

CENNO API SPECIFICATION

v.1.1

Contents

Contents.....	2
1. Introduction.....	4
2. API access	4
Authentication.....	4
Testing.....	4
3. API.....	4
Company.....	4
Get list	4
Get one by id.....	5
User.....	6
Get list	6
Get one by id.....	8
Project.....	9
Get list	9
Get one by id.....	10
Category	10
Get list	10
Get one by id.....	11
Item.....	12
Get list	12
Get one by id.....	13
Time entry.....	14
Get list	14
Get one by id.....	15
Create new.....	16
Update.....	17
Delete	18
Validation errors.....	18
Not found resource	18

User status.....	18
Get list	18
Get one by id.....	19
Create new.....	19
Update.....	20
Delete	20
Validation errors.....	21
Not found resource	21
Project comment.....	21
Get list	21
Get one by id.....	22
Create new.....	22
Update.....	23
Delete	24
Validation errors.....	24
Not found resource	24
Item comment.....	24
Get list	24
Get one by id.....	25
Create new.....	25
Update.....	26
Delete	27
Validation errors.....	27
Not found resource	27
Searching resources	27
Advanced searching.....	28
Searching using associations.....	29

1. Introduction

The document describes the external API to access Cenno resources using simple HTTP REST API. It allows to delete, update, post and get resources.

2. API access

The API is accessible via URL <http://example.cennoapp.com/api> . 'example' should be replaced with account code.

Authentication

Authentication is done using basic HTTP authentication. API key should be used as username. Password can be left empty or can be the same as username.

Testing

Basic testing can be done with web browser or Java REST client tool:

<http://code.google.com/p/rest-client/>

3. API

Company

Get list

<http://example.cennoapp.com/api/companies.xml>

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<companies type="array">
  <company>
    <business></business>
    <city></city>
    <contract type="boolean">>false</contract>
    <contract-end type="date" nil="true"></contract-end>
    <contract-start type="date" nil="true"></contract-start>
    <country></country>
    <created-at type="datetime">2009-10-02T16:50:42+03:00</created-at>
    <id type="integer">5</id>
    <phone></phone>
```

```
<post-code></post-code>
<responsible-id type="integer">5</responsible-id>
<street></street>
<title>McDermott and Sons</title>
<type-id type="integer" nil="true"></type-id>
<updated-at type="datetime">2009-10-21T17:00:01+03:00</updated-at>
<url></url>
<external-id nil="true"/>
</company>
<company>
  <business>Velit hic enim</business>
  <city>Lailaburgh</city>
  <contract type="boolean">true</contract>
  <contract-end type="date">2009-06-12</contract-end>
  <contract-start type="date">2008-10-12</contract-start>
  <country>Wales</country>
  <created-at type="datetime">2009-10-02T19:51:08+03:00</created-at>
  <id type="integer">219</id>
  <phone>695.379.2251</phone>
  <post-code>16339-6339</post-code>
  <responsible-id type="integer">471</responsible-id>
  <street>95614 Roy Square</street>
  <title>Bailey, Wisoky and Kuhlman</title>
  <type-id type="integer">20</type-id>
  <updated-at type="datetime">2009-10-22T11:00:21+03:00</updated-at>
  <url>http://www.raugrimes.biz</url>
  <external-id nil="true"/>
</company>
</companies>
```

Get one by id

<http://example.cennoapp.com/api/companies/5.xml>

HTTP request method: GET.

Example response:

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

```
<business></business>
<city></city>
<contract type="boolean">>false</contract>
<contract-end type="date" nil="true"></contract-end>
<contract-start type="date" nil="true"></contract-start>
<country></country>
<created-at type="datetime">2009-10-02T16:50:42+03:00</created-at>
<id type="integer">5</id>
<phone></phone>
<post-code></post-code>
<responsible-id type="integer">5</responsible-id>
<street></street>
<title>McDermott and Sons</title>
<type-id type="integer" nil="true"></type-id>
<updated-at type="datetime">2009-10-21T17:00:01+03:00</updated-at>
<url></url>
```

```
<external-id nil="true"/>
```

```
</company>
```

User

Get list

<http://example.cennoapp.com/api/users.xml>

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<users type="array">
  <user>
    <account-owner type="boolean">>false</account-owner>
    <allow-access type="boolean">>false</allow-access>
    <birthday type="date">1955-01-10</birthday>
    <company-id type="integer">219</company-id>
    <created-at type="datetime">2009-10-02T19:51:08+03:00</created-at>
    <crypted-password nil="true"></crypted-password>
    <current-login-at type="datetime" nil="true"></current-login-at>
```

```
<current-login-ip nil="true"></current-login-ip>
<email>rene@muller.us</email>
<first-name>Alize</first-name>
<id type="integer">470</id>
<last-login-at type="datetime" nil="true"></last-login-at>
<last-login-ip nil="true"></last-login-ip>
<last-name>Hagenes</last-name>
<password-salt nil="true"></password-salt>
<perishable-token nil="true"></perishable-token>
<persistence-token nil="true"></persistence-token>
<phone>1-617-880-6316</phone>
<single-access-token nil="true"></single-access-token>
<title nil="true"></title>
<updated-at type="datetime">2009-10-02T19:51:27+03:00</updated-at>
<external-id nil="true"/>
</user>
<user>
  <account-owner type="boolean">>false</account-owner>
  <allow-access type="boolean">>false</allow-access>
  <birthday type="date">1959-09-28</birthday>
  <company-id type="integer">219</company-id>
  <created-at type="datetime">2009-10-02T19:51:08+03:00</created-at>
  <crypted-password nil="true"></crypted-password>
  <current-login-at type="datetime" nil="true"></current-login-at>
  <current-login-ip nil="true"></current-login-ip>
  <email>madisyn@bergesipes.co.uk</email>
  <first-name>Vida</first-name>
  <id type="integer">471</id>
  <last-login-at type="datetime" nil="true"></last-login-at>
  <last-login-ip nil="true"></last-login-ip>
  <last-name>Thompson</last-name>
  <password-salt nil="true"></password-salt>
  <perishable-token nil="true"></perishable-token>
  <persistence-token nil="true"></persistence-token>
  <phone>796.689.2124 x64039</phone>
  <single-access-token nil="true"></single-access-token>
  <title nil="true"></title>
```

```
<updated-at type="datetime">2009-10-02T19:51:27+03:00</updated-at>
<external-id nil="true"/>
</user>
</users>
```

Get one by id

<http://example.cennoapp.com/api/users/471.xml>

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <account-owner type="boolean">false</account-owner>
  <allow-access type="boolean">false</allow-access>
  <birthday type="date">1959-09-28</birthday>
  <company-id type="integer">219</company-id>
  <created-at type="datetime">2009-10-02T19:51:08+03:00</created-at>
  <rypted-password nil="true"></rypted-password>
  <current-login-at type="datetime" nil="true"></current-login-at>
  <current-login-ip nil="true"></current-login-ip>
  <email>madisyn@bergesipes.co.uk</email>
  <first-name>Vida</first-name>
  <id type="integer">471</id>
  <last-login-at type="datetime" nil="true"></last-login-at>
  <last-login-ip nil="true"></last-login-ip>
  <last-name>Thompson</last-name>
  <password-salt nil="true"></password-salt>
  <perishable-token nil="true"></perishable-token>
  <persistence-token nil="true"></persistence-token>
  <phone>796.689.2124 x64039</phone>
  <single-access-token nil="true"></single-access-token>
  <title nil="true"></title>
  <updated-at type="datetime">2009-10-02T19:51:27+03:00</updated-at>
  <external-id nil="true"/>
</user>
```

Project

Get list

<http://example.cennoapp.com/api/projects.xml>

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<projects type="array">
  <project>
    <company-id type="integer">220</company-id>
    <created-at type="datetime">2009-10-05T09:21:16+03:00</created-at>
    <description>Ipsam dignissimos rem sit non ab rerum nesciunt accusamus quis ut est impedit in ullam velit cumque. Quo nemo et reiciendis eius quia nisi soluta quis cupiditate nulla impedit. Ad blanditiis consectetur mollitia deleniti consequatur possimus ut eligendi provident labore inventore earum velit porro. Dolorem a ut possimus similique aut sit hic</description>
    <end-date type="date">2009-11-11</end-date>
    <external-id>reruma</external-id>
    <id type="integer">165</id>
    <start-date type="date">2009-10-31</start-date>
    <status-id type="integer">17</status-id>
    <title>Et ipsam nulla</title>
    <updated-at type="datetime">2009-10-26T16:32:16+02:00</updated-at>
  </project>
  <project>
    <company-id type="integer">222</company-id>
    <created-at type="datetime">2009-10-20T18:14:48+03:00</created-at>
    <description>Non repudiandae reiciendis ut repudiandae fuga sit inventore sint voluptatem quam qui. Ut beatae culpa animi magnam architecto reiciendis et numquam illum est veritatis laboriosam repellat sit quos consequatur est molestiae aliquid. Ut quasi necessitatibus reprehenderit in vel rem eum harum recusandae laudantium tenetur. Eveniet impedit aut quis amet aut nesciunt quia praesentium corporis tenetur perferendis nihil accusantium dignissimos totam voluptates voluptas qui. Omnis sit laborum quibusdam quia enim ut dolor voluptas explicabo adipisci est iure reiciendis eaque accusantium</description>
    <end-date type="date">2009-12-13</end-date>
    <external-id></external-id>
    <id type="integer">166</id>
    <start-date type="date">2009-11-02</start-date>
    <status-id type="integer">16</status-id>
```

```
<title>Ex excepturi</title>
<updated-at type="datetime">2009-10-21T13:48:29+03:00</updated-at>
</project>
</projects>
```

Get one by id

<http://example.cennoapp.com/api/projects/166.xml>

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <company-id type="integer">222</company-id>
  <created-at type="datetime">2009-10-20T18:14:48+03:00</created-at>
  <description>Non repudiandae reiciendis ut repudiandae fuga sit inventore sint voluptatem quam qui. Ut beatae culpa animi magnam architecto reiciendis et numquam illum est veritatis laboriosam repellat sit quos consequatur est molestiae aliquid. Ut quasi necessitatibus reprehenderit in vel rem eum harum recusandae laudantium tenetur. Eveniet impedit aut quis amet aut nesciunt quia praesentium corporis tenetur perferendis nihil accusantium dignissimos totam voluptates voluptas qui. Omnis sit laborum quibusdam quia enim ut dolor voluptas explicabo adipisci est iure reiciendis eaque accusantium</description>
  <end-date type="date">2009-12-13</end-date>
  <external-id></external-id>
  <id type="integer">166</id>
  <start-date type="date">2009-11-02</start-date>
  <status-id type="integer">16</status-id>
  <title>Ex excepturi</title>
  <updated-at type="datetime">2009-10-21T13:48:29+03:00</updated-at>
</project>
```

Category

Get list

<http://example.cennoapp.com/api/categories.xml>

HTTP request method: GET.

An example of XML response:

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

```
<categories type="array">
  <category>
    <created-at type="datetime">2009-10-02T19:51:27+03:00</created-at>
    <id type="integer">327</id>
    <position type="integer">0</position>
    <project-id type="integer">165</project-id>
    <title>Consequatur provident sint</title>
    <updated-at type="datetime">2009-10-02T19:51:27+03:00</updated-at>
    <external-id nil="true"/>
  </category>
  <category>
    <created-at type="datetime">2009-10-02T19:51:30+03:00</created-at>
    <id type="integer">346</id>
    <position type="integer">0</position>
    <project-id type="integer">172</project-id>
    <title>Repellendus</title>
    <updated-at type="datetime">2009-10-02T19:51:30+03:00</updated-at>
    <external-id nil="true"/>
  </category>
</categories>
```

Get one by id

<http://example.cennoapp.com/api/categories/346.xml>

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<category>
  <created-at type="datetime">2009-10-02T19:51:30+03:00</created-at>
  <id type="integer">346</id>
  <position type="integer">0</position>
  <project-id type="integer">172</project-id>
  <title>Repellendus</title>
  <updated-at type="datetime">2009-10-02T19:51:30+03:00</updated-at>
  <external-id nil="true"/>
</category>
```

Item

Get list

<http://example.cennoapp.com/api/items.xml>

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<items type="array">
  <item>
    <assignee-id type="integer">480</assignee-id>
    <category-id type="integer">353</category-id>
    <closed type="boolean">>false</closed>
    <created-at type="datetime">2009-10-02T19:51:33+03:00</created-at>
    <description>Consequuntur nemo ullam tenetur vero voluptas nobis inventore
dignissimos tempore necessitatibus ad quidem voluptas totam officiis beatae et.
Possimus iste quidem rem quaerat voluptatem rem aut. Nihil et consequatur rerum
exercitationem adipisci ea doloremque sed. Maxime odit laboriosam vel temporibus qui
accusantium mollitia earum enim deserunt</description>
    <estimated-hours type="float">58.0</estimated-hours>
    <id type="integer">3328</id>
    <importance type="integer">2</importance>
    <in-budget type="boolean">>true</in-budget>
    <owner-id type="integer">475</owner-id>
    <priority-id type="integer">19</priority-id>
    <project-id type="integer">179</project-id>
    <rate type="float">95.9348</rate>
    <remaining-hours type="float">24.0</remaining-hours>
    <spent-hours type="float">34.0</spent-hours>
    <status-id type="integer">19</status-id>
    <title>Et possimus eum in</title>
    <type-id type="integer">23</type-id>
    <updated-at type="datetime">2009-10-02T19:51:33+03:00</updated-at>
    <work-type-id type="integer">18</work-type-id>
    <external-id nil="true"/>
  </item>
  <item>
    <assignee-id type="integer">472</assignee-id>
```

```
<category-id type="integer">331</category-id>
<closed type="boolean">true</closed>
<created-at type="datetime">2009-10-02T19:51:28+03:00</created-at>
<description>Voluptatem illo optio magnam assumenda. Illum et omnis enim
accusamus vitae. Earum voluptas officiis dolor odio laudantium ducimus qui beatae at
itaque facilis nemo</description>
<estimated-hours type="float">23.0</estimated-hours>
<id type="integer">3075</id>
<importance type="integer">2</importance>
<in-budget type="boolean">>false</in-budget>
<owner-id type="integer">472</owner-id>
<priority-id type="integer">19</priority-id>
<project-id type="integer">167</project-id>
<rate type="float">95.9348</rate>
<remaining-hours type="float">-2.0</remaining-hours>
<spent-hours type="float">25.0</spent-hours>
<status-id type="integer">20</status-id>
<title>Culpa sequi quia aut non</title>
<type-id type="integer">20</type-id>
<updated-at type="datetime">2009-10-21T10:46:45+03:00</updated-at>
<work-type-id type="integer">18</work-type-id>
<external-id nil="true"/>
</item>
</items>
```

Get one by id

<http://example.cennoapp.com/api/items/3057.xml>

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<item>
  <assignee-id type="integer">474</assignee-id>
  <category-id type="integer">329</category-id>
  <closed type="boolean">>false</closed>
  <created-at type="datetime">2009-10-02T19:51:28+03:00</created-at>
```

```
<description>Laudantium animi nisi impedit iusto. Autem molestias mollitia qui  
facere velit excepturi possimus consequuntur distinctio ea iure aut id odio quis  
at</description>  
  
<estimated-hours type="float">42.0</estimated-hours>  
<id type="integer">3057</id>  
<importance type="integer">0</importance>  
<in-budget type="boolean">>false</in-budget>  
<owner-id type="integer">480</owner-id>  
<priority-id type="integer">18</priority-id>  
<project-id type="integer">166</project-id>  
<rate type="float">4.09503</rate>  
<remaining-hours type="float">6.0</remaining-hours>  
<spent-hours type="float">36.0</spent-hours>  
<status-id type="integer">19</status-id>  
<title>Unde est non a</title>  
<type-id type="integer">20</type-id>  
<updated-at type="datetime">2009-10-02T19:51:28+03:00</updated-at>  
<work-type-id type="integer">22</work-type-id>  
  
<external-id nil="true"/>  
</item>
```

Time entry

Get list

http://example.cennoapp.com/api/time_entries.xml

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<time-entries type="array">  
  <time-entry>  
    <billable type="boolean">>true</billable>  
    <created-at type="datetime">2009-10-02T19:51:34+03:00</created-at>  
    <date type="date">2010-09-14</date>  
    <description>Corrupti et suscipit ut provident exercitationem neque eaque et qui  
voluptatibus sit totam quis consequuntur omnis qui veniam ab sit</description>  
    <hours type="float">56.0</hours>
```

```
<id type="integer">6667</id>
<item-id type="integer">3369</item-id>
<project-id type="integer">180</project-id>
<updated-at type="datetime">2009-10-02T19:51:34+03:00</updated-at>
<user-id type="integer">478</user-id>
<external-id nil="true"/>
</time-entry>
<time-entry>
  <billable type="boolean">true</billable>
  <created-at type="datetime">2009-10-02T19:51:34+03:00</created-at>
  <date type="date">2010-08-22</date>
  <description>Modi deleniti rerum illum rerum aliquid non maxime
occaecat</description>
  <hours type="float">16.0</hours>
  <id type="integer">6665</id>
  <item-id type="integer">3367</item-id>
  <project-id type="integer">180</project-id>
  <updated-at type="datetime">2009-10-02T19:51:34+03:00</updated-at>
  <user-id type="integer">476</user-id>
  <external-id nil="true"/>
</time-entry>
```

Get one by id

http://example.cennoapp.com/api/time_entries/6665.xml

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<time-entry>
  <billable type="boolean">true</billable>
  <created-at type="datetime">2009-10-02T19:51:34+03:00</created-at>
  <date type="date">2010-08-22</date>
  <description>Modi deleniti rerum illum rerum aliquid non maxime
occaecat</description>
  <hours type="float">16.0</hours>
  <id type="integer">6665</id>
  <item-id type="integer">3367</item-id>
  <project-id type="integer">180</project-id>
```

```
<updated-at type="datetime">2009-10-02T19:51:34+03:00</updated-at>
<user-id type="integer">476</user-id>
<external-id nil="true"/>
</time-entry>
```

Create new

http://example.cennoapp.com/api/time_entries.xml

HTTP request method: **POST**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<time-entry>
  <user_id>100</user_id>
  <date>2009-10-26</date>
  <hours>7</hours>
  <project-id type="integer">180</project-id>
  <item-id type="integer">3367</item-id>
  <description>I did some work</description>
  <external-id>17</external-id>
</time-entry>
```

Example response:

Response status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<time-entry>
  <billable type="boolean">>false</billable>
  <created-at type="datetime">2009-10-26T17:50:24+02:00</created-at>
  <date type="date">2009-10-26</date>
  <description>I did some work</description>
  <hours type="float">7.0</hours>
  <id type="integer">7874</id>
  <project-id type="integer">180</project-id>
  <item-id type="integer">3367</item-id>
  <updated-at type="datetime">2009-10-26T17:50:24+02:00</updated-at>
  <user-id type="integer">100</user-id>
  <external-id>17</external-id>
</time-entry>
```

Update

http://example.cennoapp.com/api/time_entries/7874.xml

HTTP request method: **PUT**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<time-entry>
  <billable type="boolean">>false</billable>
  <created-at type="datetime">2009-10-26T17:50:24+02:00</created-at>
  <date type="date">2009-10-26</date>
  <description>I did some hard work</description>
  <hours type="float">7.0</hours>
  <id type="integer">7874</id>
  <project-id type="integer">180</project-id>
  <item-id type="integer">3367</item-id>
  <updated-at type="datetime">2009-10-26T17:50:24+02:00</updated-at>
  <user-id type="integer">100</user-id>
  <external-id>17</external-id>
</time-entry>
```

Example response:

Response status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<time-entry>
  <billable type="boolean">>false</billable>
  <created-at type="datetime">2009-10-26T17:50:24+02:00</created-at>
  <date type="date">2009-10-26</date>
  <description>I did some hard work</description>
  <external-id>17</external-id>
  <hours type="float">7.0</hours>
  <id type="integer">7874</id>
  <project-id type="integer">180</project-id>
  <item-id type="integer">3367</item-id>
  <updated-at type="datetime">2009-10-27T15:07:51+02:00</updated-at>
  <user-id type="integer">100</user-id>
</time-entry>
```

Delete

http://example.cennoapp.com/api/time_entries/7874.xml

HTTP request method: **DELETE**.

Response Status: 200

Validation errors

If validation errors would occur during creation or updating of resource they are returned in response.

Example:

Response Status: 422

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>User can't be blank</error>
  <error>Date can't be blank</error>
  <error>Hours can't be blank</error>
</errors>
```

Not found resource

If resource is not found by id, server responds with status 404.

Example:

http://example.cennoapp.com/api/time_entries/7874.xml

HTTP request method: **GET**.

Response Status: 404

User status

Get list

http://example.cennoapp.com/api/users/5/user_statuses.xml

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<user-statuses type="array">
  <user-status>
    <created-at type="datetime">2009-12-22T16:58:15+02:00</created-at>
```

```
<id type="integer">1</id>
<message>First status update</message>
<updated-at type="datetime">2009-12-22T16:58:15+02:00</updated-at>
<user-id type="integer">5</user-id>
</user-status>
<user-status>
  <created-at type="datetime">2009-12-22T16:59:44+02:00</created-at>
  <id type="integer">2</id>
  <message>Second status update</message>
  <updated-at type="datetime">2009-12-22T16:59:44+02:00</updated-at>
  <user-id type="integer">5</user-id>
</user-status>
</user-statuses>
```

Get one by id

http://example.cennoapp.com/api/users/5/user_statuses/1.xml

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<user-status>
  <created-at type="datetime">2009-12-22T16:58:15+02:00</created-at>
  <id type="integer">1</id>
  <message>First status update</message>
  <updated-at type="datetime">2009-12-22T16:58:15+02:00</updated-at>
  <user-id type="integer">5</user-id>
</user-status>
```

Create new

http://example.cennoapp.com/api/users/5/user_statuses.xml

HTTP request method: **POST**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<user-status>
  <message>Testing status update</message>
</user-status>
```

Example response:

Response status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<user-status>
  <created-at type="datetime">2010-09-15T13:28:20+03:00</created-at>
  <id type="integer">100</id>
  <message>Testing status update</message>
  <updated-at type="datetime">2010-09-15T13:28:20+03:00</updated-at>
  <user-id type="integer">5</user-id>
</user-status>
```

Update

http://example.cennoapp.com/api/users/5/user_statuses/100.xml

HTTP request method: **PUT**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<user-status>
  <message>Testing status update modification</message>
</user-status>
```

Example response:

Response status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<user-status>
  <created-at type="datetime">2010-09-15T13:28:20+03:00</created-at>
  <id type="integer">100</id>
  <message>Testing status update modification</message>
  <updated-at type="datetime">2010-09-15T13:30:49+03:00</updated-at>
  <user-id type="integer">5</user-id>
</user-status>
```

Delete

http://example.cennoapp.com/api/users/5/user_statuses/100.xml

HTTP request method: **DELETE**.

Response Status: 200

Validation errors

If validation errors would occur during creation or updating of resource they are returned in response.

Example:

Response Status: 422

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>Message can't be blank</error>
</errors>
```

Not found resource

If resource is not found by id, server responds with status 404.

Example:

http://example.cennoapp.com/api/users/4/user_statuses/10000.xml

HTTP request method: **GET**.

Response Status: 404

Project comment

Get list

<http://example.cennoapp.com/api/projects/165/comments.xml>

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<comments type="array">
  <comment>
    <comment>test</comment>
    <commentable-id type="integer">165</commentable-id>
    <commentable-type>Project</commentable-type>
    <created-at type="datetime">2009-10-19T15:03:05+03:00</created-at>
    <id type="integer">8863</id>
    <updated-at type="datetime">2009-10-19T15:03:05+03:00</updated-at>
    <user-id type="integer">5</user-id>
  </comment>
  <comment>
```

```
<comment>aaa</comment>
<commentable-id type="integer">165</commentable-id>
<commentable-type>Project</commentable-type>
<created-at type="datetime">2010-03-17T10:00:38+02:00</created-at>
<id type="integer">9130</id>
<updated-at type="datetime">2010-03-17T10:00:38+02:00</updated-at>
<user-id type="integer">5</user-id>
</comment>
</comments>
```

Get one by id

<http://example.cennoapp.com/api/projects/165/comments/8863.xml>

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>test</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Project</commentable-type>
  <created-at type="datetime">2009-10-19T15:03:05+03:00</created-at>
  <id type="integer">8863</id>
  <updated-at type="datetime">2009-10-19T15:03:05+03:00</updated-at>
  <user-id type="integer">5</user-id>
</comment>
```

Create new

<http://example.cennoapp.com/api/projects/165/comments.xml>

HTTP request method: **POST**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing new comment</comment>
  <user-id type="integer">5</user-id>
</comment>
```

Example response:

Reponse status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing new comment</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Project</commentable-type>
  <created-at type="datetime">2010-09-15T13:45:11+03:00</created-at>
  <id type="integer">9259</id>
  <updated-at type="datetime">2010-09-15T13:45:11+03:00</updated-at>
  <user-id type="integer">5</user-id>
</comment>
```

Update

<http://example.cennoapp.com/api/projects/165/comments/9259.xml>

HTTP request method: **PUT**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing comment modification</comment>
</comment>
```

Example response:

Reponse status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing comment modification</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Project</commentable-type>
  <created-at type="datetime">2010-09-15T13:45:11+03:00</created-at>
  <id type="integer">9259</id>
  <updated-at type="datetime">2010-09-15T13:47:18+03:00</updated-at>
  <user-id type="integer">5</user-id>
</comment>
```

Delete

<http://example.cennoapp.com/api/projects/165/comments/9259.xml>

HTTP request method: **DELETE**.

Response Status: 200

Validation errors

If validation errors would occur during creation or updating of resource they are returned in response.

Example:

Response Status: 422

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>User can't be blank</error>
  <error>Comment can't be blank</error>
</errors>
```

Not found resource

If resource is not found by id, server responds with status 404.

Example:

<http://example.cennoapp.com/api/projects/165/comments/92591.xml>

HTTP request method: **GET**.

Response Status: 404

Item comment

Get list

<http://example.cennoapp.com/api/items/165/comments.xml>

HTTP request method: GET.

An example of XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<comments type="array">
  <comment>
    <comment>test</comment>
    <commentable-id type="integer">165</commentable-id>
```

```
<commentable-type>Item</commentable-type>
<created-at type="datetime">2009-10-19T15:03:05+03:00</created-at>
<id type="integer">8863</id>
<updated-at type="datetime">2009-10-19T15:03:05+03:00</updated-at>
<user-id type="integer">5</user-id>
</comment>
<comment>
  <comment>aaa</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Item</commentable-type>
  <created-at type="datetime">2010-03-17T10:00:38+02:00</created-at>
  <id type="integer">9130</id>
  <updated-at type="datetime">2010-03-17T10:00:38+02:00</updated-at>
  <user-id type="integer">5</user-id>
</comment>
</comments>
```

Get one by id

<http://example.cennoapp.com/api/items/165/comments/8863.xml>

HTTP request method: GET.

Example response:

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>test</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Item</commentable-type>
  <created-at type="datetime">2009-10-19T15:03:05+03:00</created-at>
  <id type="integer">8863</id>
  <updated-at type="datetime">2009-10-19T15:03:05+03:00</updated-at>
  <user-id type="integer">5</user-id>
</comment>
```

Create new

<http://example.cennoapp.com/api/items/165/comments.xml>

HTTP request method: **POST**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing new comment</comment>
  <user-id type="integer">5</user-id>
</comment>
```

Example response:

Reponse status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing new comment</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Item</commentable-type>
  <created-at type="datetime">2010-09-15T13:45:11+03:00</created-at>
  <id type="integer">9259</id>
  <updated-at type="datetime">2010-09-15T13:45:11+03:00</updated-at>
  <user-id type="integer">5</user-id>
</comment>
```

Update

<http://example.cennoapp.com/api/items/165/comments/9259.xml>

HTTP request method: **PUT**.

Example request:

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing comment modification</comment>
</comment>
```

Example response:

Reponse status: 201

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <comment>Testing comment modification</comment>
  <commentable-id type="integer">165</commentable-id>
  <commentable-type>Item</commentable-type>
```

```
<created-at type="datetime">2010-09-15T13:45:11+03:00</created-at>
<id type="integer">9259</id>
<updated-at type="datetime">2010-09-15T13:47:18+03:00</updated-at>
<user-id type="integer">5</user-id>
</comment>
```

Delete

<http://example.cennoapp.com/api/items/165/comments/9259.xml>

HTTP request method: **DELETE**.

Response Status: 200

Validation errors

If validation errors would occur during creation or updating of resource they are returned in response.

Example:

Response Status: 422

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>User can't be blank</error>
  <error>Comment can't be blank</error>
</errors>
```

Not found resource

If resource is not found by id, server responds with status 404.

Example:

<http://example.cennoapp.com/api/items/165/comments/92591.xml>

HTTP request method: **GET**.

Response Status: 404

Searching resources

Listing interfaces allow resources searching/filtering by parameters.

Note: "-" should be replaced to "_" in field names (*company-id* becomes *company_id*)

Example 1.

Finding all projects by company ID:

`http://example.cennoapp.com/api/projects.xml?search[company_id]=220`

Example 2.

Finding all projects by company ID and status ID:

`http://example.cennoapp.com/api/projects.xml?search[company_id]=220&search[status_id]=17`

Example 3.

Finding all projects by external ID:

`http://example.cennoapp.com/api/projects.xml?search[external_id]=SOMEID`

Advanced searching

Supported advanced searching:

field_like - search for substring

field_not_like - search for values not containing substring

field_begin_with – search for values beginning with substring

field_not_begin_with – search for values not beginning with substring

field_end_with – search for values ending with substring

field_not_end_with – search for values not ending with substring

field_null – search for NULL

field_not_null – search for NOT NULL

field_blank – search for NULL or empty values

field_not_blank – search for NOT NULL or non-empty values

field_does_not_equal (alias: **field_not**) – search for values not equal to value

field_less_than (alias: **field_lt**) – search for values less than value

field_less_than_or_equal_to (alias: **field_lte**) – search for values less than or equal to value

field_greater_than (alias: **field_gt**) – search for values greater than value

field_greater_than_or_equal_to (alias: **field_gte**) – search for values greater than or equal to value

Example 1.

Finding all projects whose title contains “alfa”:

`http://example.cennoapp.com/api/projects.xml?search[title_like]=alfa`

Example 2.

Finding all projects whose external id is empty:

[http://example.cennoapp.com/api/projects.xml?search\[external_id_blank\]=1](http://example.cennoapp.com/api/projects.xml?search[external_id_blank]=1)

Example 3.

Combining conditions:

[http://example.cennoapp.com/api/projects.xml?search\[external_id_blank\]=1&search\[title_like\]=alfa](http://example.cennoapp.com/api/projects.xml?search[external_id_blank]=1&search[title_like]=alfa)

Searching using associations

API allows searching resources by associated values.

Example 1:

Search all items belonging to project with external_id SOMEID.

[http://example.cennoapp.com/api/items.xml?search\[project_external_id_eq\]=SOMEID](http://example.cennoapp.com/api/items.xml?search[project_external_id_eq]=SOMEID)

Example 2:

Search all items belonging to projects with empty external ids.

[http://example.cennoapp.com/api/items.xml?search\[project_external_id_empty\]=1](http://example.cennoapp.com/api/items.xml?search[project_external_id_empty]=1)