

# Web Service Functions

List of Web Services functions examples and their explanation.

- [CreateFolderItem](#)
- [UpdateFormItemValues](#)
- [GetDocument](#)
- [ListDocument](#)
- [CreateGlobalText](#)
- [CreateMarketingTriggers](#)
- [NewMail](#)
- [GetSchedulerSettings](#)
- [CreateProjectItem](#)
- [ListLanguages](#)
- [ListContactList](#)
- [ModifyProjectItemValues](#)
- [CreateEvent](#)

# CreateFolderItem

## Description

Create recurring billing request. Recurring billing request is used to instruct the server to create individual instance of **Billing Request**, which is a source for Invoice creation.

## Important Parameters

amount	Unit price to be charged
billingDate	Sample date, in which the Billing Request will be used as a date for invoicing.
billingStatus	1 .. Pending 2 .. For Approval (Sales Order may be generated here) 3 .. To Be Invoiced (Invoice may be generated here)
totalQuantity	Pcs to be charged per Unit price after free Qty is reduced
catalogPrice	Price before any discounts
item	Product ID
name	Main title of the future Billing Request
note	Future description of Billing Request
purchasePrice	Unit price for which the goods is purchased
Quantity	Pcs to be charged per Unit Price
folder	Customer folder ID
totalAmount	total qty * amount
totalAmountWithVat	totalAmount + VAT
vatValue	VAT rate in numbers
ratio	fixed value (100) = 100%
taxationDate	Date relative to Billing Date, in which you may decide to put the transaction into current / previous / next periods. Put any date close to Billing Date (ie any day in the same/previous month)
isModelItem	1

validFrom	Date, from which the rule becomes active. Difference of dates between validFrom and billingDate is important. Item generated in validFrom = today should have billingDate today + 1 year, in case the first next fee is paid after one year of using the service.
validTo	Expiration date
destinationProject	Target Project ID, where the Billing Request will be created.
billingFrequency	2 .. yearly 4 .. quarterly 6 .. monthly 8 .. weekly 10 .. daily

## Request Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header />
  <soapenv:Body>
    <wSDL:CreateFolderItem>
      <server>acme</server>
      <session>7CDF678513DCBF0379DDD26A6961B604D7F1515CBD04</session>
      <ITEM>
        <amount>1234</amount>
        <billingDate>2020-05-14T00:00:00</billingDate>
        <billingStatus>2</billingStatus>
        <totalQuantity>5</totalQuantity>
        <catalogPrice>190</catalogPrice>
        <currency>CZK</currency>
        <item>133629101</item>
        <name>Repeated service</name>
        <note>Billing the yearly service.</note>
        <purchasePrice>130</purchasePrice>
        <quantity>5</quantity>
        <folder>139676101</folder>
        <totalAmount>6170</totalAmount>
        <totalAmountWithVat>6787</totalAmountWithVat>
        <vatValue>10</vatValue>
        <ratio>100</ratio>
        <taxationDate>2020-05-14T00:00:00</taxationDate>
        <validFrom>2020-05-14T00:00:00</validFrom>
```

```
        <validTo>2099-05-04T00:00:00</validTo>
        <destinationProject>139704101</destinationProject>
        <billingFrequency>2</billingFrequency>
    </ITEM>
</wsdl:CreateFolderItem>
</soapenv:Body>
</soapenv:Envelope>
```

## Response Example

```
<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wsdl">
  <ns0:Body>
    <ns1:CreateFolderItemResponse>
      <result>OK</result>
      <time>0.01345706</time>
      <count>0</count>
      <itemId>139812101</itemId>
    </ns1:CreateFolderItemResponse>
  </ns0:Body>
</ns0:Envelope>
```

# UpdateFormItemValues

There are several scenarios to use this function, based on knowledge you have about the form, where you want to save the data.

## FormValuesId + formItemName

You may gather **formValuesId** from context object using get function (getFolder, getProject, getActivity). **FormValuesId** is bigint which identifies set of values of one form instance.

**formItemName** is the name of form field. Form fields may be listed using **ListFormItem** method.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header />
  <soapenv:Body>
    <wSDL:UpdateFormItemValues>
      <server>acme</server>
      <session>...string_here...</session>
      <formValuesId>392302101</formValuesId>
      <ITEMVALUE>
        <formItemName>form- component- name- here</formItemName>
        <value>123456</value>
      </ITEMVALUE>
    </wSDL:UpdateFormItemValues>
  </soapenv:Body>
</soapenv:Envelope>
```

# GetDocument

## WSDL

document\_library.wsdl

## Required parameter

- id - id of saved file

## Example Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wsdl">
  <soapenv:Header />
  <soapenv:Body>
    <wSDL:GetDocument>
      <server>multi</server>
      <session>BDA7025DFBA9CDF53AE306DB4372AF3492FA59DBBD04</session>
      <id>141942102</id>
    </wSDL:GetDocument>
  </soapenv:Body>
</soapenv:Envelope>
```

## Example Response

```
<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wsdl">
  <ns0:Body>
    <ns1:GetDocumentResponse>
      <result>OK</result>
      <time>0.01405215</time>
      <count>1</count>
      <id>141942102</id>
      <DOCUMENT>
        <folder />
        <createdByName>AdminAtollon</createdByName>
        <id>141942102</id>
```

```
<acl>DLIB000000008997</acl>
<name>1PMP- Canvas. pdf</name>
<description/>
<descriptionId/>
<validByDate>0</validByDate>
<createdBy>133780102</createdBy>
<fileHandle>2020/06/22/141941102</fileHandle>
<lang>0</lang>
<treeHandle>134567102</treeHandle>
<fileSize>204248</fileSize>
<printed>0</printed>
<internalNumber />
<bar code/>
<internalType>3</internalType>
<fileName>1PMP- Canvas. pdf</fileName>
<lock>0</lock>
<deleted>0</deleted>
<creationDate>2020- 06- 22T16: 27: 00</creationDate>
<lastChangeDate>2020- 06- 22T16: 27: 00</lastChangeDate>
<validFrom>2020- 06- 22T00: 00: 00</validFrom>
<validTo>2020- 06- 22T00: 00: 00</validTo>
<parentName/>
<originalName/>
<categoryName/>
<locationName/>
<typeName/>
<storageId>134687102</storageId>
<storageSubdir />
<isInStorage>1</isInStorage>
<mimeType>application/pdf; charset=binary</mimeType>
<contextId>134567102</contextId>
</DOCUMENT>
```

```
</ns1: GetDocumentResponse>
```

```
</ns0: Body>
```

```
</ns0: Envelope>
```

# ListDocument

Function that lists document in given folder. Documents can be stored in most of the folders of tree structure. Therefore any context ID (Folder, Project or Activity) may act as document placeholder = **parent**

## WSDL

document\_library.wsdl

## Example Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header/>
  <soapenv:Body>
    <wSDL:ListDocument>
      <server>multi</server>
      <session>BDA7025DFBA9CDF53AE306DB4372AF3492FA59DBBD04</session>
      <limit>100</limit>
      <offset>0</offset>
      <parent>134567102</parent>
      <RESTRICTIONS>and aa.doc_parent is null</RESTRICTIONS>
    </wSDL:ListDocument>
  </soapenv:Body>
</soapenv:Envelope>
```

The restriction means no older document versions would be returned as a result of query.

## Example Response

```
<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wSDL">
  <ns0:Body>
    <ns1:ListDocumentResponse>
      <result>OK</result>
      <time>0.01323104</time>
      <count>1</count>
```



```
<DOCUMENT>
  <folder />
  <createdByName>AdminAtollon</createdByName>
  <id>141942102</id>
  <acl>DLIB000000008997</acl>
  <name>1PMP- Canvas. pdf</name>
  <description />
  <descriptionId />
  <validByDate>0</validByDate>
  <createdBy>133780102</createdBy>
  <fileHandle>2020/06/22/141941102</fileHandle>
  <lang>0</lang>
  <treeHandle>134567102</treeHandle>
  <fileSize>204248</fileSize>
  <printed>0</printed>
  <internalNumber />
  <bar code />
  <internalType>3</internalType>
  <fileName>1PMP- Canvas. pdf</fileName>
  <lock>0</lock>
  <deleted>0</deleted>
  <creationDate>2020- 06- 22T16: 27: 00</creationDate>
  <lastChangeDate>2020- 06- 22T16: 27: 00</lastChangeDate>
  <validFrom>2020- 06- 22T00: 00: 00</validFrom>
  <validTo>2020- 06- 22T00: 00: 00</validTo>
  <parentName />
  <originalName />
  <categoryName />
  <locationName />
  <typeName />
  <storageId>134687102</storageId>
  <storageSubdir />
  <isInStorage>1</isInStorage>
  <mimeType>application/pdf; charset=binary</mimeType>
</DOCUMENT>
</ns1: ListDocumentResponse>
</ns0: Body>
</ns0: Envelope>
```

# CreateGlobalText

Global texts are used as storage of localizable text blocks, usually used in printing.

Part of [translate.wSDL](#)

LANG must be EN.

Below is the example of Invoice footer print.

## Request Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header />
  <soapenv:Body>
    <wSDL:CreateGlobalText>
      <server>acme</server>
      <session>4D2...D04</session>
      <identification>invoice- footer</identification>
      <LANG>
        <id>131485101</id>
        <value>In case of any questions, or request for additional details, do not
hesitate to contact us by email to invoice@company.com or using phone number +44 00099333.
Yours sincerely, Your Team.</value>
      </LANG>
    </wSDL:CreateGlobalText>
  </soapenv:Body>
</soapenv:Envelope>
```

## Response Example

```
<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wSDL">
  <ns0:Body>
    <ns1:CreateGlobalTextResponse>
```

```
<result>OK</result>

<count>0</count>

<itemId>1936935101</itemId>

</ns1: CreateGlobalTextResponse>

</ns0: Body>

</ns0: Envelope>
```

## Related Functions

[ListLanguages](#)

# Global Text Idents

Below you can find examples of Global text idents that are used mainly as text blocks when printing documents from ATOLLON.

Document	Global text IDENT	Description
Invoice issued	invoice-footer	Print text at the bottom of invoice
Estimate	estimate-footer	Print text at the bottom of estimate
Invoice/estimate/order/offer	invoice-global-CSS	Use corporate CSS

# CreateMarketingTriggers

## Description

Marketing triggers are used to track any activity of the contact / public user, which have some meaning in your company sales or marketing process.

## Fields Description

Field name	Example	Description
baseType	new-phone-number url-visit goal-achieved mail-url-visit mail-open url-visit unsubscribe	Text identification of the marketing trigger (what actually happened)
contextId	123456789000	Identification of Folder ID, Project ID or Activity ID in Atollon's context structure. It is necessary to enter context ID in case it should be displayed in context Timeline.
contactId	113456789000	Identification of contact, who caused the trigger
url	https://yourcompany.atollon.com/pub/23423432	Page that the contact visited (optional)
trackingCode	yourSuperCampaignXYZ	Any text string representing one particular campaign / marketing activity / message template
description	Contact visited URL	Any descriptive text that will be visible to Atollon users in Timeline
triggerDate	2020-08-12 12:00	When the trigger happened?
numberOfVisits	45	In case the target user is super active (or robot), we may want to avoid spamming Atollon with unnecessary records and aggregate page visits

The example below notifies Atollon about new phone number added to contact. Atollon may therefore create call task.

## Example Request

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header/>
  <soapenv:Body>
    <wSDL:CreateMarketingTriggers>
      <server>acme</server>
      <session>61A...DBD04</session>
      <MARKETING_TRIGGER>
        <baseType>new-phone-number</baseType>
        <contextId>132716101</contextId>
        <contactId>139663101</contactId>
        <url>www.yourcompany.com/product/page</url>
        <trackingCode>www.company.com/product/page</trackingCode>
        <description>Extranet user added new phone number</description>
        <triggerDate>2020-07-31 12:00</triggerDate>
        <numberOfVisits>1</numberOfVisits>
      </MARKETING_TRIGGER>
    </wSDL:CreateMarketingTriggers>
  </soapenv:Body>
</soapenv:Envelope>

```

## Example Response

```

<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wSDL">
  <ns0:Body>
    <ns1:CreateMarketingTriggersResponse>
      <result>OK</result>
      <time>0.22972488</time>
      <newId>400097101</newId>
    </ns1:CreateMarketingTriggersResponse>
  </ns0:Body>
</ns0:Envelope>

```

## WSDL Listing

**web.wSDL**

# NewMail

## Description

New mail function is used to send regular e-mail from Atollon. It supports sending both plaintext and html e-mails. In case html e-mail is sent, plaintext part is generated automatically.

## Fields Description

Field	Description	Example
priority	E-mail priority handled by other mail clients	1, 2, 3
group	Message group with similar characteristics, required	For e-mails sent in context, there is always default "group" available in the response of request to getFolder, getProject, getActivity. For example:  <messageGroup>133185101</messageGroup>
treeHandle	Message container (mailbox)	Similar comment to "group" field above: <messageDefault>133186101</messageDefault>
html	What is the body format?	0..plaintext 1..html
encoding	Use utf-8 only	utf-8
from	Full format of sender mailbox	
to	Recipient address	
subject	Subject of the message	
body	Either plaintext or html part of the message (based on above html = 0/1 flag)	
attachments.attachment	Link to documentId in Atollon. The document will be converted to base64 and added to sent e-mail	
forwardEmails	List ids of e-mail messages that will be attached	

## Example

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header/>
  <soapenv:Body>
    <wSDL:NewMail>
      <server>acme</server>
      <session>5B5EA...8BD04</session>
      <mail>
        <priority>2</priority>
        <group>491826101</group>
        <treeHandle>491827101</treeHandle>
        <html>1</html>
        <encoding>utf-8</encoding>
        <from>"Your team" &lt;admin@mydomain.com&gt;</from>
        <to>jan@safka.cz</to>
        <subject>Welcome to your new world of success</subject>
        <body>&lt;html&gt;&lt;head&gt;&lt;title&gt;Welcome to your new world of
success&lt;/title&gt;&lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/&gt;&lt;style type="text/css"&gt;&lt;!-- body,td,th { font-family: Arial, Helvetica, sans-
serif; font-size: 12px; color: #333333; } a:link { color: #666666; } a:visited { color:
#999999; } --&gt;&lt;/style&gt;&lt;/head&gt;&lt;body&gt;&lt;div ALIGN="LEFT"&gt;&lt;span
style="font-family: helvetica, arial, sans-serif; font-size:12px; color: #000000; "&gt;This is the
secret message&lt;/span&gt;&lt;/div&gt;

&lt;br /&gt;

&lt;div ALIGN="LEFT"&gt;&lt;span style="font-family: helvetica, arial, sans-serif; font-
size: 12px; color: #000000; "&gt;Your super powerful team&lt;/span&gt;&lt;/div&gt;

&lt;/body&gt;&lt;/html&gt;</body>
      <attachments>
        <attachment>
          <documentId>496477101</documentId>
        </attachment>
      </attachments>
      <forwardEmails/>
    </mail>
  </wSDL:NewMail>
</soapenv:Body>
</soapenv:Envelope>

```

## Response

```
<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wSDL">
  <ns0:Body>
    <ns1:NewMailResponse>
      <result>OK</result>
      <time>0.07684398</time>
      <newId>496505101</newId>
      <newAcl>MSG5000000151828</newAcl>
      <needApproval>0</needApproval>
    </ns1:NewMailResponse>
  </ns0:Body>
</ns0:Envelope>
```



# GetSchedulerSettings

## Description

Function that takes care about administration of lists connected with Events (calendar), Tasks and Issues.

Settings are administered using one function only - UpdateSchedulerSettings.

## Type of settings

ListType	Settings
1	Scheduler status
2	Severity
3	Solution status
4	Type of work

# CreateProjectItem

## Description

Create **billing request**. Billing request is used as a source for creating new invoice. That invoice can be created manually from Billing Invoices, using Draft Invoice generation or using one-off function `CreateInvoicesFromItems`.

## Important Parameters

amount	Unit price to be charged
billingDate	Date, in which the Billing Request will become valid for invoicing.
billingStatus	0 .. Sales Items (representing Deal Value)  1 .. Pending 2 .. For Approval (Sales Order may be generated here) 3 .. Approved by manager (Invoice may be generated here, only users with Authorize rights to project may approve billing items) 4 .. Billed 5 .. Approved by client 6 .. Postponed 7 .. Canceled  999 .. Installed technology project items // will be canceled
type	1 .. sales items or billing items 2 .. auto-generated time sheet item 3 .. installed technology
catalogPrice	Price before any discounts (also referred as Recommended Price, Price List Price, ...). The original price, from which discount is calculated.
item	Product ID
name	Main title of the Billing Request
note	Description of Billing Request
purchasePrice	Unit price for which the goods is purchased
quantity	Pcs to be charged per Unit Price
folder	Customer folder ID
totalAmount	qty * amount

totalAmountWithVat	totalAmount + VAT
vatValue	VAT rate in numbers
ratio	fixed value (100) = 100% (used only in sales opportunity/deal to evaluate probability of the item in the sales basket)
project	Project ID, where the Billing Request will be created.

## Request Example

### CreateProjectItem (item.wsdl) Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsdl="http://atollon.com/enterprise/wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <wsdl:CreateProjectItem>
      <server>acme</server>
      <session>6A941A9BC9EBEFAF585A4B936AA958755B1C85E2BD04</session>
      <ITEM>
        <amount>1234</amount>
        <billingDate>2020-05-14T00:00:00</billingDate>
        <billingStatus>3</billingStatus>
        <totalQuantity>5</totalQuantity>
        <catalogPrice>190</catalogPrice>
        <currency>CZK</currency>
        <item>133629101</item>
        <name>AAA Batteries</name>
        <project>139704101</project>
        <purchasePrice>130</purchasePrice>
        <quantity>5</quantity>
        <totalAmount>6170</totalAmount>
        <totalAmountWithVat>6787</totalAmountWithVat>
        <vatValue>10</vatValue>
        <ratio>100</ratio>
      </ITEM>
    </wsdl:CreateProjectItem>
  </soapenv:Body>
</soapenv:Envelope>
```

### CreateProjectItem Response

```
<ns0:Envelope xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://atollon.com/enterprise/wSDL">
  <ns0:Body>
    <ns1:CreateProjectItemResponse>
      <result>OK</result>
      <time>0.01379299</time>
      <count>0</count>
      <itemId>139721101</itemId>
    </ns1:CreateProjectItemResponse>
  </ns0:Body>
</ns0:Envelope>
```

# ListLanguages

Part of [translate.wSDL](#)

## Sample Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header/>
  <soapenv:Body>
    <wSDL:ListLanguages>
      <server>saas</server>
      <session>FB...C3</session>
    </wSDL:ListLanguages>
  </soapenv:Body>
</soapenv:Envelope>
```

## Sample Response

Reduced to one item.

```
<LANGUAGE>
  <charset>iso-8859-2</charset>
  <id>2607000</id>
  <languageLong>Czech</languageLong>
  <languageShort>CZE</languageShort>
  <isoShort>cz</isoShort>
  <alternativeShort/>
  <priority>0</priority>
  <useForTranslate>1</useForTranslate>
  <useForWeb>1</useForWeb>
</LANGUAGE>
```

# ListContactList

## Description

Function listing basic contact configuration items / lists.

## Important Parameters

list	contacttype industry nationality companylegalform category department position addresstype relationtype_p (person) relationtype_c (company)
treeHandle	either use "node" attribute from <a href="#">Login</a> function, DEFAULTS->DEFAULT with "ident" CONTACTCONFIGNODE or use attrib <b>useDefaultTreeHandle</b> = true

## Request Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://atollon.com/enterprise/wSDL">
  <soapenv:Header/>
  <soapenv:Body>
    <wSDL:ListContactList>
      <server>atollon</server>
      <session>4A...04</session>
      <treeHandle>1685000</treeHandle>
      <list>contacttype</list>
    </wSDL:ListContactList>
  </soapenv:Body>
</soapenv:Envelope>
```

## Response Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <atollon:ListContactListResponse xmlns:atollon="http://iterity.com/enterprise/wSDL">
      <result>OK</result>
      <time>0.01314497</time>
      <count>12</count>
      <ITEM>
        <id>1883000</id>
        <name>E-mail (other)</name>
        <type>1</type>
      </ITEM>
      <ITEM>
        <id>1884000</id>
        <name>E-mail (personal)</name>
        <type>1</type>
      </ITEM>
      <ITEM>
        <id>1885000</id>
        <name>E-mail (work)</name>
        <type>1</type>
      </ITEM>
      ...
    </atollon:ListContactListResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

# ModifyProjectItemValues

## 8 metod nastavení cen položek kufříku

- **SetPurchaseByMargin** z: Unit Sales Price a Margin
  - Změna nákupní jednotkové ceny na základě zadané marže, vychází z jednotkové ceny prodejní.
    - **$\text{purchase\_price} = \text{amount} \times (1 - \text{value})$**

```
SetPurchaseByMargin: purchase_price = amount x (1 - value)
```

- **SetSalesByMargin** z: Unit Purchase Price a Margin
  - Změna prodejní jednotkové ceny na základě zadané marže, vychází z jednotkové ceny nákupní.
    - **$\text{amount} = \text{purchase\_price} / (1 - \text{value})$**

```
SetSalesByMargin: amount = purchase_price / (1 - value)
```

- **SetPurchaseByPurchase**
  - Změna nákupní jednotkové ceny na základě zadané nové **celkové nákupní ceny**, postup (zkráceně: zvýšit/snížit jednotkovou cenu dle % změny celkové ceny):
    - sečíst původní položky  **$\text{sum}(\text{purchase\_price} \times \text{quantity})$**  (itm\_project\_item\_par))  
**= total\_purchase\_old**
    - hodnota **value** reprezentuje novou celkovou nákupní cenu
    - zjistit % změnu  **$\text{dif} = \text{value} / \text{total\_purchase\_old}$**
    - změnit hodnotu všech stávajících jednotkových  **$\text{purchase\_price} = \text{purchase\_price}(\text{itm\_project\_item\_par}) \times \text{value}$**
    - pokud je  **$\text{sum purchase\_price} = 0$** , potom se jde na metodu **SetUnitPurchaseToTotalPurchaseBySalesUnit**

```
SetPurchaseByPurchase: purchase_price=purchase_price*( value/sum(purchase_price*quantity))
```

- **SetPurchaseBySales**
  - pokud jsou všechny nákupní ceny = 0, potom se použije výpočet obdobný, ale bude vycházet na základě cen prodejních (tedy  **$\text{sum}(\text{amount})$** )



- pokud je  $\text{sum amount} = 0$ , potom se jde na metodu `SetUnitPurchaseToTotalPurchaseByQty`

`SetPurchaseBySales: purchase_price=amount*( value/sum( amount*quantity))`

#### • **SetPurchaseByQuantity**

- pokud jsou všechny prodejní ceny = 0, potom se použije výpočet obdobný, ale bude vycházet na základě počtu ks (qty) ... tedy spočítá se jednotková nákupní cena za položku ...  $\text{purchasePerQty} = \text{value} / \text{qty}$  a tou se roznásobí hodnota - tedy  $\text{purchase\_price} * \text{purchasePerQty}$

`SetPurchaseByQuantity: purchase_price=( value/sum( quantity))`

#### • **SetSalesBySales**

- Změna prodejní jednotkové ceny na základě zadané nové **celkové prodejní ceny**, postup (zkráceně: zvýšit/snížit jednotkovou cenu dle % změny celkové ceny):
  - sečíst původní položky  $\text{sum}(\text{amount} * \text{quantity} \text{ (itm\_project\_item\_par)}) = \text{amount\_without\_vat\_old}$
  - hodnota **value** reprezentuje novou celkovou prodejní cenu
  - zjistit % změnu  $\text{diff} = \text{value} / \text{amount\_without\_vat\_old}$
  - změnit hodnotu všech stávajících jednotkových  $\text{amount} = \text{amount} \text{ (itm\_project\_item\_par)} * \text{diff}$
  - pokud jsou všechny prodejní ceny  $\text{sum}(\text{amount}) = 0$ , použije se `SetUnitSalesToTotalSalesByUnitPurchase`

`SetSalesBySales: amount=amount*( value/sum( amount*quantity))`

#### • **SetSalesByPurchase**

- pokud jsou všechny prodejní ceny = 0, potom se použije výpočet obdobný, ale bude vycházet na základě cen nákupních (tedy  $\text{sum}(\text{purchase\_price})$ )
- pokud jsou všechny nákupní ceny  $\text{sum}(\text{purchase\_price}) = 0$ , použije se `SetUnitSalesToTotalSalesByQty`

`SetSalesByPurchase: amount=purchase_price*( value/sum( purchase_price*quantity))`

#### • **SetSalesByQuantity**

- pokud jsou všechny nákupní ceny = 0, potom se použije výpočet obdobný, ale bude vycházet na základě počtu ks (qty) - vypočítá se  $\text{pricePerQty} = \text{value} / \text{Qty}$ , pak pro každou položku  $\text{Qty} * \text{pricePerQty}$

`SetSalesByQuantity: amount=( value/sum( quantity))`

# CreateEvent

## WSDL

scheduler.wsdl

## Related Services

[GetSchedulerSettings](#)

Contains several configurations of lists, incl. **Solution Status** and **Type of Work (entryType)** used in creating New Event. Please note that Solution Status should be selected based on Type of work and current user's profile.

## Example Request

```
Operation request: 'CreateEvent'
<persistency-request operationId='CreateEvent'>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <SOAP-ENV:Body>
    <ns1:CreateEvent xmlns:ns1="http://atollon.com/enterprise/wsdl">
      <server>acme</server>
      <session>...</session>
      <ITEM>
        <chargeable>0</chargeable>
        <remoteHeadId>0</remoteHeadId>
        <remoteId>0</remoteId>
        <resourceId>0</resourceId>
        <status>0</status>
        <userId>/-- ID of event owner-user--</userId>
        <completed xsi:nil="true"/>
        <multiEventTask>0</multiEventTask>
        <forUser>/-- ID of event owner-user--</forUser>
        <location>Event location text</location>
        <iversion></iversion>
        <solutionStatus>/-- ID of custom state--</solutionStatus>
```

```
<developmentStatus>0</developmentStatus>
<severity>0</severity>
<isPrivate>0</isPrivate>
<activityId xsi:nil="true" />
<projectId xsi:nil="true" />
<folderId>//-- ID of customer folder--//</folderId>
<nodeId>0</nodeId>
<customId>0</customId>
<comments></comments>
<reminder>0</reminder>
<addDuration>0</addDuration>
<estimationHours>0</estimationHours>
<actualHours>0</actualHours>
<milestones>0</milestones>
<description>Event description text ...</description>
<htmlText></htmlText>
<dateTo>2023-03-09T12:30:00</dateTo>
<dateFrom>2023-03-09T12:00:00</dateFrom>
<priority>0</priority>
<entryType>//-- ID of type of work--//</entryType>
<distributionHandle>//----//</distributionHandle>
<aclh xsi:nil="true" />
<headId>0</headId>
<headUserId>//--ID of event organizer-user--//</headUserId>
<title>Event summary main text</title>
<ident xsi:nil="true" />
<dimension1>0</dimension1>
<dimension2>0</dimension2>
<dimension3>0</dimension3>
<taskTemplateId xsi:nil="true" />
<USER>
  <id>//--ID of event owner-user--//</id>
</USER>
<REPEAT>
  <repeatMode>2</repeatMode>
  <validFrom>2023-03-09T12:06:23</validFrom>
  <validTo>2023-04-09T00:00:00</validTo>
  <repeatEvery>1</repeatEvery>
  <repeatDay>0</repeatDay>
  <repeatMonth>0</repeatMonth>
```

```

        <repeatWeekDay>0</repeatWeekDay>
        <repeatWeekCount>0</repeatWeekCount>
        <repeatWeekDayList>4</repeatWeekDayList>
        <timeFrom>12: 00</timeFrom>
        <timeTo>12: 30</timeTo>
        <useOffset>0</useOffset>
        <offset>0</offset>
        <limit>1</limit>
    </REPEAT>
    <notify>1</notify>
    <refId></refId>
</ITEM>
</ns1: CreateEvent>
</SOAP- ENV: Body>
</SOAP- ENV: Envelope>
</persistency- request>

```

## Example Response

```

Operation response: ' CreateEvent'
<persistency- response operationId=' CreateEvent' >
<ns1: CreateEventResponse xmlns: ns1="http: //atollon. com/enterprise/wsd1"
xmlns: ns0="http: //schemas. xmlsoap. org/soap/envelope/">
    <result>OK</result>
    <name>create</name>
    <headId>107751483081</headId>
    <commonId>107751484081</commonId>
    <newId>107751483081</newId>
</ns1: CreateEventResponse>
</persistency- response>

```