employee rate,

billing_use_group_settings: string - You must specify yes or no.

cost_rate: decimal

cost_affect_cost: string - You must specify yes or no.

cost_use_group_settings: string - You must specify yes or no.

time_bank_allow_negative: string - You must specify yes or no.

time_bank_allow_negative_use_group_settings: string - You must specify yes or no.

time_bank_overtime_calc_type: integer – 0=No overtime, 1=Overtime calculated per day, 2=Overtime calculated per week.

time_bank_overtime_calc_after_nbr_minutes: integer – Overtime will start to be calculated after the number of minutes indicated here.

time_bank_overtime_calc_multiplier_factor: decimal – Overtime will be multiplied by this value.

time_bank_overtime_calc_use_group_settings: string - You must specify yes or no.

For security reason, other fields of the employee records such as access or supervision rights cannot be edited via the API. You can only edit it via the application.

Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <employee_id>14</employee_id>
</response>
```

employee_id: Id of the updated employee.

employee_group_list (Get entire list of employee groups in a summary format)

Once the list of employee groups retrieved, you can use the **employee_group_get** request type to retrieve the complete record of a specific employee group using the **employee_group_id** retrieved.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <employee_group_list>
</employee_group_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <employee_group>
    <employee_group_id>0</employee_group_id>
    <description><![CDATA[Everybody]]></description>
  </employee_group>
  <employee_group>
    <employee_group_id>1</employee_group_id>
    <description><![CDATA[Engineers]]></description>
  </employee_group>
</response>
```

employee_group_get (Get an employee group record)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <employee_group_get>
    <employee_group_id>1</employee_group_id>
  </employee_group_get>
</request>
```

employee_group_id: integer – Required - Id of the employee group to retrieve. If the employee group id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
<employee_group>
  <employee_group_id>1</employee_group_id>
  <description><![CDATA[Engineers]]></description>
  <max_minutes_enterable>480</max_minutes_enterable>
  <max_minutes_enterable_type>1</max_minutes_enterable_type>
  <minutes_usually_worked_per_week>2400</minutes_usually_worked_per_week>
</employee_group>
```

```

<billing_rate>55.0000</billing_rate>
<billing_rate_type>1</billing_rate_type>
<billing_rate_billable>yes</billing_rate_billable>
<cost_rate>0.000</cost_rate>
<cost_affect_cost>yes</cost_affect_cost>
<time_bank_allow_negative>no</time_bank_allow_negative>
<time_bank_overtime_calc_type>2</time_bank_overtime_calc_type>

<time_bank_overtime_calc_after_nbr_minutes>2400</time_bank_overtime_calc_after_
nbr_minutes>

<time_bank_overtime_calc_multiplier_factor>1.5000</time_bank_overtime_calc_mult
iplier_factor>
  <employee>
    <employee_id>1</employee_id>
  </employee>
  <employee>
    <employee_id>8</employee_id>
  </employee>
  <employee>
    <employee_id>16</employee_id>
  </employee>
</employee_group>
</response>

```

See request type **employee_group_edit** for information about fields. Note that this request also returns the list of employees (employee id) that are member of the group.

employee_group_edit (Edit an existing employee group)

XML request example:

```

<request app="orotimesheet" version="8" password="abcdef-123456">
  <employee_group_edit>
    <employee_group_id>1</employee_group_id>
    <description><![CDATA[Engineers (Team #2)]]></description>
    <billing_rate>54.25</billing_rate>
    <billing_rate_billable>yes</billing_rate_billable>
  </employee_group_edit>
</request>

```

employee_group_id: integer – Required - Id of the employee group to edit. If the employee group id specified does not exist, the request will fail.

description: string(50) - Employee group description. If the employee group description already exists, the request will fail.

max_minutes_enterable: integer – Maximum minutes enterable per day or per week. See **max_minutes_enterable_type** below.

max_minutes_enterable_type: integer – 1=per day, 2=per week.

minutes_usually_worked_per_week: integer – Number of minutes usually worked week.

billing_rate: decimal

billing_rate_billable: string - You must specify yes or no.

billing_rate_type: integer – 0=Billing rate based on activity type rate, 1=Billing rate based on employee rate,

cost_rate: decimal

cost_affect_cost: string - You must specify yes or no.

cost_use_group_settings: string - You must specify yes or no.

time_bank_allow_negative: string - You must specify yes or no.

time_bank_overtime_calc_type: integer – 0=No overtime, 1=Overtime calculated per day, 2=Overtime calculated per week.

time_bank_overtime_calc_after_nbr_minutes: integer – Overtime will start to be calculated after the number of minutes indicated here.

time_bank_overtime_calc_multiplier_factor: decimal – Overtime will be multiplied by this value.

For security reason, other fields of the employee group records such as access or supervision rights cannot be edited via the API. You can only edit it via the application.

Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <employee_group_id>1</employee_group_id>
</response>
```

employee_group_id: Id of the updated employee group.

activity_type_list (Get entire list of activity types in a summary format)

Once the list of activity types retrieved, you can use the **activity_type_get** request type to retrieve the complete record of a specific activity type using the **activity_type_id** retrieved.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <activity_type_list>
</activity_type_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <activity_type>
    <activity_type_id>0</activity_type_id>
    <description><![CDATA[Management]]></description>
    <status>active</status>
  </activity_type>
  <activity_type>
    <activity_type_id>1</activity_type_id>
    <description><![CDATA[Search on internet]]></description>
    <status>inactive</status>
  </activity_type>
  <activity_type>
    <activity_type_id>2</activity_type_id>
    <description><![CDATA[Meeting]]></description>
    <status>active</status>
</response>
```

```
</activity_type>
</response>
```

activity_type_get (Get an activity type record)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <activity_type_get>
    <activity_type_id>2</activity_type_id>
  </activity_type_get>
</request>
```

customer_id: integer – Required - Id of the customer to retrieve. If the customer id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <activity_type>
    <activity_type_id>2</activity_type_id>
    <description><![CDATA[Meeting]]></description>
    <status>active</status>
    <category_id>0</category_id>
    <category_description><![CDATA[Default category]]></category_description>
    <billable>no</billable>
    <billing_rate>0.0000</billing_rate>
  </activity_type>
</response>
```

See request type **activity_type_add** for information about fields.

activity_type_add (Add a new activity type)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <activity_type_add>
    <description>Design</description>
    <status>active</status>
    <category_description>Design and Engineering</category_description>
    <billing_rate>75.00</billing_rate>
    <billable>yes</billable>
  </activity_type_add>
</request>
```

description: string(35) – Required – Activity type description. If the description already exists, the request will fail.

status: If specified, you must specify **active** or **inactive**. Default is active.

category_id: integer – If not specified, the default category will be used.

category_description: string (35) – It allows you to specify the category using the category description instead of its id. The API will retrieve the category_id for you. If category_id is specified this field will be ignored.

billing_rate: decimal – Billing rate of the activity type.

billable: string - You must specify yes or no.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <activity_type_id>39</activity_type_id>
</response>
```

activity_type_id: Id of the new created activity type.

activity_type_edit (Edit an existing activity type)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <activity_type_edit>
    <activity_type_id>39</activity_type_id>
    <billing_rate>80.00</billing_rate>
  </activity_type_edit>
</request>
```

activity_type_id: integer – Required - Id of the activity type to edit. If the activity type id specified does not exist, the request will fail.

See request type activity_type_add for information about other fields. Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <activity_type_id>39</activity_type_id>
</response>
```

activity_type_id: Id of the updated activity type.

activity_type_delete (Delete an activity type)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <activity_type_delete>
    <activity_type_id>39</activity_type_id>
  </activity_type_delete>
</request>
```

activity_type_id: integer – Required - Id of the activity type to delete. If the activity type id specified does not exist, the request will fail. To be able to delete the activity_type, no time transactions must be linked to that activity type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <activity_type_id>39</activity_type_id>
</response>
```

activity_type_id: Id of the deleted activity type.

customer_list (Get entire list of customers in a summary format)

Once the list of customers retrieved, you can use the **customer_get** request type to retrieve the complete record of a specific customer using the **customer_id** retrieved.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_list>
  </customer_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer>
    <customer_id>5271</customer_id>
    <code><![CDATA[XYZ-001]]></code>
    <name><![CDATA[XYZ Industries Inc.]]></name>
    <status>active</status>
    <nbr_projects>2</nbr_projects>
  </customer>
  <customer>
    <customer_id>5272</customer_id>
    <code><![CDATA[ABC-999]]></code>
    <name><![CDATA[ABC Inc.]]></name>
    <status>inactive</status>
    <nbr_projects>0</nbr_projects>
  </customer>
</response>
```

customer_get (Get a customer record)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_get>
    <customer_id>5271</customer_id>
```



```
</customer_get>
</request>
```

customer_id: integer – Required - Id of the customer to retrieve. If the customer id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer>
    <customer_id>5271</customer_id>
    <name><![CDATA[XYZ Industries Inc.]]></name>
    <code><![CDATA[XYZ-001]]></code>
    <address><![CDATA[330 Fortune Street
Quebec QC G1K 9C5]]></address>
    <contact><![CDATA[Mary Poppins]]></contact>
    <phone><![CDATA[418-524-5066]]></phone>
    <mobile><![CDATA[418-524-0000]]></mobile>
    <fax><![CDATA[418-524-1063]]></fax>
    <email><![CDATA[mary.poppins@OroTimesheet.com]]></email>
    <status>active</status>
    <category_id>0</category_id>
    <category_description><![CDATA[Default category]]></category_description>
    <tax_rate_id>1</tax_rate_id>
    <tax_rate_description><![CDATA[Default tax rate]]></tax_rate_description>
    <payment_terms><!--[CDATA[NET 30 DAYS]]--></payment_terms>
    <notes><![CDATA[Bla bla bla]]></notes>
    <nbr_projects>2</nbr_projects>
    <udf204 description="Salesman"><![CDATA[Bob White]]></udf204>
    <special_rate>
      <employee_id>1</employee_id>
      <description><!--[CDATA[John Doe]]--></description>
      <billable>yes</billable>
      <rate>65.0000</rate>
    </special_rate>
    <special_rate>
      <activity_type_id>25</activity_type_id>
      <description><!--[CDATA[Other]]--></description>
      <billable>no</billable>
      <rate>0.0000</rate>
    </special_rate>
    <special_rate>
      <activity_type_id>28</activity_type_id>
      <description><!--[CDATA[Management]]--></description>
      <billable>yes</billable>
      <rate>78.50</rate>
    </special_rate>
  </customer>
</response>
```

See request type **customer_add** for information about fields.

customer_add (Add a new customer)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
<customer_add>
  <code>XYZ-001</code>
  <name>XYZ Industries Inc.</name>
  <address><![CDATA[330 St-Vallier East
Suite 110
Quebec QC G1K 9C5]]></address>
  <notes><![CDATA[Bla bla bla]]></notes>
  <status>active</status>
  <category_description>Privileged customers</category_description>
  <udf204>Janet Wilson</udf204>
  <contact>Alex Martin</contact>
  <phone>418-524-5066</phone>
  <mobile>418-524-0000</mobile>
  <fax>418-524-1063</fax>
  <email>alex.martin@OroTimesheet.com</email>
  <payment_terms>NET 30 DAYS</payment_terms>
</customer_add>
</request>
```

code: string(30) – If specified, must be unique else the request will fail.

name: string(50) – Required – Customer name. If the customer name already exists, the request will fail.

address: text

notes: text

status: If specified, you must specify **active** or **inactive**. Default is active.

category_id: integer – If not specified, the default category will be used.

category_description: string (35) – It allows you to specify the category using the category description instead of its id. The API will retrieve the category_id for you. If category_id is specified this field will be ignored.

Tax_rate_id: integer – If not specified, the default tax rate will be used.

Tax_rate_description: string (35) – It allows you to specify the tax rate using the tax rate description instead of its id. The API will retrieve the tax_rate_id for you. If tax_rate_id is specified this field will be ignored.

contact: string (50) – Name of the contact to add.

phone: string (25)

mobile: string (25)

fax: string (25)

email: string (60) - If you provide an e-mail address, the e-mail address must be in a valid e-mail format.

payment_terms: string (30)

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer_id>5271</customer_id>
</response>
```

customer_id: Id of the new created customer.

customer_edit (Edit an existing customer)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_edit>
    <customer_id>5271</customer_id>
    <address><![CDATA[330 St-Vallier East
Suite 110-B
Quebec QC G1K 9C5]]></address>

  </customer_edit>
</request>
```

customer_id: integer – Required - Id of the customer to edit. If the customer id specified does not exist, the request will fail.

See request type **customer_add** for information about other fields. Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer_id>5271</customer_id>
</response>
```

customer_id: Id of the updated customer.

customer_delete (Delete a customer)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_delete>
    <customer_id>5271</customer_id>
  </customer_delete>
</request>
```

customer_id: integer – Required - Id of the customer to delete. If the customer id specified does not exist, the request will fail. To be able to delete the customer, no project must be linked to that customer.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer_id>5271</customer_id>
</response>
```

customer_id: Id of the deleted customer.

customer_special_rate_add (Add a special rate to an existing customer)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_special_rate_add>
    <customer_id>5271</customer_id>
    <activity_type_id>451</activity_type_id>
    <billable>yes</billable>
    <rate>75.25</rate>
  </customer_special_rate_add>
</request>
```

customer_id: integer – Required - Id of an existing customer. Refer to request types **customer_add** and **customer_list** for more information.

activity_type_id: integer – Required only if you add a special rate for an activity type- Id of an existing activity type. Refer to request types **activity_type_add** and **activity_type_list** for more information.

employee_id: integer – Required only if you add a special rate for an employee- Id of an existing employee. Refer to request types **employee_list** for more information.

billable: Required. You must specify **yes** or **no**.

rate: Decimal – Required - Must be different from 0.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer_id>3834</customer_id>
</response>
```

customer_id: Id of the updated customer.

customer_special_rate_delete (Delete a special rate from an existing customer)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_special_rate_delete>
    <customer_id>5271</customer_id>
    <activity_type_id>451</activity_type_id>
  </customer_special_rate_delete>
</request>
```

customer_id: integer – Required - Id of an existing customer. Refer to request types **customer_add** and **customer_list** for more information.

activity_type_id: integer – Required only if you delete a special rate for an activity type- Id of an existing activity type. Refer to request types **activity_type_add** and **activity_type_list** for more information.

employee_id: integer – Required only if you delete a special rate for an employee- Id of an existing employee. Refer to request types **employee_list** for more information.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
```

```
<customer_id>3834</customer_id>
</response>
```

customer_id: Id of the updated customer.

project_list (Get entire list of projects and sub-projects in a summary format)

Once the list of projects retrieved, you can use the **project_get** request type to retrieve the complete record of a specific project using the **project_id** retrieved.

Note that when a project does not contains any sub-project, the field **nbr_sub_projects** will be equals to 0. However, in the database, there is always a hidden sub-project even for projects that does not have any sub-projects. So, this sub-project will be listed as well in such case.

Optional filters

customer_id: integer – When specified, only projects for the specified customer will be returned.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_list>
    <customer_id>391</customer_id>
  </project_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project>
    <project_id>169</project_id>
    <code><!--[CDATA[P1]]--></code>
    <description><!--[CDATA[My project 1]]--></description>
    <customer_id>391</customer_id>
    <status>active</status>
    <has_sub_projects>yes</has_sub_projects>
    <nbr_sub_projects>2</nbr_sub_projects>
    <sub_project>
      <sub_project_id>215</sub_project_id>
      <description><!--[CDATA[phase 1]]--></description>
      <status>active</status>
    </sub_project>
    <sub_project>
      <sub_project_id>216</sub_project_id>
      <description><!--[CDATA[phase 2]]--></description>
      <status>active</status>
    </sub_project>
  </project>
  <project>
    <project_id>261</project_id>
    <code><!--[CDATA[P2]]--></code>
    <description><!--[CDATA[My project 2]]--></description>
    <customer_id>391</customer_id>
    <status>active</status>
```

```

<has_sub_projects>no</has_sub_projects>
<nbr_sub_projects>0</nbr_sub_projects>
<sub_project>
  <sub_project_id>781</sub_project_id>
  <description><!--[CDATA[This project does not have any sub-project]]-->
</description>
  <status>active</status>
</sub_project>
</project>
</response>

```

project_get (Get a project record)

XML request example:

```

<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_get>
    <project_id>5271</project_id>
  </project_get>
</request>

```

project_id: integer – Required - Id of the project to retrieve. If the project id specified does not exist, the request will fail.

Response successful example:

```

<response app="orotimesheet" version="8" status="1">
  <project>
    <project_id>5271</project_id>
    <description><![CDATA[My project 1]]></description>
    <code><![CDATA[P1]]></code>
    <customer_id>391</customer_id>
    <customer_name><![CDATA[Industries XYZ Inc.]]></customer_name>
    <status>active</status>
    <category_id>0</category_id>
    <category_description><![CDATA[Default Category]]></category_description>
    <project_manager_employee_id>510</project_manager_employee_id>
    <notes><![CDATA[]]></notes>
    <date_begin>2013-11-29</date_begin>
    <date_end>2013-12-31</date_end>
    <billing_mode>fixed_amount</billing_mode>
    <fixed_price_amount>7500.00</fixed_price_amount>
    <nbr_sub_projects>2</nbr_sub_projects>
    <budget_minutes>6000</budget_minutes>
    <budget_amount_billing_time>7500.00</budget_amount_billing_time>
    <budget_amount_billing_expenses>0.00</budget_amount_billing_expenses>
    <budget_amount_cost_time>0.00</budget_amount_cost_time>
    <budget_amount_cost_expenses>0.00</budget_amount_cost_expenses>
    <real_minutes>3000</real_minutes>
    <real_amount_billing_time>3750.00</real_amount_billing_time>
    <real_amount_billing_expenses>0.00</real_amount_billing_expenses>
    <real_amount_cost_time>0.00</real_amount_cost_time>
    <real_amount_cost_expenses>0.00</real_amount_cost_expenses>
  </project>
</response>

```

```
<real_amount_cost_expenses_refundable>0.00</real_amount_cost_expenses_refundable>
e>
  <udf322 description="Project Type"><![CDATA[Standard]]></udf322>
  <manager>
    <employee_id>341</employee_id>
  </manager>
  <manager>
    <employee_id>510</employee_id>
  </manager>
  <sub_project>
    <sub_project_id>6638</sub_project_id>
    <position>1</position>
    <description><![CDATA[Phase 1]]></description>
    <status>active</status>
    <fixed_price_amount>5625.00</fixed_price_amount>
    <time_bank_type>accumulated_time</time_bank_type>
    <budget_minutes>4500</budget_minutes>
    <budget_amount_billing_time>5625.00</budget_amount_billing_time>
    <budget_amount_billing_expenses>0.00</budget_amount_billing_expenses>
    <budget_amount_cost_time>0.00</budget_amount_cost_time>
    <budget_amount_cost_expenses>0.00</budget_amount_cost_expenses>
    <real_minutes>2250</real_minutes>
    <real_amount_billing_time>2812.50</real_amount_billing_time>
    <real_amount_billing_expenses>0.00</real_amount_billing_expenses>
    <real_amount_cost_time>0.00</real_amount_cost_time>
    <real_amount_cost_expenses>0.00</real_amount_cost_expenses>
  </sub_project>
  <sub_project>
    <sub_project_id>6639</sub_project_id>
    <position>2</position>
    <description><![CDATA[Phase 2]]></description>
    <status>active</status>
    <fixed_price_amount>1875.00</fixed_price_amount>
    <time_bank_type>accumulated_time</time_bank_type>
    <budget_minutes>1500</budget_minutes>
    <budget_amount_billing_time>1875.00</budget_amount_billing_time>
    <budget_amount_billing_expenses>0.00</budget_amount_billing_expenses>
    <budget_amount_cost_time>0.00</budget_amount_cost_time>
    <budget_amount_cost_expenses>0.00</budget_amount_cost_expenses>
    <real_minutes>750</real_minutes>
    <real_amount_billing_time>937.50</real_amount_billing_time>
    <real_amount_billing_expenses>0.00</real_amount_billing_expenses>
    <real_amount_cost_time>0.00</real_amount_cost_time>
    <real_amount_cost_expenses>0.00</real_amount_cost_expenses>
  </sub_project>
</project>
</response>
```

project_add (Add a new project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_add>
    <customer_id>417</customer_id>
    <code>P2</code>
    <description><![CDATA[My Project #2]]></description>
    <notes><![CDATA[This is an important project.]]></notes>
    <status>active</status>
    <category_id>2</category_id>
    <date_begin>2015-02-14</date_begin>
    <date_end>2015-11-27</date_end>
    <billing_mode>per_hour</billing_mode>
    <fixed_price_amount>0</fixed_price_amount>
    <project_manager_employee_id>74</project_manager_employee_id>
    <time_bank_type>accumulated_time</time_bank_type>
    <budget_minutes>30000</budget_minutes>
    <budget_amount_billing_time>37500.00</budget_amount_billing_time>
    <budget_amount_billing_expenses>15000.00</budget_amount_billing_expenses>
    <budget_amount_cost_time>26250.00</budget_amount_cost_time>
    <budget_amount_cost_expenses>2500.00</budget_amount_cost_expenses>
    <udf322>Special Project</udf322>
  </project_add>
</request>
```

customer_id: integer – Required – Customer id that will be linked to the project. The customer id must exist else the request will fail.

code: string(30) – If specified, must be unique else the request will fail.

description: string(65) – Required – Project description. If the description already exists, the request will fail.

notes: text

status: If specified, you must specify **active** or **inactive**. Default is active.

category_id: integer – If not specified, the default category will be used.

category_description: string (35) – It allows you to specify the category using the category description instead of its id. The API will retrieve the category_id for you. If category_id is specified this field will be ignored.

date_begin: date

date_end: date

billing_mode: string – If specified, must be per_hour or fixed_amount. If not specified, the default will be per_hour.

project_manager_employee_id: integer – Id of the employee that will be the main manager of the project. The employee id must exist else the request will fail. If not specified, the default will be the employee 1 that is the default employee id in the system and this employee cannot be deleted.

fixed_price_amount: decimal. Default 0.

time_bank_type: string – Must be accumulated_time, vacations, sick or ignore. If not specified, the default will be accumulated_time.

budget_minutes: integer – Budget - Number of minutes. Default 0.

budget_amount_billing_time: decimal – Budget – Amount billable (Time). Default 0.

budget_amount_billing_expenses: decimal – Budget – Amount billable (Expenses). Default 0.

budget_amount_cost_time: decimal – Budget – Amount cost (Time). Default 0.

budget_amount_cost_expenses: decimal – Budget – Amount cost (Expenses). Default 0.

Fields `fixed_price_amount`, `time_bank_type`, `budget_minutes`, `budget_amount_billing_time`, `budget_amount_billing_expenses`, `budget_amount_cost_time` and `budget_amount_cost_expenses` can be specified when you add a new project. However, once the project created, to edit these fields using the API, you must use the request type **sub_project_edit** instead of the request type **project_edit**. The reason is that this information is kept at the sub-project level. See notes about sub-projects below.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>8304</project_id>
  <sub_project_id>9731</sub_project_id>
</response>
```

project_id: Id of the new created project.

sub_project_id: Id of the new created sub-project.

Notes about sub-projects:

When you create a new project, by default, the project does not contain any sub-project. If you want, you can use the request type **sub_project_add** to add sub-projects to your newly created project.

However, into the database, a project always contains at least one sub-project even if this project does not officially have any sub-projects. So, in such case, the sub-project is hidden to the user but it exists physically into the database. This is for this reason that when you create a new project, the request returns a **project_id** as well as a **sub_project id**.

When you add a sub-project to a project that does not officially have any sub-project yet, no new sub-project is created in fact. This is just the existing sub-project that become visible to the user. New sub-projects will be added to the database only when you add a second sub-project to the project. This is the same thing when you delete the last sub-project of a project. The last sub-project become hidden and is not erased from the database.

project_edit (Edit an existing project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_edit>
    <project_id>8304</project_id>
    <description><![CDATA[My new project description]]></description>
  </project_edit>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

See request type **project_add** for information about other fields. Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>8304</project_id>
</response>
```

project_id: Id of the updated project.

project_delete (Delete a project)**XML request example:**

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_delete>
    <project_id>5271</project_id>
  </project_delete>
</request>
```

project_id: integer – Required - Id of the project to delete. If the project id specified does not exist, the request will fail. To be able to delete the project, no time transactions must be linked to that project.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>5271</project_id>
</response>
```

project_id: Id of the deleted project.

project_member_add (Add a member (employee or employee group) to a project)**XML request example:**

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_member_add>
    <project_id>755</project_id>
    <employee_id>2</employee_id>
  </project_member_add>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

employee_id: integer – Required only if you add an employee.

employee_group_id: integer – Required only if you add an employee_group.

So, you can either add an employee or an employee group using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>755</project_id>
  <employee_id>2</employee_id>
</response>
```

project_id: Id of the updated project.

employee_id: Id of the added employee. (Only when you add an employee to the project)

employee_group_id: Id of the added employee group. (Only when you add an employee group to the project)

project_member_delete (Delete a member (employee or employee group) from a project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_member_delete>
    <project_id>755</project_id>
    <employee_id>2</employee_id>
  </project_member_delete>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

employee_id: integer – Required only if you delete an employee.

employee_group_id: integer – Required only if you delete an employee_group.

So, you can either delete an employee or an employee group using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>755</project_id>
  <employee_id>2</employee_id>
</response>
```

project_id: Id of the updated project.

employee_id: Id of the deleted employee. (Only when you delete an employee from the project)

employee_group_id: Id of the deleted employee group. (Only when you delete an employee group from the project)

project_additional_manager_add (Add an additional manager (employee or employee group) to a project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_additional_manager_add>
```

```
<project_id>755</project_id>
<employee_id>2</employee_id>
</project_additional_manager_add>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

employee_id: integer – Required only if you add an employee.

employee_group_id: integer – Required only if you add an employee group.

So, you can either add an employee or an employee group using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
<project_id>755</project_id>
<employee_id>2</employee_id>
</response>
```

project_id: Id of the updated project.

employee_id: Id of the added employee. (Only when you add an employee to the project)

employee_group_id: Id of the added employee group. (Only when you add an employee group to the project)

project_additional_manager_delete (Delete a manager (employee or employee group) from a project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
<project_additional_manager_delete>
<project_id>755</project_id>
<employee_id>2</employee_id>
</project_additional_manager_delete>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

employee_id: integer – Required only if you delete an employee.

employee_group_id: integer – Required only if you delete an employee group.

So, you can either delete an employee or an employee group using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
<project_id>755</project_id>
<employee_id>2</employee_id>
</response>
```

project_id: Id of the updated project.

employee_id: Id of the deleted employee. (Only when you delete an employee from the project)

employee_group_id: Id of the deleted employee group. (Only when you delete an employee group from the project)

project_activity_type_add (Add an activity type or activity type category to a project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_activity_type_add>
    <project_id>755</project_id>
    <activity_type_id>2</activity_type_id>
  </project_activity_type_add>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

activity_type_id: integer – Required only if you add an activity type.

activity_type_category_id: integer – Required only if you add an activity type category.

So, you can either add an activity type or an activity type category using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>755</project_id>
  <activity_type_id>2</activity_type_id>
</response>
```

project_id: Id of the updated project.

activity_type_id: Id of the added activity type. (Only when you add an activity type to the project)

activity_type_category_id: Id of the added activity type category. (Only when you add an activity type category to the project)

project_activity_type_delete (Delete an activity type or activity type category from a project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_activity_type_delete>
    <project_id>755</project_id>
    <activity_type_id>2</activity_type_id>
  </project_activity_type_delete>
</request>
```

project_id: integer – Required - Id of the project to edit. If the project id specified does not exist, the request will fail.

activity_type_id: integer – Required only if you delete an activity type.

activity_type_category_id: integer – Required only if you delete an activity type category.

So, you can either delete an activity type or an activity type category using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>755</project_id>
  <activity_type_id>2</activity_type_id>
</response>
```

project_id: Id of the updated project.

activity_type_id: Id of the deleted activity type. (Only when you delete an activity type from the project)

activity_type_category_id: Id of the deleted activity type category. (Only when you delete an activity type category from the project)

project_special_rate_add (Add a special rate to an existing project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_special_rate_add>
    <project_id>755</project_id>
    <activity_type_id>451</activity_type_id>
    <billable>yes</billable>
    <rate>75.25</rate>
  </project_special_rate_add>
</request>
```

project_id: integer – Required - Id of an existing project. Refer to request types **project_add** and **project_list** for more information.

activity_type_id: integer – Required only if you add a special rate for an activity type- Id of an existing activity type. Refer to request types **activity_type_add** and **activity_type_list** for more information.

employee_id: integer – Required only if you add a special rate for an employee- Id of an existing employee. Refer to request types **employee_list** for more information.

billable: Required. You must specify **yes** or **no**.

rate: Decimal – Required - Must be different from 0.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>755</project_id>
</response>
```

project_id: Id of the updated project.

project_special_rate_delete (Delete a special rate from an existing project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_special_rate_delete>
    <project_id>755</project_id>
    <activity_type_id>451</activity_type_id>
  </project_special_rate_delete>
</request>
```

project_id: integer – Required - Id of an existing project. Refer to request types **project_add** and **project_list** for more information.

activity_type_id: integer – Required only if you delete a special rate for an activity type- Id of an existing activity type. Refer to request types **activity_type_add** and **activity_type_list** for more information.

employee_id: integer – Required only if you delete a special rate for an employee- Id of an existing employee. Refer to request types **employee_list** for more information.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_id>755</project_id>
</response>
```

project_id: Id of the updated project.

sub_project_add (Add a sub-project to a project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_add>
    <project_id>417</project_id>
    <description><![CDATA[Phase 1]]></description>
    <position>1</position>
    <status>active</status>
    <fixed_price_amount>0</fixed_price_amount>
    <time_bank_type>accumulated_time</time_bank_type>
    <budget_minutes>30000</budget_minutes>
    <budget_amount_billing_time>37500.00</budget_amount_billing_time>
    <budget_amount_billing_expenses>15000.00</budget_amount_billing_expenses>
    <budget_amount_cost_time>26250.00</budget_amount_cost_time>
    <budget_amount_cost_expenses>2500.00</budget_amount_cost_expenses>
  </sub_project_add>
</request>
```

project_id: integer – Required – Project id for which you want to add the sub-project The project id must exist else the request will fail.

description: string(50) – Required – Sub-project description. If the description already exists for another sub-project in a same project, the request will fail.

position: integer – Position order of the sub-project in the list. Default is 0.

status: If specified, you must specify **active** or **inactive**. Default is active.

fixed_price_amount: decimal. Default 0.

time_bank_type: string – Must be accumulated_time, vacations, sick or ignore. If not specified, the default will be accumulated_time.

budget_minutes: integer – Budget - Number of minutes. Default 0.

budget_amount_billing_time: decimal – Budget – Amount billable (Time). Default 0.

budget_amount_billing_expenses: decimal – Budget – Amount billable (Expenses). Default 0.

budget_amount_cost_time: decimal – Budget – Amount cost (Time). Default 0.

budget_amount_cost_expenses: decimal – Budget – Amount cost (Expenses). Default 0.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>9733</sub_project_id>
</response>
```

sub_project_id: Id of the new created sub-project.

sub_project_edit (Edit an existing sub-project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_edit>
    <sub_project_id>9733</sub_project_id>
    <description><![CDATA[My new sub-project description]]</description>
  </sub_project_edit>
</request>
```

sub_project_id: integer – Required - Id of the sub-project to edit. If the sub-project id specified does not exist, the request will fail.

See request type **sub_project_add** for information about other fields. Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>9733</sub_project_id>
</response>
```

sub_project_id: Id of the updated project.

sub_project_delete (Delete a sub-project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
```



```
<sub_project_delete>
  <sub_project_id>9733</sub_project_id>
</sub_project_delete>
</request>
```

sub_project_id: integer – Required - Id of the sub-project to delete. If the sub-project id specified does not exist, the request will fail. To be able to delete the sub-project, no time transactions must be linked to that sub-project.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>9733</sub_project_id>
</response>
```

sub_project_id: Id of the deleted sub-project.

See notes about sub-projects in the **project_add** request type section above.

sub_project_activity_type_hide (Hide an activity type or activity type category for a sub-project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_activity_type_hide>
    <sub_project_id>298</sub_project_id>
    <activity_type_id>28</activity_type_id>
  </sub_project_activity_type_hide>
</request>
```

sub_project_id: integer – Required - Id of the sub-project to edit. If the sub-project id specified does not exist, the request will fail.

activity_type_id: integer – Required only if you hide an activity type.

activity_type_category_id: integer – Required only if you hide an activity type category.

So, you can either hide an activity type or an activity type category using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>298</sub_project_id>
  <activity_type_id>28</activity_type_id>
</response>
```

sub_project_id: Id of the updated sub-project.

activity_type_id: Id of the hidden activity type. (Only when you hide an activity type)

activity_type_category_id: Id of the hidden activity type category. (Only when you hide an activity type category)

sub_project_activity_type_unhide (Unhide an activity type or activity type category from a sub-project)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_activity_type_unhide>
    <sub_project_id>298</sub_project_id>
    <activity_type_id>28</activity_type_id>
  </sub_project_activity_type_unhide>
</request>
```

sub_project_id: integer – Required - Id of the sub-project to edit. If the sub-project id specified does not exist, the request will fail.

activity_type_id: integer – Required only if you unhide an activity type.

activity_type_category_id: integer – Required only if you unhide an activity type category.

So, you can either unhide an activity type or an activity type category using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>298</sub_project_id>
  <activity_type_id>28</activity_type_id>
</response>
```

sub_project_id: Id of the updated sub-project.

activity_type_id: Id of the unhidden activity type. (Only when you unhide an activity type)

activity_type_category_id: Id of the unhidden activity type category. (Only when you unhide an activity type category)

sub_project_budget_distribution_edit (Set budget distribution for a sub-project and an activity type, an employee or an expense type)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <sub_project_budget_distribution_edit>
    <sub_project_id>298</sub_project_id>
    <activity_type_id>28</activity_type_id>
    <budget_minutes>600</budget_minutes>
    <budget_amount_billing_time>500</budget_amount_billing_time>
    <budget_amount_billing_expenses>105.49</budget_amount_billing_expenses>
    <budget_amount_cost_time>250</budget_amount_cost_time>
    <budget_amount_cost_expenses>50</budget_amount_cost_expenses>
  </sub_project_budget_distribution_edit>
</request>
```

sub_project_id: integer – Required - Id of the sub-project to set the budget distribution for. If the sub-project id specified does not exist, the request will fail.

activity_type_id: integer – Required only if you set the budget distribution for an activity type.

employee_id: integer – Required only if you set the budget distribution for an employee.

expense_type_id: integer – Required only if you set the budget distribution for an expense type.

So, you can set budget distribution per activity type, employee or expense type using the same request type.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <sub_project_id>298</sub_project_id>
  <activity_type_id>28</activity_type_id>
</response>
```

sub_project_id: Id of the updated sub-project.

activity_type_id: Id of the updated activity type. (Only when you set budget distribution for an activity type)

employee_id: Id of the updated employee. (Only when you set budget distribution for an employee)

expense_type_id: Id of the updated expense type. (Only when you set budget distribution for an expense type)

timesheet_tx_list (Get list of timesheet transactions in a summary format)

Once the list of transactions retrieved, you can use the **timesheet_tx_get** request type to retrieve the complete record of a specific timesheet transaction using the **timesheet_tx_id** retrieved.

Optional filters

customer_id: integer – When specified, only transactions where projects are linked to the specified customer will be returned.

project_id: integer – When specified, only transactions for the specified project will be returned.

sub_project_id: integer – When specified, only transactions for the specified sub-project will be returned.

employee_id: integer – When specified, only transactions for the specified employee will be returned.

activity_type_id: integer – When specified, only transactions for the specified activity type will be returned.

invoice_id: integer – When specified, only invoiced transactions related to the specified invoice will be returned.

date_from: date – When specified, only transactions between **date_from** and **date_to** will be returned. **date_to** must be obligatory specified as well.

date_to: date – When specified, only transactions between **date_from** and **date_to** will be returned. **date_from** must be obligatory specified as well.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_tx_list>
    <employee_id>28</employee_id>
    <date_from>2014-09-15</date_from>
    <date_to>2014-09-21</date_to>
  </timesheet_tx_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <timesheet_tx>
```

```

<timesheet_tx_id>2940</timesheet_tx_id>
<date>2014-09-16</date>
<employee_id>28</employee_id>
<customer_id>380</customer_id>
<project_id>541</project_id>
<sub_project_id>987</sub_project_id>
<activity_type_id>20</activity_type_id>
<invoiced>yes</invoiced>
<invoice_id>263</invoice_id>
<invoice_line_id>617</invoice_line_id>
<invoice_numero>141</invoice_numero>
<shift>1</shift>
<minutes>60</minutes>
<cost_amount>16.80</cost_amount>
<billing_amount>36.00</billing_amount>
<expenses_nbr_tx>2</expenses_nbr_tx>
<expenses_cost_amount>16.24</expenses_cost_amount>
<expenses_billable_amount>23.94</expenses_billable_amount>
</timesheet_tx>
<timesheet_tx>
  <timesheet_tx_id>2942</timesheet_tx_id>
  <date>2014-09-16</date>
  <employee_id>28</employee_id>
  <customer_id>381</customer_id>
  <project_id>544</project_id>
  <sub_project_id>687</sub_project_id>
  <activity_type_id>21</activity_type_id>
  <invoiced>no</invoiced>
  <invoice_id></invoice_id>
  <invoice_line_id></invoice_line_id>
  <invoice_numero></invoice_numero>
  <shift>1</shift>
  <minutes>15</minutes>
  <cost_amount>4.20</cost_amount>
  <billing_amount>9.00</billing_amount>
  <expenses_nbr_tx>0</expenses_nbr_tx>
  <expenses_cost_amount>0.00</expenses_cost_amount>
  <expenses_billable_amount>0.00</expenses_billable_amount>
</timesheet_tx>
</response>

```

timesheet_tx_get (Get a timesheet transaction record)

XML request example:

```

<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_tx_get>
    <timesheet_tx_id>2940</timesheet_tx_id>
  </timesheet_tx_get>
</request>

```

timesheet_tx_id: integer – Required - Id of the timesheet transaction to retrieve. If the transaction id specified does not exist, the request will fail.

Response successful example:

```

<response app="orotimesheet" version="8" status="1">
  <timesheet_tx>
    <timesheet_tx_id>2940</timesheet_tx_id>
    <date>2014-09-16</date>
    <employee_id>28</employee_id>
    <employee_name><![CDATA[Bob White]]></employee_name>
    <customer_id>380</customer_id>
    <customer_name><![CDATA[ABC Inc.]]></customer_name>
    <customer_code><![CDATA[ABC]]></customer_code>
    <project_id>541</project_id>
    <project_description><![CDATA[Project test]]></project_description>
    <project_code><![CDATA[PT-001]]></project_code>
    <sub_project_id>987</sub_project_id>
    <sub_project_description><![CDATA[Phase 1]]></sub_project_description>
    <activity_type_id>20</activity_type_id>
    <activity_type_description><![CDATA[Meeting]]></activity_type_description>
    <invoiced>yes</invoiced>
    <invoice_id>263</invoice_id>
    <invoice_line_id>617</invoice_line_id>
    <invoice_numero>141</invoice_numero>
    <shift>1</shift>
    <minutes>60</minutes>
    <affect_cost>yes</affect_cost>
    <hourly_cost>16.80</hourly_cost>
    <cost_amount>16.80</cost_amount>
    <billable>yes</billable>
    <billing_rate>36.00</billing_rate>
    <billing_amount>36.00</billing_amount>
    <time_in>00:00:00</time_in>
    <time_out>00:00:00</time_out>
    <notes><![CDATA[I went at the customer site for a special meeting.]]></notes>
    <created_by_employee_id>28</created_by_employee_id>
    <created_on>2014-09-16 16:24:45</created_on>
    <last_updated_on></last_updated_on>
    <expenses_nbr_tx>2</expenses_nbr_tx>
    <expenses_cost_amount>32.90</expenses_cost_amount>
    <refundable_amount_without_taxes>32.90</refundable_amount_without_taxes>
    <refundable_amount_tax_1>0.63</refundable_amount_tax_1>
    <refundable_amount_tax_2>1.28</refundable_amount_tax_2>
    <refundable_amount_total>34.81</refundable_amount_total>
    <expenses_billable_amount>25.00</expenses_billable_amount>
    <udf328 description="Work Type"><![CDATA[normal]]></udf328>
    <expense>
      <expense_id>141</expense_id>
      <expense_type_id>16</expense_type_id>
      <expense_type_description><![CDATA[Photocopies]]></expense_type_description>
      <quantity>50.0000</quantity>
      <unit_cost>0.2500</unit_cost>
      <cost_amount>12.50</cost_amount>
      <refundable_to_employee>yes</refundable_to_employee>
      <refundable_amount_without_taxes>12.50</refundable_amount_without_taxes>
      <refundable_amount_tax_1>0.63</refundable_amount_tax_1>
      <refundable_amount_tax_2>1.28</refundable_amount_tax_2>
      <refundable_amount_total>14.41</refundable_amount_total>
    </expense>
  </timesheet_tx>
</response>

```

```

    <billable_to_customer>yes</billable_to_customer>
    <unit_price>0.5000</unit_price>
    <billable_amount>25.00</billable_amount>
    <notes><![CDATA[Photocopies for the meeting]]></notes>
  </expense>
  <expense>
    <expense_id>142</expense_id>
    <expense_type_id>20</expense_type_id>
    <expense_type_description><![CDATA[Travel]]></expense_type_description>
    <quantity>40.0000</quantity>
    <unit_cost>0.5100</unit_cost>
    <cost_amount>20.40</cost_amount>
    <refundable_to_employee>yes</refundable_to_employee>
    <refundable_amount_without_taxes>20.40</refundable_amount_without_taxes>
    <refundable_amount_tax_1>0.00</refundable_amount_tax_1>
    <refundable_amount_tax_2>0.00</refundable_amount_tax_2>
    <refundable_amount_total>20.40</refundable_amount_total>
    <billable_to_customer>no</billable_to_customer>
    <unit_price>0.0000</unit_price>
    <billable_amount>0.00</billable_amount>
    <notes><![CDATA[]]></notes>
  </expense>
</timesheet_tx>
</response>

```

timesheet_tx_add (Add a timesheet transaction)

XML request example:

```

<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_tx_add>
    <sub_project_id>987</sub_project_id>
    <employee_id>28</employee_id>
    <activity_type_id>20</activity_type_id>
    <date>2014-07-23</date>
    <shift>1</shift>
    <minutes>90</minutes>
    <affect_cost>yes</affect_cost>
    <hourly_cost>15.50</hourly_cost>
    <billable>yes</billable>
    <billing_rate>47.00</billing_rate>
    <time_in>08:30:00</time_in>
    <time_out>10:00:00</time_out>
    <notes><![CDATA[Bla bla bla...]]></notes>
  </timesheet_tx_add>
</request>

```

sub_project_id: integer – Required – Sub-project id that will be linked to the transaction. The sub-project id must exist else the request will fail.

employee_id: integer – Required – Employee id that will be linked to the transaction. The employee id must exist else the request will fail.

activity_type_id: integer – Required – Activity type id that will be linked to the transaction. The activity type id must exist else the request will fail.

date_begin: date – Required.
shift: integer - If specified, must be between 1 and 10. Default 1.
minutes: integer – Required. Number of minutes worked.
affect_cost: string – If specified, must be **yes** or **no**. Default is **yes**.
hourly_cost: decimal – Default 0.
billable: string – If specified, must be **yes** or **no**. Default is **yes**.
billing_rate: decimal – Default 0.
time_in: time. Default 00:00:00.
time_out: time. Default 00:00:00.
notes: text

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <timesheet_tx_id>3151</timesheet_tx_id>
</response>
```

timesheet_tx_id: Id of the new created timesheet transaction.

timesheet_tx_edit (Edit an existing timesheet transaction)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_tx_edit>
    <timesheet_tx_id>3151</timesheet_tx_id>
    <billing_rate>50.00</billing_rate>
  </timesheet_tx_edit>
</request>
```

timesheet_tx_id: integer – Required - Id of the timesheet transaction to edit. If the transaction id specified does not exist, the request will fail.

See request type **timesheet_tx_add** for information about other fields. Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <timesheet_tx_id>3151</timesheet_tx_id>
</response>
```

timesheet_tx_id: Id of the updated timesheet transaction.

timesheet_tx_delete (Delete a timesheet transaction)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_tx_delete>
    <timesheet_tx_id>3151</timesheet_tx_id>
  </timesheet_tx_delete>
</request>
```

timesheet_tx_id: integer – Required - Id of the timesheet transaction to delete. If the timesheet transaction id specified does not exist, the request will fail. To be able to delete a timesheet transaction, the transaction must not be invoiced nor in a closed fiscal period nor in a week that was approved.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <timesheet_tx_id>3151</timesheet_tx_id>
</response>
```

timesheet_tx_id: Id of the deleted timesheet transaction.

expense_get (Get an expense transaction record)**XML request example:**

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <expense_get>
    <expense_id>141</expense_id>
  </expense_get>
</request>
```

expense_id: integer – Required - Id of the expense transaction to retrieve. If the transaction id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <expense>
    <timesheet_tx_id>2940</timesheet_tx_id>
    <expense_id>141</expense_id>
    <expense_type_id>16</expense_type_id>
    <expense_type_description><![CDATA[Photocopies]]></expense_type_description>
    <quantity>50.0000</quantity>
    <unit_cost>0.2500</unit_cost>
    <cost_amount>12.50</cost_amount>
    <refundable_to_employee>yes</refundable_to_employee>
    <refundable_amount_without_taxes>12.50</refundable_amount_without_taxes>
    <refundable_amount_tax_1>0.63</refundable_amount_tax_1>
    <refundable_amount_tax_2>1.28</refundable_amount_tax_2>
    <refundable_amount_total>14.41</refundable_amount_total>
    <billable_to_customer>yes</billable_to_customer>
    <unit_price>0.5000</unit_price>
    <billable_amount>25.00</billable_amount>
    <notes><![CDATA[Photocopies for the meeting]]></notes>
```



```
</expense>
</response>
```

expense_add (Add an expense to an existing timesheet transaction)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <expense_add>
    <timesheet_tx_id>2940</timesheet_tx_id>
    <expense_type_id>16</expense_type_id>
    <quantity>50.00</quantity>
    <unit_cost>0.25</unit_cost>
    <refundable_to_employee>yes</refundable_to_employee>
    <refundable_amount_tax_1>0.63</refundable_amount_tax_1>
    <refundable_amount_tax_2>1.28</refundable_amount_tax_2>
    <billable_to_customer>yes</billable_to_customer>
    <unit_price>0.5000</unit_price>
    <notes><![CDATA[Photocopies for the meeting]]></notes>
  </expense_add>
</request>
```

timesheet_tx_id: integer – Required – Timesheet transaction id which you want to add expense to. The timesheet transaction id must exist else the request will fail.

expense_type_id: integer – Required – Expense type id. The expense type id must exist else the request will fail.

quantity: decimal – Required.

unit_cost: decimal – Default 0.

refundable_to_employee: string – If specified, must be **yes** or **no**. Default is **no**.

refundable_amount_tax_1: decimal – Default 0.

refundable_amount_tax_2: decimal – Default 0.

unit_price: decimal – Default 0.

billable_to_customer: string – If specified, must be **yes** or **no**. Default is **no**.

notes: text

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <expense_id>258</expense_id>
</response>
```

expense_id: Id of the new created expense transaction.

expense_edit (Edit an existing expense transaction)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
```

```
<expense_edit>
  <expense_id>258</expense_id>
  <notes>Bla bla bla</notes>
</expense_edit>
</request>
```

expense_id: integer – Required - Id of the expense transaction to edit. If the transaction id specified does not exist, the request will fail.

See request type **expense_add** for information about other fields. Of course, fields not specified in your XML request will not be updated and will not be erased either.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <expense_id>258</expense_id>
</response>
```

expense_id: Id of the updated expense transaction.

expense_delete (Delete an expense transaction)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <expense_delete>
    <expense_id>258</expense_id>
  </expense_delete>
</request>
```

expense_id: integer – Required - Id of the expense transaction to delete. If the expense iid specified does not exist, the request will fail. To be able to delete an expense transaction, the parent timesheet transaction must not be invoiced nor in a closed fiscal period nor in a week that was approved.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <expense_id>258</expense_id>
</response>
```

expense_id: Id of the deleted expense transaction.

timesheet_week_state_get (Get the state of the timesheet of an employee/week)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_week_state_get>
    <employee_id>35</employee_id>
```

```
<date>2014-07-23</date>
</timesheet_week_state_get>
</request>
```

employee_id: integer – Required - Id of the employee. If the employee id specified does not exist, the request will fail.

date: date – Required – Date of the week you want to query. For example, in the application, if you configured the first day of the week to be Monday and you want to get the timesheet state for the week of July 23, 2014, you can pass any date from Mon July 21, 2014 to Sun July 27, 2014 and you will get the same result since these dates belong to the same week.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <timesheet_week_state>
    <employee_id>35</employee_id>
    <state>not_submitted</state>
    <week_date_begin>2014-07-21</week_date_begin>
    <week_date_end>2014-07-27</week_date_end>
    <reason_reject><![CDATA[]]></reason_reject>
  </timesheet_week_state>
</response>
```

timesheet_week_state_set (Set the state of the timesheet of an employee/week)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <timesheet_week_state_set>
    <employee_id>35</employee_id>
    <date>2014-07-23</date>
    <state>rejected</state>
    <reason_reject><![CDATA[Hours are missing.]]></reason_reject>
  </timesheet_week_state_set>
</request>
```

employee_id: integer – Required - Id of the employee. If the employee id specified does not exist, the request will fail.

date: date – Required – Date of the week you want to set. For example, in the application, if you configured the first day of the week to be Monday and you want to get the timesheet state for the week of July 23, 2014, you can pass any date from Mon July 21, 2014 to Sun July 27, 2014 and you will get the same result since these dates belong to the same week.

state: string – Required. Must be **not_submitted**, **submitted**, **approved** or **rejected**.

reason_reject: text – Can be used only when you set the stats to rejected else this field is ignored.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <employee_id>35</employee_id>
  <week_date_begin>2014-07-21</week_date_begin>
  <week_date_end>2014-07-27</week_date_end>
```

```
<state>rejected</state>
</response>
```

employee_id: Id of the employee.
week_date_begin: Date of the first day of the week.
week_date_end: Date of the last day of the week.
state: New state of the week.

time_bank_update_list (Get entire list of time banks updates in a summary format)

Once the list of time bank updates retrieved, you can use the **time_bank_update_get** request type to get detailed information about a specific time bank update using the **time_bank_update_id** retrieved.

Optional filters

employee_group_id: integer – When specified, only time bank updates for the specified employee group will be returned.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <time_bank_update_list>
    <employee_group_id>6</employee_group_id>
  </time_bank_update_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <time_bank_update>
    <time_bank_update_id>209</time_bank_update_id>
    <date_from>2014-07-21</date_from>
    <date_to>2014-07-27</date_to>
    <number_weeks>1</number_weeks>
    <minutes_total>7800</minutes_total>
    <minutes_paid>7200</minutes_paid>
    <employee_group_id>6</employee_group_id>
    <employee_group_description><![CDATA[Engineers]]></employee_group_description>
  </time_bank_update>
  <time_bank_update>
    <time_bank_update_id>213</time_bank_update_id>
    <date_from>2014-07-28</date_from>
    <date_to>2014-08-03</date_to>
    <number_weeks>1</number_weeks>
    <minutes_total>7380</minutes_total>
    <minutes_paid>7200</minutes_paid>
    <employee_group_id>6</employee_group_id>
    <employee_group_description><![CDATA[Engineers]]></employee_group_description>
  </time_bank_update>
</response>
```

time_bank_update_get (Get detailed information about a time bank update)**XML request example:**

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <time_bank_update_get>
    <time_bank_update_id>209</time_bank_update_id>
  </time_bank_update_get>
</request>
```

time_bank_update_id: integer – Required - Id of the time bank update to retrieve. If the id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <time_bank_update>
    <time_bank_update_id>209</time_bank_update_id>
    <date_from>2014-07-21</date_from>
    <date_to>2014-08-27</date_to>
    <number_weeks>1</number_weeks>
    <employee_group_id>6</employee_group_id>
    <employee_group_description><![CDATA[Engineers]]></employee_group_description>
    <minutes_regular_time>7200</minutes_regular_time>
    <minutes_overtime>600</minutes_overtime>
    <minutes_vacations>0</minutes_vacations>
    <minutes_sick>0</minutes_sick>
    <minutes_ignored>0</minutes_ignored>
    <minutes_total>7800</minutes_total>
    <minutes_expected_work>7200</minutes_expected_work>
    <minutes_paid>7200</minutes_paid>
    <minutes_difference>600</minutes_difference>
    <time_bank_accumulated_before>21900</time_bank_accumulated_before>
    <time_bank_accumulated_after>22500</time_bank_accumulated_after>
    <time_bank_vacations_before>100800</time_bank_vacations_before>
    <time_bank_vacations_after>100800</time_bank_vacations_after>
    <time_bank_sick_before>0</time_bank_sick_before>
    <time_bank_sick_after>0</time_bank_sick_after>
    <time_bank_tx>
      <time_bank_tx_id>568</time_bank_tx_id>
      <employee_id>28</employee_id>
      <employee_name><![CDATA[Bob White]]></employee_name>
      <minutes_regular_time>2400</minutes_regular_time>
      <minutes_overtime>240</minutes_overtime>
      <minutes_vacations>0</minutes_vacations>
      <minutes_sick>0</minutes_sick>
      <minutes_ignored>0</minutes_ignored>
      <minutes_total>2640</minutes_total>
      <minutes_expected_work>2400</minutes_expected_work>
      <minutes_paid>2400</minutes_paid>
      <minutes_difference>240</minutes_difference>
      <time_bank_accumulated_before>1800</time_bank_accumulated_before>
      <time_bank_accumulated_after>2040</time_bank_accumulated_after>
      <time_bank_vacations_before>4800</time_bank_vacations_before>
      <time_bank_vacations_after>4800</time_bank_vacations_after>
```

```

    <time_bank_sick_before>0</time_bank_sick_before>
    <time_bank_sick_after>0</time_bank_sick_after>
  </time_bank_tx>
  <time_bank_tx>
    <time_bank_tx_id>569</time_bank_tx_id>
    <employee_id>29</employee_id>
    <employee_name><![CDATA[Janet Black]]></employee_name>
    <minutes_regular_time>2400</minutes_regular_time>
    <minutes_overtime>120</minutes_overtime>
    <minutes_vacations>0</minutes_vacations>
    <minutes_sick>0</minutes_sick>
    <minutes_ignored>0</minutes_ignored>
    <minutes_total>2520</minutes_total>
    <minutes_expected_work>2400</minutes_expected_work>
    <minutes_paid>2400</minutes_paid>
    <minutes_difference>120</minutes_difference>
    <time_bank_accumulated_before>0</time_bank_accumulated_before>
    <time_bank_accumulated_after>120</time_bank_accumulated_after>
    <time_bank_vacations_before>4800</time_bank_vacations_before>
    <time_bank_vacations_after>4800</time_bank_vacations_after>
    <time_bank_sick_before>0</time_bank_sick_before>
    <time_bank_sick_after>0</time_bank_sick_after>
  </time_bank_tx>
  <time_bank_tx>
    <time_bank_tx_id>570</time_bank_tx_id>
    <employee_id>30</employee_id>
    <employee_name><![CDATA[Mary Smith]]></employee_name>
    <minutes_regular_time>2400</minutes_regular_time>
    <minutes_overtime>240</minutes_overtime>
    <minutes_vacations>0</minutes_vacations>
    <minutes_sick>0</minutes_sick>
    <minutes_ignored>0</minutes_ignored>
    <minutes_total>2640</minutes_total>
    <minutes_expected_work>2400</minutes_expected_work>
    <minutes_paid>2400</minutes_paid>
    <minutes_difference>240</minutes_difference>
    <time_bank_accumulated_before>60</time_bank_accumulated_before>
    <time_bank_accumulated_after>300</time_bank_accumulated_after>
    <time_bank_vacations_before>4800</time_bank_vacations_before>
    <time_bank_vacations_after>4800</time_bank_vacations_after>
    <time_bank_sick_before>0</time_bank_sick_before>
    <time_bank_sick_after>0</time_bank_sick_after>
  </time_bank_tx>
</time_bank_update>
</response>

```

invoice_list (Get entire list of invoices in a summary format)

Once the list of invoices retrieved, you can use the **invoice_get** request type to get detailed information about a specific invoice using the **invoice_id** retrieved.

Optional filters

customer_id: integer – When specified, only invoices for the specified customer will be returned.

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <invoice_list>
    <customer_id>155</customer_id>
  </invoice_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <invoice>
    <invoice_id>254</invoice_id>
    <numero>132</numero>
    <customer_id>155</customer_id>
    <customer_name><![CDATA[YYZ Inc.]]></customer_name>
    <status>3</status>
    <date>2014-07-24</date>
  </invoice>
  <invoice>
    <invoice_id>271</invoice_id>
    <numero>149</numero>
    <customer_id>155</customer_id>
    <customer_name><![CDATA[YYZ Inc.]]></customer_name>
    <status>3</status>
    <date>2014-10-19</date>
  </invoice>
</response>
```

invoice_get (Get detailed information about an invoice)**XML request example:**

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <invoice_get>
    <invoice_id>254</invoice_id>
  </invoice_get>
</request>
```

invoice_id: integer – Required - Id of the invoice to retrieve. If the id specified does not exist, the request will fail.

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <invoice>
    <invoice_id>254</invoice_id>
    <numero>158</numero>
    <customer_id>155</customer_id>
    <customer_name><![CDATA[XYZ Inc.]]></customer_name>
    <customer_address><![CDATA[979 Avenue de Bourgogne
Suite 330
```

```
Quebec QC G1W 2L4
Canada]]></customer_address>
  <customer_phone><![CDATA[(418) 524-5066]]></customer_phone>
  <customer_mobile><![CDATA[]]></customer_mobile>
  <customer_fax><![CDATA[]]></customer_fax>
  <customer_contact><![CDATA[Matt Simms]]></customer_contact>
  <customer_email><![CDATA[matt@company-yyz-inc.com]]></customer_email>
  <status>3</status>
  <date>2014-07-24</date>
  <reference_customer><![CDATA[PO-4527]]></reference_customer>
  <reference_internal><![CDATA[]]></reference_internal>
  <payment_terms><![CDATA[NET 30 DAYS]]></payment_terms>
  <project_id>100272</project_id>
  <project_description><![CDATA[Project YYZ]]></project_description>
  <subtotal>1523.26</subtotal>
  <tax1_percent>5.0000</tax1_percent>
  <tax2_percent>7.0000</tax2_percent>
  <tax1_amount>76.16</tax1_amount>
  <tax2_amount>91.40</tax2_amount>
  <tax2_formula>0</tax2_formula>
  <inv_total>1690.82</inv_total>
  <invoice_line>
    <invoice_line_id>6226</invoice_line_id>
    <sequence>1</sequence>
    <sub_sequence>0</sub_sequence>
    <type>10</type>
    <description><![CDATA[Invoicable time]]></description>
    <quantity>1680.0000</quantity>
    <price>0.0000</price>
    <amount>1400.00</amount>
    <project_id>100272</project_id>
    <project_description><![CDATA[Project YYZ]]></project_description>
  </invoice_line>
  <invoice_line>
    <invoice_line_id>6227</invoice_line_id>
    <sequence>2</sequence>
    <sub_sequence>0</sub_sequence>
    <type>50</type>
    <description><![CDATA[Billable expenses]]></description>
    <quantity>0.0000</quantity>
    <price>0.0000</price>
    <amount>23.26</amount>
    <project_id>100272</project_id>
    <project_description><![CDATA[Project YYZ]]></project_description>
  </invoice_line>
  <invoice_line>
    <invoice_line_id>6228</invoice_line_id>
    <sequence>3</sequence>
    <sub_sequence>0</sub_sequence>
    <type>100</type>
    <description><![CDATA[SADASDAS]]></description>
    <quantity>2.0000</quantity>
    <price>50.0000</price>
    <amount>100.00</amount>
    <project_id>100272</project_id>
    <project_description><![CDATA[Project YYZ]]></project_description>
  </invoice_line>
```



```
</invoice>  
</response>
```

employee_category_list (Get entire list of employee categories)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">  
  <employee_category_list>  
  </employee_category_list>  
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">  
  <employee_category>  
    <category_id>0</category_id>  
    <description><![CDATA[Default category]]></description>  
  </employee_category>  
  <employee_category>  
    <category_id>1</category_id>  
    <description><![CDATA[Other Category]]></description>  
  </employee_category>  
</response>
```

activity_type_category_list (Get entire list of activity_type categories)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">  
  <activity_type_category_list>  
  </activity_type_category_list>  
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">  
  <activity_type_category>  
    <category_id>0</category_id>  
    <description><![CDATA[Default category]]></description>  
  </activity_type_category>  
  <activity_type_category>  
    <category_id>1</category_id>  
    <description><![CDATA[Other Category]]></description>  
  </activity_type_category>  
</response>
```

customer_category_list (Get entire list of customer categories)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <customer_category_list>
  </customer_category_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <customer_category>
    <category_id>0</category_id>
    <description><![CDATA[Default category]]></description>
  </customer_category>
  <customer_category>
    <category_id>1</category_id>
    <description><![CDATA[Other Category]]></description>
  </customer_category>
</response>
```

project_category_list (Get entire list of project categories)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <project_category_list>
  </project_category_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_category>
    <category_id>0</category_id>
    <description><![CDATA[Default category]]></description>
  </project_category>
  <project_category>
    <category_id>1</category_id>
    <description><![CDATA[Other Category]]></description>
  </project_category>
</response>
```

expense_type_list (Get entire list of expense types)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <expense_type_list>
```

```
</expense_type_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <project_category>
    <expense_type>
      <expense_type_id>16</expense_type_id>
      <description><![CDATA[Photocopies]]></description>
      <status>inactive</status>
      <unit_cost>0.2500</unit_cost>
      <unit_price>0.5000</unit_price>
      <billable_to_customer>yes</billable_to_customer>
      <refundable_to_employee>yes</refundable_to_employee>
    </expense_type>
    <expense_type>
      <expense_type_id>20</expense_type_id>
      <description><![CDATA[Travel]]></description>
      <status>active</status>
      <unit_cost>0.5100</unit_cost>
      <unit_price>0</unit_price>
      <billable_to_customer>no</billable_to_customer>
      <refundable_to_employee>yes</refundable_to_employee>
    </expense_type>
  </response>
```

fiscal_period_list (Get entire list of fiscal periods)**XML request example:**

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <fiscal_period_list>
  </fiscal_period_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <fiscal_period>
    <fiscal_period_id>70</fiscal_period_id>
    <description><![CDATA[2014 - July]]></description>
    <status>closed</status>
    <date_begin>2014-07-01</date_begin>
    <date_end>2014-07-31</date_end>
    <fiscal_year>2014</fiscal_year>
    <fiscal_period_number>7</fiscal_period_number>
  </fiscal_period>
  <fiscal_period>
    <fiscal_period_id>71</fiscal_period_id>
    <description><![CDATA[2014 - August]]></description>
    <status>open</status>
    <date_begin>2014-08-01</date_begin>
```

```
<date_end>2014-08-31</date_end>
<fiscal_year>2014</fiscal_year>
<fiscal_period_number>8</fiscal_period_number>
</fiscal_period>
</response>
```

tax_rate_list (Get entire list of tax rates)

XML request example:

```
<request app="orotimesheet" version="8" password="abcdef-123456">
  <tax_rate_list>
</tax_rate_list>
</request>
```

Response successful example:

```
<response app="orotimesheet" version="8" status="1">
  <tax_rate>
    <tax_rate_id>0</tax_rate_id>
    <description><![CDATA[No tax]]></description>
    <tax1_percent>0.0000</tax1_percent>
    <tax2_percent>0.0000</tax2_percent>
  </tax_rate>
  <tax_rate>
    <tax_rate_id>1</tax_rate_id>
    <description><![CDATA[Default tax rate]]></description>
    <tax1_percent>5.0000</tax1_percent>
    <tax2_percent>0.0000</tax2_percent>
  </tax_rate>
</response>
```

Appendix A. List of error codes

- 1001 Unable to connect to database.
 - 1002 Error retrieving data from database.
 - 1003 Error saving data to database.

 - 1101 Record does not exist.
 - 1102 Record duplicate error. A record with the supplied value already exists.
 - 1103 The API request you made returned too many results. Please use filters to narrow it down.
 - 1104 You cannot delete this item.
 - 1105 This operation is not allowed.

 - 1201 Field cannot be empty. Field value required.
 - 1202 No field to edit specified.
 - 1203 Invalid e-mail. If you provide an e-mail address, the e-mail address must be in a valid format.
 - 1204 Quantity cannot be 0.
 - 1205 Duration cannot be 0.
 - 1206 Invalid parameter received.

 - 1301 You must provide an activity_type_id or an employee_id.

 - 1401 The billing_mode specified is invalid. Accepted values are fixed_amount or per_hour.
 - 1402 The time_bank_type specified is invalid.
Accepted values are accumulated_time, vacations, sick or ignore.
 - 1403 The week state specified is invalid.
Accepted values are not_submitted, submitted, approved or rejected.

 - 2001 Invalid XML document received.
 - 2002 Invalid attribute value in element.

 - 2101 Maximum one request is allowed in the XML document.
 - 2102 No request was found in the XML document.
 - 2104 Invalid request type found in the XML document.

 - 3001 Invalid API password.
 - 3002 Either your account is disabled or the API option is disabled in your account.
 - 3003 The application is currently in maintenance mode. Please try again later.

 - 4001 This date, time or datetime is not in a valid format.
 - 4002 A numeric value was expected but another format was given..
-

Appendix B. Shift and invoice status descriptions.

Shift descriptions

These shift descriptions are defined by yourself in the Application configuration menu of OroTimesheet.

Here is a sample setup for these descriptions.

Shift description	Example	Id
Shift description #1	Day shift	1
Shift description #2	Evening shift	2
Shift description #3	Night shift	3
Shift description #4	Weekend	4
Shift description #5	Special	5
Shift description #6	Shift 6	6
Shift description #7	Shift 7	7
Shift description #8	Shift 8	8
Shift description #9	Shift 9	9
Shift description #10	Shift 10	10

Invoice status descriptions

These status descriptions are also defined in the Application configuration menu of OroTimesheet.

Here is a sample setup for these descriptions.

Status descriptions	Example	Id
Status #1	New	1
Status #2	Verified	2
Status #3	Printed	3
Status #4	Sent to customer	4
Status #5	Other	5

Copyright

OroTimesheet

Copyright© 1996-2017 OroLogic Inc., All rights reserved.

Warning: This software and its documentation are protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

OroTimesheet and OROLOGIC are trademarks of OroLogic Inc., 979 Avenue de Bourgogne, Suite 330, Quebec, (Quebec), Canada, G1W 2L4. The names of other products, services and societies mentioned are trademarks of their respective owners. The societies, names used in the examples are fictive. Association with any society, name, product or events existing is not desired and is not an insinuation.