Deltek

Deltek People Planner 4.2

Web API Guide

September 24, 2021

While Deltek has attempted to verify that the information in this document is accurate and complete, some typographical or technical errors may exist. The recipient of this document is solely responsible for all decisions relating to or use of the information provided herein.

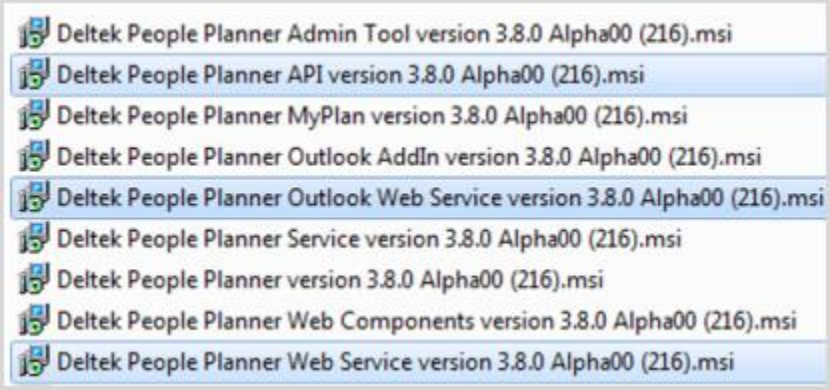
The information contained in this publication is effective as of the publication date below and is subject to change without notice.

This edition published September 2021.

Contents

People Planner API Web Services

The People Planner suite contains three web services, as highlighted in the following figure.



The following sections describe the Deltek People Planner API web service. This is the web service that Maconomy uses for its integration with People Planner. The People Planner API contains a set of SOAP web service methods to cover some very specific needs in different areas of the integration.

For instructions about how to install the Deltek People Planner API and to configure the connection to the database, see the *Deltek People Planner Technical Installation Guide*.

List of Web Service Methods

The list of web services in the People Planner API is organized by the general area of functionality. The web services are listed by:

- Test connection and setup for use of other web services
- Import jobs from Maconomy to People Planner
- Get estimates or planned hours for projects from People Planner
- Set status for projects or assignments in People Planner
- Set and get half-year budgets
- People Planner Silent Sign-In (SSI)

Test Connection and Setup for Use of Other Web Services

GetStatus

Call this method to verify that People Planner is running.

This method only calls the validation code in People Planner. It checks whether the People Planner API is running and is set up correctly—that is, it has a DataConnection.xml file with a correct server name, database name, and validation information for the database. It also checks whether the People Planner user and password that are used in the method call work.

Arguments

This method takes no arguments.

Reply

Returns a struct of type Response.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Code>200</Code>
    <Status>OK</Status>
    <Message>Connection status: Database</Message>
</Response>
```

Import Jobs from Maconomy to People Planner

ImportJobOrOpportunity

Call this method to request People Planner to import a job from Maconomy.

Deprecated: This method is only included for backward compatibility; you should use ImportJobOrOpportunityV2, instead. It calls through to ImportJobOrOpportunityV2 with the argument importSubjobsToo set to true.

Arguments

This method takes the following arguments:

- string externalID — A project number, Job number in Maconomy
- string projectType — Job or Opportunity
- string budgetType — The budget type, such as working budget, baseline budget; this is case-sensitive

Reply

Returns a struct of type Response (see GetStatus)

Example reply:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Code>200</Code>
    <Status>OK</Status>
    <Message>Status OK (Imported 1 events in total)</Message>
</Response>
```

ImportJobOrOpportunityV2

Call this method to request People Planner to import a job from Maconomy. This method is used when a user clicks the Send Job to People Planner button in the Maconomy Workspace Client.

This method calls in to Maconomy to make an import of the job or opportunity in Maconomy with the job number given as externalId and budget type given as budgetType.

The import from Maconomy will be done by using the MaconomyWS web service or by using the Maconomy RESTful web service. You configure this in the People Planner Admin Tool.

If projectType is **Job**, a method in Maconomy to return a job is called; if projectType is **Opportunity**, a method in Maconomy to return an opportunity is called.

If the flag for importSubjobsToo is true, sub-jobs for the job are imported as well.

Arguments

This method takes the following arguments:

- string externalID — A project number, Job number in Maconomy
- string projectType — Job or Opportunity
- string budgetType — The budget type, such as working budget, baseline budget; this is case-sensitive
- bool importSubjobsToo — If true, all sub-jobs are imported

Reply

Returns a struct of type Response (see GetStatus)

Example reply:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Code>200</Code>
    <Status>OK</Status>
    <Message>Status OK (Imported 1 events in total)</Message>
</Response>
```

GetAll

Call this method to force People Planner to import all jobs from Maconomy.

This method calls in to Maconomy to get a list of all jobs returned as job number and budget type and a list of all opportunities returned as opportunity number and budget type. This full list is then looped, and each job or opportunity is imported using the method from ImportJobOrOpportunityV2. Sub-jobs are always imported as well.

Arguments

This method takes no arguments.

Reply

Returns a struct of type Response (see GetStatus)

Example reply:

```
<?xml version="1.0" encoding="UTF-8"?>

<Response xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <Code>200</Code>

    <Status>OK</Status>

    <Message>Imported 695 events</Message>

</Response>
```

GetAllJobListItems

Call this method to get a full list of all jobs and opportunities from Maconomy.

This method calls in to Maconomy to get a list of all jobs returned as job number and budget type and a list of all opportunities returned as opportunity number and budget type. It returns the two lists that are concatenated.

Arguments

This method takes no arguments.

Reply

Returns an array of structs of type JobItem.

JobItem:

- string JobNumber — The project number (Job Number in Maconomy)
- bool IsOpportunity — A flag that indicates whether the information is about a job or an opportunity
- string BudgetType — The budget type, such as working budget, baseline budget; this is case-sensitive

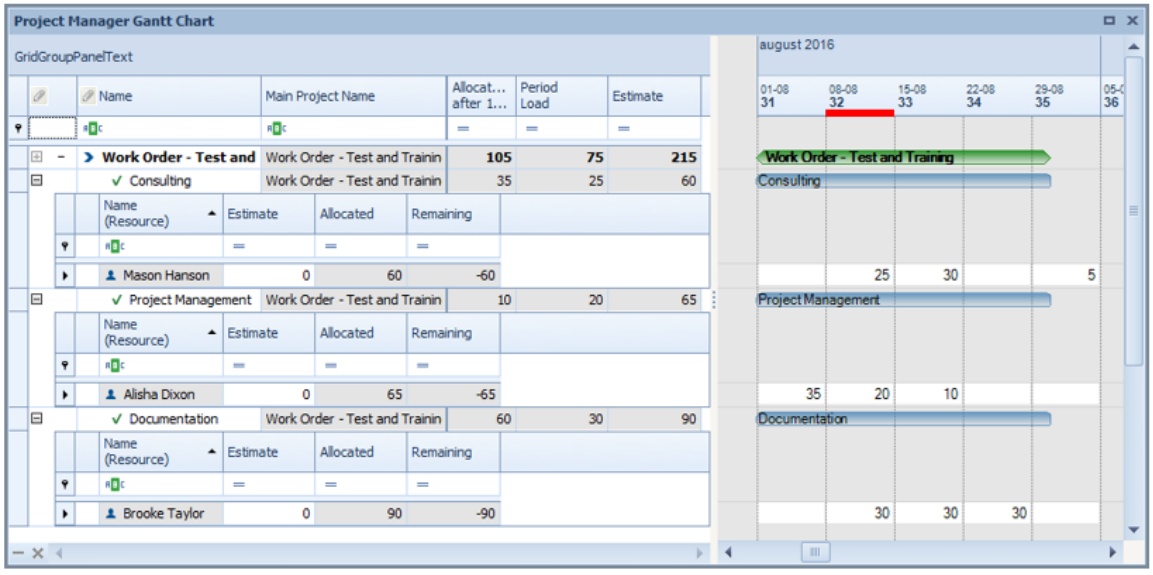
Example reply:

```
<?xml version="1.0" encoding="UTF-8"?>

<ArrayOfJobItem xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <JobItem>

        <JobNumber>1020001</JobNumber>

        <IsOpportunity>false</IsOpportunity>

        <BudgetType>Working Budget</BudgetType>

    </JobItem>

    <JobItem>

        <JobNumber>1020002</JobNumber>

        <IsOpportunity>false</IsOpportunity>

        <BudgetType>Working Budget</BudgetType>

    </JobItem>
```

```
    <JobItem>
        <JobNumber>1020003</JobNumber>
        <IsOpportunity>false</IsOpportunity>
        <BudgetType>Working Budget</BudgetType>
    </JobItem>
    <JobItem>
        <JobNumber>608002</JobNumber>
        <IsOpportunity>true</IsOpportunity>
        <BudgetType>Sales Estimate</BudgetType>
    </JobItem>
    <JobItem>
        <JobNumber>608003</JobNumber>
        <IsOpportunity>true</IsOpportunity>
        <BudgetType>Sales Estimate</BudgetType>
    </JobItem>
    <JobItem>
        <JobNumber>608004</JobNumber>
        <IsOpportunity>true</IsOpportunity>
        <BudgetType>Sales Estimate</BudgetType>
    </JobItem>
</ArrayOfJobItem>
```

Get Estimates or Planned Hours for Projects from People Planner

The following figure shows the project that is used as an example of the responses.



GetBudgetedHours

Call this method to get the summed-up bookings on a project.

This method finds information for all tasks and summary lines on the project (found by the project number and the budget type). For each task or summary line, a struct is returned with information to identify the event in People Planner and the budget line in Maconomy. The estimate for the event (working budget or quantity from Maconomy) and a calculation of the cost for the event are also returned.

The cost is calculated based on entries from the RegisteredCostByTime account. The RegisteredCostByTime account values are normally entered manually, but you can customize this. The cost calculation uses values from the RegisteredCostByTime account until the last registered entry. Values that occur after that date are based on planned hours (entries on the ManHoursAccount). The calculation method that is used depends on the method that you chose for each assignment or task. It can be based on:

- The cost of the resource.
- The resource category for the resource.
- Chosen for the task.
- A specific price list for the task or project.

Arguments

This method takes the following arguments:

- string projectNr — The project number (the Job Number in Maconomy)
- string budgetType — The type of budget, such as working budget, baseline budget; this is case-sensitive

Reply

Returns a struct of type WorkingBudget

WorkingBudget:

- string Status — A status for the call, for example OK, or an error with a brief explanation
- WorkingBudgetLine[] Lines — A list of struct of type WorkingBudgetLine, one for each budget line (task or summary line in People Planner)

WorkingBudgetLine:

- string Id — The People Planner Oid for the event
- string ExternalId — The first instance key from Maconomy to identify the budget line
- string Kind — The task, summary line, or milestone (for a milestone, the budget is always zero)
- string Name — The name of the event
- DateTime StartDate — The start of event
- DateTime EndDate — The finish of the event
- float BudgettedHours — The estimate of the event (quantity imported from Maconomy, can be overwritten in People Planner when using the functionality of copy of budget)
- float TotalCost — The sum of RegisteredCostByTimeEntry for as long a period as the event has any entries and cost based on planned hours (man hours entries) after that

If amount budget line importing is enabled, this method also returns data for amount line entries, but the Kind is set to Amount.

Example reply:

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkingBudget xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Status>OK</Status>
    <Lines>
        <WorkingBudgetLine>
            <Id>6a422d2a-95d9-4def-83d3-738b275dde47</Id>
            <ExternalId>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</ExternalId>
            <Kind>Task</Kind>
            <Name>Project Management</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <BudgettedHours>65</BudgettedHours>
            <TotalCost>5200</TotalCost>
        </WorkingBudgetLine>
        <WorkingBudgetLine>
            <Id>be9b5efb-2c89-488e-8fee-d46e7cdeabe4</Id>
            <ExternalId>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</ExternalId>
            <Kind>Task</Kind>
            <Name>Consulting</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <BudgettedHours>60</BudgettedHours>
            <TotalCost>4800</TotalCost>
        </WorkingBudgetLine>
        <WorkingBudgetLine>
            <Id>9ec04c03-82de-4e75-89c7-de596f31039e</Id>
            <ExternalId>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</ExternalId>
            <Kind>Task</Kind>
            <Name>Documentation</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <BudgettedHours>90</BudgettedHours>
            <TotalCost>4500</TotalCost>
        </WorkingBudgetLine>
    </Lines>
</WorkingBudget>
```

GetAllocatedHoursAfter

Call this method to get the summed-up bookings on a project after a specified date.

This method returns the number of allocated hours (man-hours entries) after the date provided as input for each task/summary line on the project.

Entries that start before the date and finish after the date are split based on the calendars for the resource, and the sum of the entries after the date is included.

Arguments

This method takes the following arguments:

- string projectNr — The project number, Job Number in Maconomy
- string budgetType — The budget type, such as working budget, baseline budget; this is case-sensitive
- DateTime startDate — This method returns allocations after this date (the date itself is excluded)

Reply

Returns a struct of type AllocatedHours.

AllocatedHours:

- string Status — A status for the call, such as OK, or an error with a brief explanation
- AllocatedHoursLine[] Lines — A list of struct of type AllocatedHoursLine, one for each budget line (task or summary line in People Planner)

AllocatedHoursLine:

- string Id —— The People Planner Oid for the event
- string ExternalId — The external ID for the event
- string Kind — The type of event (task, summary line or milestone)
- string Name — The name of the event
- DateTime StartDate — The start of the event
- DateTime EndDate — The end of the event
- float AllocatedHours — The sum of man-hours entries for a period (which period depends on the method)
- string TaskListId — The name of the Maconomy task referenced by the budget line
- AllocatedHoursResourceLine[] Resources — A list of structs of type AllocatedHoursResourceLine, one for each resource assigned to the event

AllocatedHoursResourceLine:

AllocatedHoursResourceLine is used in GetAllocatedHoursAfterWithResources and GetAllocatedHoursInPeriodWithResources. For the GetAllocatedHoursAfter method, AllocatedHoursResourceLine is always empty.

- float AllocatedHours —The sum of man-hours entries for this resource on one task
- string Id — The People Planner Oid for the resource

- string ExternalId — The external ID for the resource
- string Kind — The kind of the resource, such as individual, budget
- string Name — The name of the resource

If amount budget line importing is enabled, this method also returns data for amount line entries, but the Kind is set to Amount.

Example reply (asking for hours after August 14, 2016):

```
<?xml version="1.0" encoding="UTF-8"?>
<AllocatedHours xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Status>OK</Status>
   <Lines>
      <AllocatedHoursLine>
         <Id>6a422d2a-95d9-4def-83d3-738b275dde47</Id>
         <ExternalId>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</ExternalId>
         <Kind>Task</Kind>
         <Name>Project Management</Name>
         <StartDate>2016-08-01T00:00:00</StartDate>
         <EndDate>2016-08-30T00:00:00</EndDate>
         <AllocatedHours>10</AllocatedHours>
         <TaskListId>110</TaskListId>
      </AllocatedHoursLine>
      <AllocatedHoursLine>
         <Id>be9b5efb-2c89-488e-8fee-d46e7cdeabe4</Id>
         <ExternalId>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</ExternalId>
         <Kind>Task</Kind>
         <Name>Consulting</Name>
         <StartDate>2016-08-01T00:00:00</StartDate>
         <EndDate>2016-08-30T00:00:00</EndDate>
         <AllocatedHours>35</AllocatedHours>
         <TaskListId>100</TaskListId>
      </AllocatedHoursLine>
      <AllocatedHoursLine>
         <Id>9ec04c03-82de-4e75-89c7-de596f31039e</Id>
         <ExternalId>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</ExternalId>
         <Kind>Task</Kind>
         <Name>Documentation</Name>
         <StartDate>2016-08-01T00:00:00</StartDate>
         <EndDate>2016-08-30T00:00:00</EndDate>
         <AllocatedHours>60</AllocatedHours>
         <TaskListId>120</TaskListId>
      </AllocatedHoursLine>
   </Lines>
```

```
</AllocatedHours>
```

GetAllocatedHoursInPeriod

Call this method to get the summed-up bookings on a project within the specified date interval.

This method finds the number of allocated hours (man-hours entries) after the start date provided as input for each task/summary line on the project and up to and including the end date.

All entries that start before the start date and finish after the start date are divided based on the calendars for the resource; the parts of the entries that occur after the start date are included. Similarly, the entries that start before the end date and finish after the end date are also divided based on the calendars, and the parts that occur before the end date are included.

The AllocatedHoursResourceLine is not used in this method; it is used only for GetAllocatedHoursAfterWithResources and GetAllocatedHoursInPeriodWithResources.

Arguments

This method takes the following arguments:

- string projectNr — The project number, Job number in Maconomy
- string budgetType — The budget type, such as working budget, baseline budget; this is case-sensitive
- DateTime startDate — This method returns allocations after and on this date
- DateTime endDate — This method returns allocations before or on this date

Reply

Returns a struct of type AllocatedHours (see GetAllocatedHoursAfter)

If amount budget line importing is enabled, this method also returns data for amount line entries, but the Kind is set to Amount.

Example reply (requesting hours from August 8 to August 14, 2016):

```
<?xml version="1.0" encoding="UTF-8"?>
<AllocatedHours xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Status>OK</Status>
    <Lines>
        <AllocatedHoursLine>
            <Id>6a422d2a-95d9-4def-83d3-738b275dde47</Id>
            <ExternalId>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</ExternalId>
            <Kind>Task</Kind>
            <Name>Project Management</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>20</AllocatedHours>
            <TaskListId>110</TaskListId>
        </AllocatedHoursLine>
        <AllocatedHoursLine>
            <Id>be9b5efb-2c89-488e-8fee-d46e7cdeabe4</Id>
```

```
            <ExternalId>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</ExternalId>
            <Kind>Task</Kind>
            <Name>Consulting</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>25</AllocatedHours>
            <TaskListId>100</TaskListId>
        </AllocatedHoursLine>
        <AllocatedHoursLine>
            <Id>9ec04c03-82de-4e75-89c7-de596f31039e</Id>
            <ExternalId>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</ExternalId>
            <Kind>Task</Kind>
            <Name>Documentation</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>30</AllocatedHours>
            <TaskListId>120</TaskListId>
        </AllocatedHoursLine>
    </Lines>
</AllocatedHours>
```

GetAllocatedHoursAfterWithResources

Call this method to get the summed-up bookings on a project after the specified date. The numbers are further broken down on individual resources.

This method performs the same calculations as GetAllocatedHoursAfter. The only difference is that for each event/budget line this method also returns a list of structs of the type AllocatedHoursResourceLine, one for each resource assigned to the event.

This struct contains information that identifies the assigned resource in both People Planner and Maconomy and the number of hours that this resource is allocated on the specific event after the date provided as input to the method (entries are also divided to perform the calculation as described for the GetAllocatedHoursAfter method).

Arguments

This method takes the following arguments:

- string projectNr — The project number, Job number in Maconomy
- string budgetType — The budget type, such as working budget, baseline budget; this is case-sensitive
- DateTime startDate — This method returns allocations after this date (the date itself is excluded)

Reply

Returns a struct of type AllocatedHours (see GetAllocatedHoursAfter)

If amount budget line importing is enabled, this method also returns data for amount line entries, but the Kind is set to Amount.

Example reply (requesting hours after August 14, 2016):

```
<?xml version="1.0" encoding="UTF-8"?>
<AllocatedHours xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Status>OK</Status>
    <Lines>
        <AllocatedHoursLine>
            <Id>6a422d2a-95d9-4def-83d3-738b275dde47</Id>
            <ExternalId>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</ExternalId>
            <Kind>Task</Kind>
            <Name>Project Management</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>10</AllocatedHours>
            <TaskListId>110</TaskListId>
            <Resources>
                <AllocatedHoursResourceLine>
                    <AllocatedHours>10</AllocatedHours>
                    <Id>c5c8f0ea-090d-46ed-9cfd-3616f9906e1c</Id>
                    <ExternalId>2064</ExternalId>
                    <Kind>Individual</Kind>
                    <Name>Alisha Dixon</Name>
                </AllocatedHoursResourceLine>
            </Resources>
        </AllocatedHoursLine>
        <AllocatedHoursLine>
            <Id>be9b5efb-2c89-488e-8fee-d46e7cdeabe4</Id>
            <ExternalId>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</ExternalId>
            <Kind>Task</Kind>
            <Name>Consulting</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>35</AllocatedHours>
            <TaskListId>100</TaskListId>
            <Resources>
                <AllocatedHoursResourceLine>
                    <AllocatedHours>35</AllocatedHours>
                    <Id>f197f94b-c3aa-4231-befa-239ee04e515f</Id>
                    <ExternalId>2032</ExternalId>
                    <Kind>Individual</Kind>
                    <Name>Mason Hanson</Name>
```

```
            </AllocatedHoursResourceLine>
         </Resources>
      </AllocatedHoursLine>
      <AllocatedHoursLine>
         <Id>9ec04c03-82de-4e75-89c7-de596f31039e</Id>
         <ExternalId>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</ExternalId>
         <Kind>Task</Kind>
         <Name>Documentation</Name>
         <StartDate>2016-08-01T00:00:00</StartDate>
         <EndDate>2016-08-30T00:00:00</EndDate>
         <AllocatedHours>60</AllocatedHours>
         <TaskListId>120</TaskListId>
         <Resources>
            <AllocatedHoursResourceLine>
               <AllocatedHours>60</AllocatedHours>
               <Id>199ea9e9-baa5-4b54-ad29-7d282eda3a65</Id>
               <ExternalId>3038</ExternalId>
               <Kind>Individual</Kind>
               <Name>Brooke Taylor</Name>
            </AllocatedHoursResourceLine>
         </Resources>
      </AllocatedHoursLine>
   </Lines>
</AllocatedHours>
```

GetAllocatedHoursInPeriodWithResources

Call this method to get the summed-up bookings on a project in the specified date interval. The numbers are further broken down on the individual resources.

This method performs the same calculations as GetAllocatedHoursInPeriod. The only difference is that for each event/budget line this method also returns a list of struct of type AllocatedHoursResourceLine, one for each resource assigned to the event.

This struct contains information that identifies the assigned resource in both People Planner and Maconomy and the number of hours that this resource is allocated on the specific event after the start date provided as input to the method and before the end date provided as input (entries are also divided to perform the calculation as described for the GetAllocatedHoursInPeriod method).

Arguments

This method takes the following arguments:

- string projectNr — The project number, Job number in Maconomy
- string budgetType — The budget type, such as working budget, baseline budget; this is case-sensitive
- DateTime startDate — This method returns allocations after and on this date

- DateTime endDate — This method returns allocations before or on this date

Reply

Returns a struct of type AllocatedHours (see GetAllocatedHoursAfter).

If amount budget line importing is enabled, this method also returns data for amount line entries, but the Kind is set to Amount.

Example reply (requesting hours from August 8 to August 14, 2016):

```
<?xml version="1.0" encoding="UTF-8"?>
<AllocatedHours xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Status>OK</Status>
    <Lines>
        <AllocatedHoursLine>
            <Id>6a422d2a-95d9-4def-83d3-738b275dde47</Id>
            <ExternalId>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</ExternalId>
            <Kind>Task</Kind>
            <Name>Project Management</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>20</AllocatedHours>
            <TaskListId>110</TaskListId>
            <Resources>
                <AllocatedHoursResourceLine>
                    <AllocatedHours>20</AllocatedHours>
                    <Id>c5c8f0ea-090d-46ed-9cfd-3616f9906e1c</Id>
                    <ExternalId>2064</ExternalId>
                    <Kind>Individual</Kind>
                    <Name>Alisha Dixon</Name>
                </AllocatedHoursResourceLine>
            </Resources>
        </AllocatedHoursLine>
        <AllocatedHoursLine>
            <Id>be9b5efb-2c89-488e-8fee-d46e7cdeabe4</Id>
            <ExternalId>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</ExternalId>
            <Kind>Task</Kind>
            <Name>Consulting</Name>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <EndDate>2016-08-30T00:00:00</EndDate>
            <AllocatedHours>25</AllocatedHours>
            <TaskListId>100</TaskListId>
            <Resources>
                <AllocatedHoursResourceLine>
```

```
                <AllocatedHours>25</AllocatedHours>
                <Id>f197f94b-c3aa-4231-befa-239ee04e515f</Id>
                <ExternalId>2032</ExternalId>
                <Kind>Individual</Kind>
                <Name>Mason Hanson</Name>
            </AllocatedHoursResourceLine>
        </Resources>
    </AllocatedHoursLine>
    <AllocatedHoursLine>
        <Id>9ec04c03-82de-4e75-89c7-de596f31039e</Id>
        <ExternalId>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</ExternalId>
        <Kind>Task</Kind>
        <Name>Documentation</Name>
        <StartDate>2016-08-01T00:00:00</StartDate>
        <EndDate>2016-08-30T00:00:00</EndDate>
        <AllocatedHours>30</AllocatedHours>
        <TaskListId>120</TaskListId>
        <Resources>
            <AllocatedHoursResourceLine>
                <AllocatedHours>30</AllocatedHours>
                <Id>199ea9e9-baa5-4b54-ad29-7d282eda3a65</Id>
                <ExternalId>3038</ExternalId>
                <Kind>Individual</Kind>
                <Name>Brooke Taylor</Name>
            </AllocatedHoursResourceLine>
        </Resources>
    </AllocatedHoursLine>
  </Lines>
</AllocatedHours>
```

GetPeriodicBudget

Call this method to get the summed-up bookings on a project. The hours are drilled down in periods of individual days, weeks, or months, depending on the specified periodType.

Arguments

This method takes the following arguments:

- string projectNumber — The project number, Job Project in Maconomy
- DateTime startDate — The starting date for getting the budget.
- string periodType — The granularity of the period budget (day, week, or month; this is not case-sensitive)

- int numberOfPeriods — The number of periods to get the budget for (1 for all periods for the project)

Reply

Returns a struct of type PeriodicCost

PeriodicCost:

- string JobNumber — A project number
- List<PeriodCostLine> PeriodCostLines — A list of structs of type PeriodCostLines, one for each task on the project that originally came from Maconomy

PeriodCostLines:

- string InstanceKey — The first instance key to be able to match the task to a budget line in Maconomy
- DateTime AllocationStartDate — The start date for the first period with planned hours for this task after the start date provided as input to the method
- List<double> PeriodHours — The sum of planned hours for each period on the task
- double BeforePeriodBookedHours — The sum of all planned hours on the task before the allocation start date
- double AfterPeriodBookedHours — The sum of all planned hours on the task after the last period in the period hours list
- double RemainingHours — The difference between the estimate for the task and the number of booked hours
- DateTime StartDate — The start of the task
- DateTime FinishDate — The finish of the task

The start date provided as input is snapped to the first valid start date depending on the chosen granularity (next midnight for day, next midnight to a Monday for week, and next midnight to the first in a month for month). If the start date is given as a null date (01-01-0001) the result is as if the start date were set to a valid start date just before the project start.

The number of periods is normally the number of periods for which the budget is needed. If the number of periods is -1, the method returns all periods from the duration of the project.

All bookings from before the startDate are summarized as BeforePeriodBookedHours. If any planned hours start before the start date and end after the start date, they are divided based on the calendars for the resource. The sum is returned in the BeforePeriodBookedHours attribute for each task.

For each period after the start date the planned hours are also summed and, just as for planned hours before the start, the planned hours are also divided, based on the calendars if they cover more periods. The AllocationStartDate is the same as the snapped start date by default. If the first periods do not contain any planned hours, the AllocationStartDate is set to the start of the first period with planned hours, and the list of allocated hours in the period does not contain any entry for the first periods. If the last periods for a task contain zeros, they are also removed from the list of allocated hours.

All planned hours after the last period and until the end of the project are summed up in the AfterPeriodBookedHours.

Even if amount budget line importing is enabled this method will *not* return data for amount line entries.

Example reply (requesting data after August 1, 2016, on the week level and all periods):

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<PeriodicCost xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <JobNumber>1120006</JobNumber>
    <PeriodCostLines>
        <PeriodCostLine>
            <InstanceKey>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</InstanceKey>
            <AllocationStartDate>2016-08-01T00:00:00</AllocationStartDate>
            <PeriodHours>
                <double>35</double>
                <double>20</double>
                <double>10</double>
            </PeriodHours>
            <BeforePeriodBookedHours>0</BeforePeriodBookedHours>
            <AfterPeriodBookedHours>0</AfterPeriodBookedHours>
            <RemainingHours>0</RemainingHours>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <FinishDate>2016-08-30T23:59:59</FinishDate>
        </PeriodCostLine>
        <PeriodCostLine>
            <InstanceKey>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</InstanceKey>
            <AllocationStartDate>2016-08-08T00:00:00</AllocationStartDate>
            <PeriodHours>
                <double>25</double>
                <double>30</double>
                <double>0</double>
                <double>5</double>
            </PeriodHours>
            <BeforePeriodBookedHours>0</BeforePeriodBookedHours>
            <AfterPeriodBookedHours>0</AfterPeriodBookedHours>
            <RemainingHours>0</RemainingHours>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <FinishDate>2016-08-30T23:59:59</FinishDate>
        </PeriodCostLine>
        <PeriodCostLine>
            <InstanceKey>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</InstanceKey>
            <AllocationStartDate>2016-08-08T00:00:00</AllocationStartDate>
            <PeriodHours>
                <double>30</double>
                <double>30</double>
                <double>30</double>
            </PeriodHours>
            <BeforePeriodBookedHours>0</BeforePeriodBookedHours>
```

```
            <AfterPeriodBookedHours>0</AfterPeriodBookedHours>
            <RemainingHours>0</RemainingHours>
            <StartDate>2016-08-01T00:00:00</StartDate>
            <FinishDate>2016-08-30T23:59:59</FinishDate>
        </PeriodCostLine>
    </PeriodCostLines>
</PeriodicCost>
```

GetPeriodicAssignments

This method returns all tasks and all assignments on a project. The allocated hours and amounts on assignments are split into a number of periods specified by *periodType* and *numberOfPeriods*.

In the context of this section, the term *budget line* refers to a budget line in Maconomy that is a *task* or an *amount line* in People Planner.

Arguments

This method takes the following arguments:

- string projectNumber — The project number, Job Project in Maconomy
- DateTime startDate — The starting date for getting the budget
- string periodType — The granularity of the period budget (day, week, or month; this is not case-sensitive)
- int numberOfPeriods — The number of periods to get the budget for (1 for all periods for the project)

Reply

Returns a struct of type *GetPeriodicAssignmentsResult*

GetPeriodicAssignmentsResult:

- string JobNumber — A project number
- List<Line> Lines — An array of *Line* data structures, one for each budget line that originally came from Maconomy

Line:

- string InstanceKey — The first instance key of the budget line in Maconomy
- string LineType Kind — The type of budget line (event kind in People Planner)
- DateTime StartDate — The start date of the budget line
- DateTime FinishDate — The finish date of the budget line
- List<Assignment> Assignments — An array of *Assignment* data structures on the budget line

Assignment:

- string EmployeeNumber — The Maconomy employee number of the resource assigned to the budget line (PP resource external ID in People Planner)

- DateTime AllocationStartDate — The start date for the first period for this assignment
- List<double> PeriodValues — A consecutive array of planned values (hours, amount) for this assignment. Each value is the planned time or amount for the periods starting from the *AllocationStartDate*.

The start date that is provided as input is snapped to the first valid start date, depending on the chosen granularity (next midnight for day, next midnight to a Monday for week, and next midnight to the first in a month for month). If the start date is provided as a null date (01-01-0001) the result is as if the start date were set to a valid start date just before the project start.

The number of periods is normally the number of periods for which the budget is needed. If the number of periods is -1, the method returns all periods from the duration of the project.

For time allocations:

For each period after the start date the planned hours are distributed based on the calendars, if they cover more periods. The AllocationStartDate is the same as the snapped start date by default. If the first periods do not contain any planned hours, the AllocationStartDate is set to the start of the first period with planned hours, and the list of allocated hours in the period does not contain any entry for the first periods. If the last periods for a task contain zeros, they are also removed from the list of allocated hours.

Unlike with GetPeriodicBudget, all planned hours before and after the period are ***not*** included in the response.

For amount allocations:

This is the same as for time allocations, except that the distributed values are amounts, and allocation is *not* based on the calendars.

Example reply (requesting all data, on month level, all periods):

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
   <soap:Header>
      <wsa:Action>http://marstrand-
innovation.com/GetPeriodicAssignmentsResponse</wsa:Action>
      <wsa:MessageID>urn:uuid:685fa3f0-1e4c-4537-b0e9-1cfc4ceb3463</wsa:MessageID>
      <wsa:RelatesTo>uuid:c803e624-47b3-48bf-8890-0854715c064c</wsa:RelatesTo>
      <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
      <wsse:Security>
         <wsu:Timestamp wsu:Id="Timestamp-64558d9c-0d7d-4a56-aa38-667f1a227d8e">
            <wsu:Created>2018-05-08T09:03:02Z</wsu:Created>
            <wsu:Expires>2018-05-08T09:08:02Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </soap:Header>
   <soap:Body>
```

```
<GetPeriodicAssignmentsResponse xmlns="http://marstrand-innovation.com/">
   <GetPeriodicAssignmentsResult>
      <JobNumber>1020005</JobNumber>
      <Lines>
         <Line>
            <InstanceKey>JobBudgetLineD6072AC7-A379-4681-A6A5-
8809FB6F2E2B</InstanceKey>
            <LineType>Time</LineType>
            <StartDate>2017-12-01T00:00:00</StartDate>
            <FinishDate>2018-03-31T23:59:59</FinishDate>
            <Assignments>
               <Assignment>
                  <EmployeeNumber>3037</EmployeeNumber>
                  <AllocationStartDate>2017-12-01T00:00:00</AllocationStartDate>
                  <PeriodValues>
                     <double>100</double>
                     <double>120</double>
                     <double>0</double>
                     <double>50</double>
                  </PeriodValues>
               </Assignment>
            </Assignments>
         </Line>
         <Line>
            <InstanceKey>JobBudgetLine153816C2-1BF1-47AD-B445-
9651E8811185</InstanceKey>
            <LineType>Time</LineType>
            <StartDate>2017-12-01T00:00:00</StartDate>
            <FinishDate>2018-01-31T23:59:59</FinishDate>
            <Assignments>
               <Assignment>
                  <EmployeeNumber>1138</EmployeeNumber>
                  <AllocationStartDate>2017-12-01T00:00:00</AllocationStartDate>
                  <PeriodValues>
                     <double>20</double>
                     <double>30</double>
                  </PeriodValues>
               </Assignment>
               <Assignment>
```

```
                        <EmployeeNumber>3038</EmployeeNumber>
                        <AllocationStartDate>2018-01-01T00:00:00</AllocationStartDate>
                        <PeriodValues>
                           <double>40</double>
                        </PeriodValues>
                     </Assignment>
                  </Assignments>
               </Line>
               <Line>
                  <InstanceKey>JobBudgetLine286BEC8E-77C2-4A82-8903-
A8CB25A3BBB0</InstanceKey>
                  <LineType>Time</LineType>
                  <StartDate>2017-11-01T00:00:00</StartDate>
                  <FinishDate>2018-04-30T23:59:59</FinishDate>
                  <Assignments>
                     <Assignment>
                        <EmployeeNumber>2064</EmployeeNumber>
                        <AllocationStartDate>2018-04-01T00:00:00</AllocationStartDate>
                        <PeriodValues>
                           <double>1</double>
                        </PeriodValues>
                     </Assignment>
                  </Assignments>
               </Line>
               <Line>
                  <InstanceKey>JobBudgetLine933EB6DA-5E97-4D13-AEBD-
92FCF8C2B054</InstanceKey>
                  <LineType>Time</LineType>
                  <StartDate>2017-11-01T00:00:00</StartDate>
                  <FinishDate>2018-04-30T23:59:59</FinishDate>
                  <Assignments>
                     <Assignment>
                        <EmployeeNumber/>
                        <AllocationStartDate>2018-04-01T00:00:00</AllocationStartDate>
                        <PeriodValues>
                           <double>1</double>
                        </PeriodValues>
                     </Assignment>
                  </Assignments>
```

```
            </Line>
            <Line>
               <InstanceKey>JobBudgetLine_AmountTestKey</InstanceKey>
               <LineType>Amount</LineType>
               <StartDate>2018-05-07T00:00:00</StartDate>
               <FinishDate>2018-05-21T00:59:59</FinishDate>
               <Assignments>
                  <Assignment>
                     <EmployeeNumber/>
                     <AllocationStartDate>2018-05-01T00:00:00</AllocationStartDate>
                     <PeriodValues>
                        <double>60</double>
                     </PeriodValues>
                  </Assignment>
               </Assignments>
            </Line>
            <Line>
               <InstanceKey>JobBudgetLineD1DC1A6B-881B-4D6E-AEF6-
C089FDCB2E1F</InstanceKey>
               <LineType>Time</LineType>
               <StartDate>2017-12-01T00:00:00</StartDate>
               <FinishDate>2018-01-31T23:59:59</FinishDate>
               <Assignments>
                  <Assignment>
                     <EmployeeNumber>2064</EmployeeNumber>
                     <AllocationStartDate>2018-06-01T00:00:00</AllocationStartDate>
                     <PeriodValues/>
                  </Assignment>
               </Assignments>
            </Line>
            <Line>
               <InstanceKey>JobBudgetLine32997C6E-981E-40C0-8F0C-
26DA6EA0663A</InstanceKey>
               <LineType>Time</LineType>
               <StartDate>2017-12-01T00:00:00</StartDate>
               <FinishDate>2018-01-31T23:59:59</FinishDate>
               <Assignments>
                  <Assignment>
                     <EmployeeNumber>1138</EmployeeNumber>
```

```
                        <AllocationStartDate>2018-06-01T00:00:00</AllocationStartDate>
                        <PeriodValues/>
                     </Assignment>
                  </Assignments>
               </Line>
               <Line>
                  <InstanceKey>JobBudgetLine5574A218-00BE-47B3-834C-
B101714AF962</InstanceKey>
                  <LineType>Time</LineType>
                  <StartDate>2017-11-01T00:00:00</StartDate>
                  <FinishDate>2018-04-30T23:59:59</FinishDate>
                  <Assignments/>
               </Line>
               <Line>
                  <InstanceKey>JobBudgetLineD9AFC304-6016-44DD-9648-
E11C6B4D5DF4</InstanceKey>
                  <LineType>Time</LineType>
                  <StartDate>2017-11-01T00:00:00</StartDate>
                  <FinishDate>2018-04-30T23:59:59</FinishDate>
                  <Assignments/>
               </Line>
            </Lines>
         </GetPeriodicAssignmentsResult>
      </GetPeriodicAssignmentsResponse>
   </soap:Body>
</soap:Envelope>
```

Set Status for Projects or Assignments in People Planner

SetProjectStatus

Call this method to set the status of a project in People Planner.

This method locates the project based on the project number. The project status is then changed to the new project status. If the project status does not exist in the People Planner database, an exception occurs.

Arguments

The method takes the following arguments:

- string projectNr — The project number, Job number in Maconomy
- string projectStatusName — The name of a project status (this should match a name of a status in the People Planner database)

Reply

Returns nothing

Set and Get Half-Year Budgets

MaconomySetHalfYearBudget

Call this method to create bookings in People Planner for a half-year period. The bookings are created as bookings on the monthly level.

Arguments

This method takes the following argument:

- MaconomyHalfYearBudget budget

MaconomyHalfYearBudget:

- DateTime StartDate — The start for when the half-year budget should be inserted
- string ProjectNumber — The project number
- List<MaconomyBudgetLine> BudgetLines — The list of structs of type MaconomyBudgetLine, one for each budget line (works for a classic budget in Maconomy, where each budget line references one and only one employee)

MaconomyBudgetLine:

- string EmployeeNumber — The employee number for a resource
- string MaconomyID — The external ID for an event (first instance key from Maconomy)
- string Name — The name of the budget line in Maconomy
- double Month1PlannedHours — The number of hours for the first month
- double Month2PlannedHours — The number of hours for the second month
- double Month3PlannedHours — The number of hours for the third month
- double Month4PlannedHours — The number of hours for the fourth month
- double Month5PlannedHours — The number of hours for the fifth month
- double Month6PlannedHours — The number of hours for the sixth month
- double FuturePlannedHours — The number of hours for the remaining time on the budget line

Reply

Returns nothing

The input to this method is validated. The start date that is provided as input must be midnight on a first of the month. An exception occurs if the date does not match this. All employee numbers referenced by the budget lines must correspond to resources in People Planner.

All of the budget lines given in the input are found as tasks in People Planner, and all bookings (man-hour entries) from the date provided as input and forward are deleted. All bookings that start before the date and finish after the date are divided using calendar information for the resource, and the part after the date is deleted.

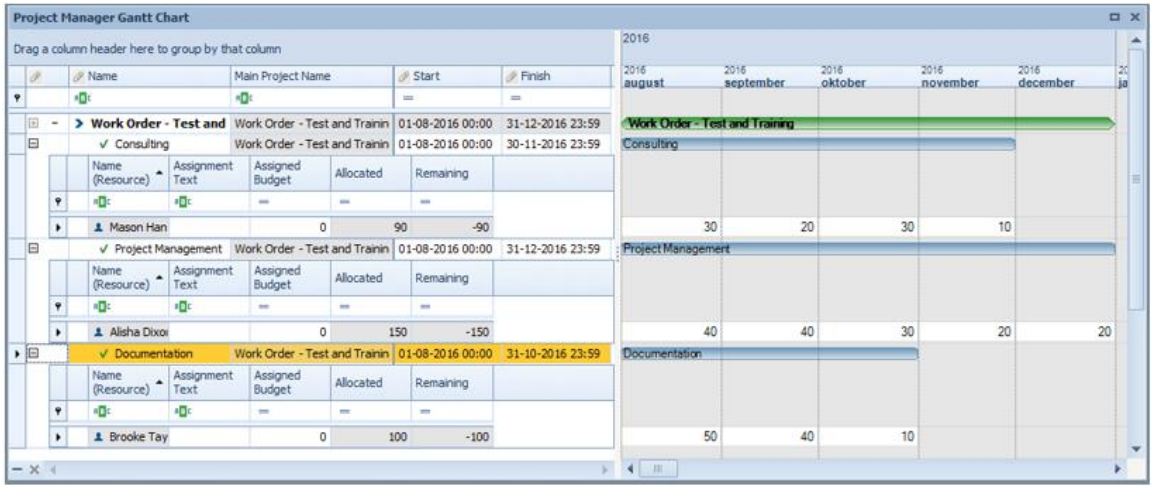
All tasks in People Planner that correspond to the budget lines are checked to verify that they cover the period for which they get new values (not zeroes). If a task gets values for all six periods and after, it must start on or before the date given in the import and finish more than six months after the date. If only some of the periods contain values, the task only needs to cover those periods. If a task does not cover the needed period, it is extended to cover that period.

When the tasks have the right duration, bookings are created for each of the periods for a task that contains a value (not zero).

Example input, which results in the bookings seen in the image following the code example (the view is changed to show the project on the month level):

```
<MaconomySetHalfYearBudget xmlns="http://marstrand-innovation.com/">
    <budget>
        <BudgetLines>
            <MaconomyBudgetLine>
                <EmployeeNumber>2064</EmployeeNumber>
                <MaconomyID> JobBudgetLine87C022EA-D989-4E08-93ED-
                23DBAA7E8049</MaconomyID>
                <Name> Project Management </Name>
                <Month1PlannedHours>40</Month1PlannedHours>
                <Month2PlannedHours>40</Month2PlannedHours>
                <Month3PlannedHours>30</Month3PlannedHours>
                <Month4PlannedHours>20</Month4PlannedHours>
                <Month5PlannedHours>20</Month5PlannedHours>
                <Month6PlannedHours>0</Month6PlannedHours>
                <FuturePlannedHours>0</FuturePlannedHours>
            </MaconomyBudgetLine>
            <MaconomyBudgetLine>
                <EmployeeNumber>2032</EmployeeNumber>
                <MaconomyID> JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D
                </MaconomyID>
                <Name> Consulting </Name>
                <Month1PlannedHours>30</Month1PlannedHours>
                <Month2PlannedHours>20</Month2PlannedHours>
                <Month3PlannedHours>30</Month3PlannedHours>
                <Month4PlannedHours>10</Month4PlannedHours>
                <Month5PlannedHours>0</Month5PlannedHours>
                <Month6PlannedHours>0</Month6PlannedHours>
                <FuturePlannedHours>0</FuturePlannedHours>
            </MaconomyBudgetLine>
            <MaconomyBudgetLine>
                <EmployeeNumber>3038</EmployeeNumber>
                <MaconomyID> JobBudgetLineE3871BE6-A7FB-4784-A7AB-
                828C71F8EDA4</MaconomyID>
```

```
            <Name> Documentation </Name>
            <Month1PlannedHours>50</Month1PlannedHours>
            <Month2PlannedHours>40</Month2PlannedHours>
            <Month3PlannedHours>10</Month3PlannedHours>
            <Month4PlannedHours>0</Month4PlannedHours>
            <Month5PlannedHours>0</Month5PlannedHours>
            <Month6PlannedHours>0</Month6PlannedHours>
            <FuturePlannedHours>0</FuturePlannedHours>
        </MaconomyBudgetLine>
    </BudgetLines>
</budget>
</MaconomySetHalfYearBudget>
```



GetHalfYearBudget

Call this method to get the bookings for a half-year period. The bookings are summed up on the monthly level.

Arguments

This method takes the following arguments:

- string projectNumber — The project number, Job Number in Maconomy
- DateTIme startDate — The date for which part of the budget in People Planner should be returned

Reply

Returns a struct of type PeoplePlannerHalfYearBudget

PeoplePlannerHalfYearBudget:

- DateTime StartDate — The planned hours after this date are returned
- string ProjectNumber — The Project number

- List<PeoplePlannerBudgetLine> BudgetLines — A list of structs of type PeoplePlannerBudgetLine, one for each task on the project originally imported from Maconomy

PeoplePlannerBudgetLine:

- string MaconomyID — The first instance key for the budget line in Maconomy
- string Name — The name of the task in People Planner
- List<Employee> Employees — The list of structs of type Employee, one for each resource assigned to the task
- string PeoplePlannerID — The Oid for the task in People Planner

Employee:

- string EmployeeNumber — The employee number for the employee in Maconomy
- string PeoplePlannerId — The Oid for the resource in People Planner
- double Month1PlannedHours — The hours planned in People Planner first month after start
- double Month2PlannedHours — The hours planned in People Planner second month after start
- double Month3PlannedHours — The hours planned in People Planner third month after start
- double Month4PlannedHours — The hours planned in People Planner fourth month after start
- double Month5PlannedHours — The hours planned in People Planner fifth month after start
- double Month6PlannedHours — The hours planned in People Planner sixth month after start
- double FuturePlannedHours — The hours planned more than six months after start

The input to this method is validated. The start date must be midnight on a first of a month; an exception is thrown if the date does not match this. The project number must match a project in People Planner that originally was imported from Maconomy.

All tasks on the project that are also imported from Maconomy are found and remembered by their name, Oid in People Planner, and first instance key from Maconomy.

For these tasks, all assignments and bookings that finish after the start date that is provided as input are found. The bookings are summed up on the month level. If a booking covers more than a single month, it is split across months, taking into account the resource's capacity. The first six months are returned as Month(1-6)PlannedHours, and bookings after the sixth month are returned as FuturePlannedHours. The splitting up of a booking that spans more than one month takes into account the resource's calendars to make the sum correct.

Example reply (requesting half-year budget after August 1, 2016 on the original project without the call of MaconomySetHalfYearBudget):

```
<?xml version="1.0" encoding="UTF-8"?>
<PeoplePlannerHalfYearBudget xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <StartDate>2016-08-01T00:00:00</StartDate>
    <ProjectNumber>1120006</ProjectNumber>
    <BudgetLines>
        <PeoplePlannerBudgetLine>
            <MaconomyID>JobBudgetLine87C022EA-D989-4E08-93ED-23DBAA7E8049</MaconomyID>
            <Name>Project Management</Name>
```

```
        <Employees>
            <Employee>
                <EmployeeNumber>2064</EmployeeNumber>
                <PeoplePlannerId>c5c8f0ea-090d-46ed-9cfd-
                3616f9906e1c</PeoplePlannerId>
                <Month1PlannedHours>65</Month1PlannedHours>
                <Month2PlannedHours>0</Month2PlannedHours>
                <Month3PlannedHours>0</Month3PlannedHours>
                <Month4PlannedHours>0</Month4PlannedHours>
                <Month5PlannedHours>0</Month5PlannedHours>
                <Month6PlannedHours>0</Month6PlannedHours>
                <FuturePlannedHours>0</FuturePlannedHours>
            </Employee>
        </Employees>
        <PeoplePlannerID>6a422d2a-95d9-4def-83d3-738b275dde47</PeoplePlannerID>
    </PeoplePlannerBudgetLine>
    <PeoplePlannerBudgetLine>
        <MaconomyID>JobBudgetLineD4E8A260-530B-4865-BD79-5EB38670B84D</MaconomyID>
        <Name>Consulting</Name>
        <Employees>
            <Employee>
                <EmployeeNumber>2032</EmployeeNumber>
                <PeoplePlannerId>f197f94b-c3aa-4231-befa-
                239ee04e515f</PeoplePlannerId>
                <Month1PlannedHours>60</Month1PlannedHours>
                <Month2PlannedHours>0</Month2PlannedHours>
                <Month3PlannedHours>0</Month3PlannedHours>
                <Month4PlannedHours>0</Month4PlannedHours>
                <Month5PlannedHours>0</Month5PlannedHours>
                <Month6PlannedHours>0</Month6PlannedHours>
                <FuturePlannedHours>0</FuturePlannedHours>
            </Employee>
        </Employees>
        <PeoplePlannerID>be9b5efb-2c89-488e-8fee-d46e7cdeabe4</PeoplePlannerID>
    </PeoplePlannerBudgetLine>
    <PeoplePlannerBudgetLine>
        <MaconomyID>JobBudgetLineE3871BE6-A7FB-4784-A7AB-828C71F8EDA4</MaconomyID>
        <Name>Documentation</Name>
        <Employees>
            <Employee>
                <EmployeeNumber>3038</EmployeeNumber>
```

```
                <PeoplePlannerId>199ea9e9-baa5-4b54-ad29-
                7d282eda3a65</PeoplePlannerId>
                <Month1PlannedHours>90</Month1PlannedHours>
                <Month2PlannedHours>0</Month2PlannedHours>
                <Month3PlannedHours>0</Month3PlannedHours>
                <Month4PlannedHours>0</Month4PlannedHours>
                <Month5PlannedHours>0</Month5PlannedHours>
                <Month6PlannedHours>0</Month6PlannedHours>
                <FuturePlannedHours>0</FuturePlannedHours>
            </Employee>
        </Employees>
        <PeoplePlannerID>9ec04c03-82de-4e75-89c7-de596f31039e</PeoplePlannerID>
    </PeoplePlannerBudgetLine>
  </BudgetLines>
</PeoplePlannerHalfYearBudget>
```

People Planner Silent Sign-In (SSI)

GetSecurityToken

Call this method to get a Silent Sign-In token (SSI) from People Planner.

To call GetSecurityToken, People Planner must be set up correctly, with both a People Planner and a Maconomy secret key. See the *Deltek People Planner Technical Installation Guide* for information about how to do this.

Arguments

This method takes the following argument:

- string encryptedMessage — An encrypted message that must match settings for SSI Maconomy secret

This method decrypts the request received using the Maconomy secret key. It uses AES (Rijndael) encryption.

The decrypted message must contain information about a user that exists in the People Planner database and a timestamp in the following format:

```
user=<NetworkDomainName>\<NetworkUsername>;timestamp=2017-06-21T06:35:26;
```

Reply

Returns a struct of type WebSecurityToken

If the decrypted message did not contain a user that exists in People Planner, GetSecurityToken still returns an SSI token. However, subsequent calls to People Planner using this token will fail.

WebSecurityToken:

- string Token — An encrypted message about a People Planner user (encrypted with People Planner secret)

- DateTime ValidTo — How long the access to People Planner is valid

The token is encrypted with the People Planner secret key. However, it is recommended that you do not store this password at the client or try to decrypt the token. The token should simply be appended to any URL that calls either MyPlan or the People Planner Web Components.

Example reply:

```
<?xml version="1.0" encoding="UTF-8"?>

<WebSecurityToken xmlns="http://marstrand-innovation.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

   <Token>Tbxo4eI8MWT%2bnR0I2nlx1AQMDwMBAw8BKQMKBAUJCwEAKbT%2bpG02jSUK4k1ygUe2MI%2ft9Z
   ZxoMVKDBlC%2fOh6GDyhSAwnMz4YA7rKX%2fw2Bq1KmjifQc81KovyLNI4yhyZ</Token>

   <ValidTo>2017-06-16T16:25:19</ValidTo>

</WebSecurityToken>
```

About Deltek

Better software means better projects. Deltek is the leading global provider of enterprise software and information solutions for project-based businesses. More than 23,000 organizations and millions of users in over 80 countries around the world rely on Deltek for superior levels of project intelligence, management and collaboration. Our industry-focused expertise powers project success by helping firms achieve performance that maximizes productivity and revenue. www.deltek.com