

SKYRIX<sup>®</sup> Software AG

# SKYRIX<sup>®</sup> Web Groupware

*XML-RPC Handbook*

Version 4.1

*31. Juli 2003*



This version is optimized for on-screen viewing. If you don't want the hyperlinks and the colored hints when printing this book, get the specially optimized print version. The content of both versions is identical.

# Inhaltsverzeichnis

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>Miscellaneous</b>	<b>11</b>
2.1	Working with date and time values . . . . .	11
<b>3</b>	<b>Data type overview</b>	<b>13</b>
3.1	Appointment Document . . . . .	13
3.2	Job Document . . . . .	15
3.3	Account Document . . . . .	17
3.4	Team Document . . . . .	18
3.5	Person Document . . . . .	19
<b>4</b>	<b>Package: account</b>	<b>23</b>
4.1	Method:getByNumber . . . . .	24
4.2	Method:fetchIds . . . . .	25
4.3	Method:passwordForLogin . . . . .	26
4.4	Method:getTeamsForLogin . . . . .	27
4.5	Method:getByLogin . . . . .	28
4.6	Method:deleteByNumber . . . . .	29
4.7	Method:deleteByLogin . . . . .	30
4.8	Method:fetch . . . . .	31
4.9	Method:insert . . . . .	32
4.10	Method:update . . . . .	33
4.11	Method:delete . . . . .	34
4.12	Method:getAllTemplateUserLogins . . . . .	35

4.13 Method:setPassword . . . . .	36
4.14 Method:updatePassword . . . . .	38
4.15 Method:getById . . . . .	40
4.16 Method:getLoginAccount . . . . .	41
4.17 Method:getLoginAccountId . . . . .	42
<b>5 Package: appointment</b>	<b>43</b>
5.1 Method:fetch . . . . .	44
5.2 Method:setParticipants . . . . .	45
5.3 Method:delete . . . . .	46
5.4 Method:fetchIds . . . . .	47
5.5 Method:getById . . . . .	48
5.6 Method:insert . . . . .	50
5.7 Method:update . . . . .	51
<b>6 Package: enterprise</b>	<b>53</b>
6.1 Method:getByNumber . . . . .	54
6.2 Method:deleteByNumber . . . . .	56
6.3 Method:fetchIds . . . . .	58
6.4 Method:fetch . . . . .	59
6.5 Method:insert . . . . .	60
6.6 Method:update . . . . .	61
6.7 Method:delete . . . . .	62
6.8 Method:get . . . . .	64
6.9 Method:getPersons . . . . .	66
6.10 Method:fetchPersons . . . . .	68
6.11 Method:getProjects . . . . .	70
6.12 Method:deletePerson . . . . .	72
6.13 Method:getById . . . . .	75
6.14 Method:insertPerson . . . . .	76

<b>7</b>	<b>Package: jobs</b>	<b>77</b>
7.1	Method:markDone . . . . .	78
7.2	Method:archiveJob . . . . .	80
7.3	Method:acceptJob . . . . .	82
7.4	Method:annotateJob . . . . .	84
7.5	Method:rejectJob . . . . .	85
7.6	Method:reactivateJob . . . . .	87
7.7	Method:getToDoList . . . . .	89
7.8	Method:getDelegatedJobs . . . . .	91
7.9	Method:getArchivedJobs . . . . .	93
7.10	Method:getJobs . . . . .	95
7.11	Method:getHistory . . . . .	96
7.12	Method:assignToProject . . . . .	97
7.13	Method:detachFromProject . . . . .	99
7.14	Method:getProjects . . . . .	101
7.15	Method:fetchJobs . . . . .	103
7.16	Method:fetchJobIds . . . . .	104
7.17	Method:insertJob . . . . .	105
7.18	Method:updateJob . . . . .	106
7.19	Method:deleteJob . . . . .	107
<b>8</b>	<b>Package: mail</b>	<b>109</b>
8.1	Method:createMail . . . . .	110
8.2	Method:setFrom . . . . .	111
8.3	Method:addTo . . . . .	112
8.4	Method:setTo . . . . .	113
8.5	Method:addCC . . . . .	114
8.6	Method:setCC . . . . .	115
8.7	Method:addBCC . . . . .	116
8.8	Method:setBCC . . . . .	117
8.9	Method:setSubject . . . . .	118

8.10 Method:addHeaderValue . . . . .	119
8.11 Method:setHeaderValue . . . . .	120
8.12 Method:addHeaders . . . . .	121
8.13 Method:setHeaders . . . . .	122
8.14 Method:setTextBody . . . . .	123
8.15 Method:setHtmlBody . . . . .	124
8.16 Method:addAttachment . . . . .	125
8.17 Method:send . . . . .	126
8.18 Method:getTransactions . . . . .	127
8.19 Method:deleteAllTransactions . . . . .	128
8.20 Method:closeTransaction . . . . .	129
<b>9 Package: person</b>	<b>131</b>
9.1 Method:getById . . . . .	132
9.2 Method:getByNumber . . . . .	133
9.3 Method:deleteByNumber . . . . .	134
9.4 Method:fetchIds . . . . .	135
9.5 Method:fetch . . . . .	136
9.6 Method:insert . . . . .	137
9.7 Method:update . . . . .	138
9.8 Method:delete . . . . .	139
9.9 Method:getEnterprises . . . . .	140
9.10 Method:fetchEnterprises . . . . .	141
9.11 Method:insertEnterprise . . . . .	142
9.12 Method:deleteEnterprise . . . . .	143
9.13 Method:get . . . . .	144
9.14 Method:fetchJobs . . . . .	145
9.15 Method:fetchJobIds . . . . .	146
9.16 Method:insertJob . . . . .	147
9.17 Method:updateJob . . . . .	148
9.18 Method:deleteJob . . . . .	149

<b>10 Package: project</b>	<b>151</b>
10.1 Method:fetch . . . . .	152
10.2 Method:insert . . . . .	154
10.3 Method:update . . . . .	155
10.4 Method:delete . . . . .	156
10.5 Method:cwd . . . . .	157
10.6 Method:cd . . . . .	158
10.7 Method:ls . . . . .	159
10.8 Method:mkdir . . . . .	161
10.9 Method:rmdir . . . . .	163
10.10 Method:attr . . . . .	165
10.11 Method:zip . . . . .	167
10.12 Method:rm . . . . .	169
10.13 Method:cp . . . . .	171
10.14 Method:mv . . . . .	173
10.15 Method:ln . . . . .	175
10.16 Method:exists . . . . .	177
10.17 Method:isdir . . . . .	179
10.18 Method:islink . . . . .	181
10.19 Method:flush . . . . .	183
10.20 Method:newDocument . . . . .	184
10.21 Method:loadDocument . . . . .	185
10.22 Method:saveDocument . . . . .	186
<b>11 Package: resource</b>	<b>187</b>
11.1 Method:insert . . . . .	188
11.2 Method:getById . . . . .	189
11.3 Method:getByName . . . . .	190
11.4 Method:update . . . . .	191
11.5 Method:delete . . . . .	192
11.6 Method:categories . . . . .	193

11.7 Method:fetch . . . . .	195
<b>12 Package: system</b>	<b>197</b>
12.1 Method:getHostName . . . . .	198
12.2 Method:getServerTimeZone . . . . .	199
12.3 Method:getServerTime . . . . .	200
12.4 Method:listMethods . . . . .	201
12.5 Method:methodSignature . . . . .	202
12.6 Method:methodHelp . . . . .	203
<b>13 Package: team</b>	<b>205</b>
13.1 Method:getTeams . . . . .	206
13.2 Method:getByLogin . . . . .	207
13.3 Method:getByNumber . . . . .	208
13.4 Method:getMembersForTeam . . . . .	209
13.5 Method:insert . . . . .	210
13.6 Method:setMembers . . . . .	211



# 1 Introduction

The XML-RPC server which was included in SKYRiX 4.1 for the first time, replaces the Python bridge which was used until version 4.0. By using XML-RPC (XML Remote Procedure Call), which is based on open standards, a higher interoperability is provided. It's now possible to access SKYRiX from every language where an XML-RPC implementation is available (which applies to nearly all programming and script languages used today).

XML-RPC uses HTTP for the transport, the real data is encoded (as the name suggests) in an XML document. By using a clear design, which nevertheless allows you to transport complex data structures, XML-RPC provides an easy to use and flexible format to connect external programs to SKYRiX.

This handbook describes the functions provided by the SKYRiX XML-RPC server. It shows the arguments and return values as well as the description of the functionality of each function and includes examples written in Python.

**Note:** The Python examples used herein are using a specially adapted version of the Python XML-RPC library *xmlrpc.lib.py*. This library is officially integrated in Python since version 2.2, but doesn't support HTTP basic authentication which is needed by the SKYRiX XML-RPC daemon. It also can't handle data conversion to unicode, which can cause trouble when you are working with german umlauts in records. The SKYRiX<sup>®</sup> Software AG therefore provides a new version of this library which includes the needed features. It can be downloaded from [SKYRiX developer site](#), but you can also find it in your local SKYRiX installation in the *Tools* directory.

Up-to-date information concerning the SKYRiX XML-RPC Daemon as well as many examples for interacting with this server can be found at <http://developer.skyrix.com>.



## 2 Miscellaneous

### 2.1 Working with date and time values

XML-RPC uses *dateTime.iso8601* for displaying date and time values, which formats those values according to ISO 8601. Unfortunately this display format doesn't support timezones, so the SKYRiX XML-RPC server expects all time values in GMT/UTC. When you create scripts and programs which work together with the XML-RPC server, keep in mind to convert values in your local time zone to GMT when sending them to your server and vica versa.

XML-RPC uses the format "YYYYMMDDTHH:MM:SS" for date/time values, so January, 1st 2003 8:30pm GMT becomes "20030101T20:30:00". The XML-RPC implementations of the different programming languages contain own types for handling dates in most cases - the following example will show how to work with such a date object in Python.

```
#!/usr/bin/env python

import xmlrpclib, time

# as string
dateString = "20030101T20:30:00"

# creating a XML-RPC DateTime object
dateTime = xmlrpclib.DateTime(time.mktime(
    time.strptime(dateString, "%Y%m%dT%H:%M:%S")))

# converting back
newDateString = time.strftime("%Y%m%dT%H:%M:%S",
    time.strptime(dateTime.value,
        "%Y%m%dT%H:%M:%S"))
```



## 3 Data type overview

This part of the handbook should introduce the reader in the so-called 'documents', which are used in the return values of several functions of the SKYRiX XML-RPC server. This document will take a look at their structure and tries to show the connection between the elements of these documents and the SKYRiX web interface.

The examples used herein are written in the notation used in the Python programming language (i.e. [] are arrays, are associative arrays/hashtables/dictionaries).

### 3.1 Appointment Document

#### Structure

```
{'id': 'skyrix://skyrix.com/sx_skyrix/319748',
 'startDate': <DateTime 20030224T21:23:00 at 81e44321>,
 'endDate': <DateTime 20030225T23:59:59 at 81e8174>,
 'owner': {'id': 'skyrix://skyrix.com/sx_skyrix/122282',
           'login': 'foo'},
 'participants': [{'companyId': 122282,
                   'dbStatus': 'updated',
                   'firstname': 'Foo',
                   'isAccount': 1,
                   'isExtraAccount': 0,
                   'isIntraAccount': 1,
                   'isLocked': 0,
                   'isPerson': 1,
                   'isPrivate': 0,
                   'isReadonly': 0,
                   'keywords': 'Kunde',
                   'login': 'foo',
                   'name': 'Bar',
```

```

'number': 'SKY122282',
'objectVersion': 28,
'ownerId': 10000,
'password': 'gOjdsdsadkTE',
'templateUserId': 9999}}}
```

### Description

Name	Data type	Description
id	String	unique ID
startDate	DateTime	start date
endDate	DateTime	end date
owner	Account (3.3)	creator
participants	Person (3.5)	participants
type	String	type (optional)
cycleEndDate	DateTime	end date of a appt. series (optional)
location	String	location

### Field Association

Name	German	English
id	-	-
startDate	Beginn	Start
endDate	Ende	End
owner	Ersteller	Creator
participants	Teilnehmer	Participants
type	Zyklus	Type
cycleEndDate	Zyklusende	Cycle End Date
location	Ort	Location

## 3.2 Job Document

### Structure

```
{'category': 'todo',
 'creator': {'id': 'skyrix://skyrix.com/sx_skyrix/10000',
             'login': 'root'},
 'endDate': <DateTime 20030225T23:59:59 at 81e8174>,
 'executor': {'id': 'skyrix://skyrix.com/sx_skyrix/10000',
              'login': 'root'},
 'id': 'skyrix://skyrix.com/sx_skyrix/10150',
 'isTeamJob': <Boolean False at 81a2e84>,
 'keywords': 'todo',
 'name': 'Todo',
 'objectVersion': 1,
 'priority': 3,
 'startDate': <DateTime 20030218T00:00:00 at 821219c>,
 'status': '00_created' }
```

### Description

Name	Data type	Description
id	String	unique ID
objectVersion	Integer	object version
category	String	job category
creator	Account (3.3)	creator
executor	Account (3.3)	executant
name	String	job name
startDate	DateTime	start date
endDate	DateTime	end date
priority	Integer	priority
keywords	String	keywords
status	String	current job state
isTeamJob	Boolean	normal or team job

**Field Association**

Name	German	English
id	-	-
objectVersion	-	-
category	Kategorie	Kategorie
creator	Ersteller	Creator
executor	Bearbeiter	Executant
name	Name	Name
startDate	Startdatum	Start date
endDate	Enddatum	End date
priority	Priorität	Priority
keywords	Suchworte	Keywords
status	Status	Status
isTeamJob	-	-



### 3.3 Account Document

#### Structure

```
{'login': 'root',  
  'id': 'skyrix://skyrix.com/sx_skyrix/10000'}
```

#### Description

Name	Data type	Description
id	String	unique ID
login	String	login

#### Field Association

Name	German	English
id	-	-
login	Login	Login

## 3.4 Team Document

### Structure

```
{'description': 'all intranet',
  'email': '',
  'id': 'skyrix://skyrix.com/sx_skyrix/10003',
  'isLocationTeam': '',
  'objectVersion': 1,
  'ownerId': ''}
```

### Description

Name	Data type	Description
id	String	unique ID
objectVersion	Integer	object version
description	String	description
email	String	email address
isLocationTeam	String	is team a location team
ownerId	String	ID of the owner (creator)

### Field Association

Name	German	English
id	-	-
objectVersion	-	-
description	Name	Name
email	Email	eMail
isLocationTeam	Standort	Location Team
ownerId	-	-

### 3.5 Person Document

#### Structure

```
{'addresses': {
  'location': {'city': 'City Location',
               'country': 'Country Location',
               'id': 'skyrix://skyrix.com/sx_skyrix/10800',
               'name1': 'Name Location 1',
               'name2': 'Name Location 2',
               'name3': 'Name Location 3',
               'state': 'State Location',
               'street': 'Street Location',
               'type': 'location',
               'zip': 'ZIP Location'},
  'mailing': {'city': 'City Mailing',
              'country': 'Country Mailing',
              'id': 'skyrix://skyrix.com/sx_skyrix/10790',
              'name1': 'Name Mailing 1',
              'name2': 'Name Mailing 2',
              'name3': 'Name Mailing 3',
              'state': 'State Mailing',
              'street': 'Street Mailing',
              'type': 'mailing',
              'zip': 'ZIP Mailing'},
  'private': {'city': 'City Private ',
              'country': 'Country Private',
              'id': 'skyrix://skyrix.com/sx_skyrix/10780',
              'name1': 'Name Private 1',
              'name2': 'Name Private 2',
              'name3': 'Name Private 3',
              'state': 'State Private',
              'street': 'Street Private',
              'type': 'private',
              'zip': 'ZIP Private'}
},
'birthday': <DateTime 19700101T01:00:00 at 822476c>,
'comment': 'Just A comment',
'contact': {'id': 'skyrix://skyrix.com/sx_skyrix/10360',
            'login': 'contact'},
'degree': 'Ph.D.',
```

```
'extendedAttrs': {'email1': 'luser@example.com',
                  'email2': 'joeuser@example.com',
                  'job_title': 'CEO',
                  'other_title1': 'Marketing',
                  'other_title2': 'Support'},
'extendedKeys': ['email1', 'email2', 'job_title',
                 'other_title1', 'other_title2'],
'firstname': 'Larry',
'gender': 'male',
'id': 'skyrix://skyrix.com/sx_skyrix/10710',
'isAccount': <Boolean False at 81a2ed4>,
'isPrivate': <Boolean False at 81a2ed4>,
'keywords': 'foobar',
'login': 'SKY10710',
'middlename': 'M.',
'name': 'User',
'nickname': 'luser',
'number': 'SKY10710',
'objectVersion': 4,
'owner': {'id': 'skyrix://skyrix.com/sx_skyrix/10000',
          'login': 'root'},
'phoneTypes': ['01_tel', '02_tel', '03_tel_funk', '05_tel_private',
               '10_fax', '15_fax_private'],
'phones': {
  '01_tel': {'info': 'Phone 1',
             'number': '555-1234',
             'telephoneId': 10860,
             'type': '01_tel'},
},
'02_tel': {'info': 'Phone 2',
           'number': '555-1235',
           'telephoneId': 10830,
           'type': '02_tel'},
},
'03_tel_funk': {'info': 'Mobile',
                'number': '555-1236',
                'telephoneId': 10850,
                'type': '03_tel_funk'},
},
'05_tel_private': {'info': 'Private',
                  'number': '555-1237',
```

### 3.5 Person Document

---

```
        'telephoneId': 10840,  
        'type': '05_tel_private'},  
    '10_fax': {'info': 'Fax',  
        'number': '555-1238',  
        'telephoneId': 10820,  
        'type': '10_fax'},  
    '15_fax_private': {'info': 'Fax Private',  
        'number': '555-1239',  
        'telephoneId': 10810,  
        'type': '15_fax_private'}  
},  
'salutation': '03_dear_mr',  
'url': 'http://example.com'}
```

#### Feldbeschreibung

Name	Data type	Description
id	String	unique ID
objectVersion	Int	object version
addresses	Struct	addresses
birthday	String	birthday
comment	String	comment
contact	Account	associated contact
extendedAttrs	Struct	extended attributes
extendedKeys	Array	keys of extended attributes
firstname	String	firstname
gender	String	gender
isAccount	Boolean	is account or just a normal person record
isPrivate	Boolean	is read access set private
keywords	String	categories
login	String	login
middlename	String	middle name
name	String	surname
nickname	String	nickname
number	String	unique number
owner	Account	owner (i.e. creator)
phoneTypes	Array	telephone types
phones	Struct	phones
salutation	String	salutation
url	String	url

**Field Association**

Name	German	English
id	-	-
objectVersion	-	-
addresses	Adressen	Adresses
birthday	Geburtstag	Birthday
comment	Kommentar	Commeent
contact	Kontakt	Contact
extendedAttrs	-	-
extendedKeys	erweiterte Attribute	Extended Attributes
firstname	Vorname	Firstname
gender	Geschlecht	Gender
isAccount	-	-
isComplete	-	-
isPrivate	Privat	Private
keywords	Kategorien	Categories
login	-	-
middlename	Mitteldname	Middlename
name	Nachname	Name
nickname	Spitzname	Nickname
number	-	-
owner	Eigentümer	Owner
phoneTypes	-	-
phones	-	-
salutation	Anrede	Salutation
url	URL	URL

## **4 Package: account**

## 4.1 Method: getByNumber

### Documentation

None

---

### Arguments

accountnumber	string	unique account number
---------------	--------	-----------------------

### Result

struct	Account document
--------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.getByNumber("SKY122282")
```

Requests the account document with the account number "SKY122282".

---



## 4.2 Method:fetchIds

### Documentation

None

---

### Arguments

fspec	struct	fetch specification
-------	--------	---------------------

### Result

array	array of Account Ids
-------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.fetchIds("name like '*'")
```

Requests the IDs of all accounts.

---

## 4.3 Method:passwordForLogin

### Documentation

None

---

### Arguments

login	string	login
-------	--------	-------

### Result

string	password
--------	----------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.passwordForLogin('mike')
```

Requests the crypted password of user 'mike'.

---

# 4.4 Method:getTeamsForLogin

**Documentation**

None

---

**Arguments**

login	string	login
-------	--------	-------

**Result**

array	Array of team documents
-------	-------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.getTeamsForLogin('mike')
```

Requests the teams for user 'mike'.

---

## 4.5 Method: getByLogin

### Documentation

None

---

### Arguments

login	string	Login name of an account.
-------	--------	---------------------------

### Result

struct	Account document
--------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.getByLogin("mike")
```

Requests the account document for the login "mike".

---

## 4.6 Method:deleteByNumber

### Documentation

None

---

### Arguments

accountnumber	string	Number of an account.
---------------	--------	-----------------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.deleteByNumber("SKY122282")
```

Deletes the account with the account number "SKY122282".

---

## 4.7 Method:deleteByLogin

### Documentation

None

---

### Arguments

loginname	string	Login name of an account.
-----------	--------	---------------------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.deleteByLogin("mike")
```

Deletes the account with the login "mike".

---

## 4.8 Method:fetch

### Documentation

None

---

### Arguments

fspec	struct	fetch specification
-------	--------	---------------------

### Result

array	Array of account documents.
-------	-----------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.fetch(
    {
        'qualifier'      : "login = 'mike'",
        'sortOrdering'   : ["login",],
    }
)
```

Requests all accounts with a login equal to 'mike', sorted by the login.

---

## 4.9 Method:insert

### Documentation

None

---

### Arguments

dict	struct	dictionary
------	--------	------------

### Result

struct	This method returns the new account document including the account number.
--------	--

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.insert(
    {
        'login'      : "pmeier",
        'firstname'   : "Peter",
        'name'        : "Meier",
    }
)
```

Inserts a new account with the login "pmeier" for a person called "Peter Meier".

---



## 4.10 Method:update

### Documentation

None

---

### Arguments

accountdoc	struct	AccountDocument. Must contain the account number.
------------	--------	---

### Result

	struct	The updated AccountDocument.
--	--------	------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
account = server.account.getByLogin('pmeier')
account['name'] = "Maier"
result = server.account.update(account)
```

Updates the account with the login "pmeier", changes only the attributes "name".

---

## 4.11 Method:delete

### Documentation

None

---

### Arguments

accountdoc	struct	AccountDocument
------------	--------	-----------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.delete(
    {
        'number' : "122282",
    }
)
```

Deletes the account with the number "122282".

---

## 4.12 Method:getAllTemplateUserLogins

### Documentation

None

---

### Arguments

no args

### Result

array	Array of logins
-------	-----------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.getAllTemplateUserLogins()
```

Gets the logins of all template users.

---

## 4.13 Method:setPassword

### Documentation

None

---

### Arguments

id	string	unique account ID
password	string	new password

### Result

boolean	success indicator
---------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.setPassword('11324','newpwd')
```

Sets the password of the user with ID '11324' to 'newpwd'.

---

### Arguments

id	string	unique account ID
password	string	new password
crypted	boolean	crypted password

### Result

boolean	success indicator
---------	-------------------

##### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.setPassword('11324','Xf345Fsds',1)
```

Sets the password of the user with ID '11324', assuming the given password is already crypted.

---

## 4.14 Method:updatePassword

### Documentation

None

---

### Arguments

password	string	new password
----------	--------	--------------

### Result

boolean	success indicator
---------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.updatePassword('newpwd')
```

Sets the password for the current user to 'newpwd'.

---

### Arguments

password	string	new password
crypted	boolean	crypted password

### Result

boolean	success indicator
---------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.updatePassword('Xf345Fsds',1)
```

#### 4.14 Method:updatePassword

---

Sets the password of the current user, assuming the given password is already crypted.

---

## 4.15 Method: getById

### Documentation

None

---

### Arguments

id	string	unique account ID
----	--------	-------------------

### Result

struct	account document
--------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.getById("11342")
```

Requests the account document with the account id "11342".

---



## 4.16 Method:getLoginAccount

### Documentation

None

---

### Arguments

no args

### Result

struct	account document
--------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.account.getLoginAccount()
```

Get the currently logged in account.

---

## 4.17 Method: getLoginAccountId

### Documentation

None

---

### Arguments

no args

### Result

id	string	unique account ID
----	--------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.account.getLoginAccountId()
```

Get the currently logged in account ID.

---

## **5 Package: appointment**

## 5.1 Method:fetch

### Documentation

None

---

### Arguments

fspec	struct	fetch specification
-------	--------	---------------------

### Result

array	Array of appointment documents.
-------	---------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

import xmlrpclib, time
fSpec = {}
startDate = xmlrpclib.DateTime(time.strptime(
    '2002-01-01', '%Y-%m-%d'))
endDate = xmlrpclib.DateTime(time.strptime(
    '2002-12-01', '%Y-%m-%d'))

fSpec['qualifier'] = {}
fSpec['qualifier']['startDate'] = startDate
fSpec['qualifier']['endDate'] = endDate
result = server.appointment.fetch(fSpec)
```

Request all appointments between January 1st and January, 7th 2002

---

## 5.2 Method:setParticipants

### Documentation

None

---

### Arguments

app	struct	AppointmentDocument
part	array	Array of person- or team documents

### Result

struct	Appointment document
--------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
appointment = server.appointment.getById("312358")
person = server.person.fetch("name like '*beeblebrox*")[0]
result = server.appointment.setParticipants(appointment, [person,])
```

Sets the persons with name like "Beeblebrox" as participants for the appointment with ID "312358".

---

## 5.3 Method:delete

### Documentation

None

---

### Arguments

app	struct	appointment document
-----	--------	----------------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

import xmlrpclib, time
startDate = xmlrpclib.DateTime(time.strptime(
                                "2002-10-01", "%Y-%m-%d"))
endDate = xmlrpclib.DateTime(time.strptime(
                              "2002-10-02", "%Y-%m-%d"))

dict = {}
dict['startDate'] = startDate
dict['endDate'] = endDate
dict['name'] = 'Exhibition'
appointment = server.appointment.insert(dict)
result = server.appointment.delete(appointment)
```

Deletes the appointment.

---

## 5.4 Method:fetchIds

### Documentation

None

---

### Arguments

fspec	struct	fetch specification
-------	--------	---------------------

### Result

array	Array of appointment IDs.
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

import xmlrpclib, time
fSpec = {}
startDate = xmlrpclib.DateTime(time.strptime(
    '2002-01-01', '%Y-%m-%d'))
endDate = xmlrpclib.DateTime(time.strptime(
    '2002-01-31', '%Y-%m-%d'))

fSpec['qualifier'] = {}
fSpec['qualifier']['startDate'] = startDate
fSpec['qualifier']['endDate'] = endDate
result = server.appointment.fetchIds(fSpec)
```

Request the appointment IDs of the appointments in January 2002.

---

## 5.5 Method: getById

### Documentation

None

---

### Arguments

id	string	unique document id
----	--------	--------------------

### Result

struct	appointment document
--------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.appointment.getById(["312358", "312353"])
```

Request the appointments with the IDs "312358" and "312353".

---

### Arguments

id	array	array of unique appointment ids
----	-------	---------------------------------

### Result

array	array of appointment documents
-------	--------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.appointment.getById("312358")
```



## 5.5 Method:getByld

---

Request the appointment with the ID "312358".

---

## 5.6 Method:insert

### Documentation

None

---

### Arguments

dict	struct	dictionary
------	--------	------------

### Result

struct	appointment document
--------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

import xmlrpclib, time
startDate = xmlrpclib.DateTime(time.strptime(
    "2002-10-01", "%Y-%m-%d"))
endDate = xmlrpclib.DateTime(time.strptime(
    "2002-10-02", "%Y-%m-%d"))

dict = {}
dict['startDate'] = startDate
dict['endDate'] = endDate
dict['name'] = 'Exhibition'
result = server.appointment.insert(dict)
```

Inserts the new appointment with the name "Exhibition".

---

XMLRPC_FAULT_MISSING_PARAMETER Missing 'startDate' attribute in appointment
XMLRPC_FAULT_MISSING_PARAMETER Missing 'title' attribute in appointment

## 5.7 Method:update

### Documentation

None

---

### Arguments

document	struct	appointment document
----------	--------	----------------------

### Result

struct	appointment document
--------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
appointment = server.appointment.getById("312358")
appointment['name'] = 'Changed'
result = server.appointment.update(appointment)
```

Updates the appointment with the ID "312358", sets the name to "Changed".

---



## **6 Package: enterprise**

## 6.1 Method: getByNumber

### Documentation

None

---

### Arguments

number	int	Unique number of a enterprise
--------	-----	-------------------------------

### Result

struct	EnterpriseDocument
--------	--------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.getByNumber(10003)
```

Request the enterprise document with the enterprise number "10003".

---

### Arguments

number	string	unique enterprise number
--------	--------	--------------------------

### Result

struct	Unternehmensdokument
--------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.getByNumber("145128")
```

## 6.1 Method:getByNumber

---

Request the enterprise document with the enterprise number "145128".

---

## 6.2 Method:deleteByNumber

### Documentation

None

---

### Arguments

number	int	Unique number of a enterprise
--------	-----	-------------------------------

### Result

struct	The deleted document
--------	----------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

dict = {}
dict['description'] = 'Enterprise'
enterprise = server.enterprise.insert(dict)
result = server.enterprise.deleteByNumber(enterprise['number'])
```

Deletes the enterprise.

---

### Arguments

number	string	Unique number of a enterprise
--------	--------	-------------------------------

### Result

struct	The deleted document
--------	----------------------



### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

dict = {}
dict['description'] = "Enterprise"
enterprise = server.enterprise.insert(dict)
result = server.enterprise.deleteByNumber(enterprise['number'])
```

Deletes the enterprise.

---

## 6.3 Method:fetchIds

### Documentation

None

---

### Arguments

fSpec	struct	fetch specification
-------	--------	---------------------

### Result

array	Array of enterprise IDs
-------	-------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

fSpec = {}
fSpec['qualifier'] = "description like '*skyrix*"
result = server.enterprise.fetchIds(fSpec)
```

Fetches all enterprise ids for enterprises with a description attribute containing the string "Skyrix".

---

## 6.4 Method:fetch

### Documentation

None

---

### Arguments

fSpec	struct	fetch specification
-------	--------	---------------------

### Result

array	array of enterprise documents
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

fSpec = {}
fSpec['qualifier'] = "description like '*skyrix'"
fSpec['sortOrderings'] = ['description',]
result = server.enterprise.fetch(fSpec)
```

Fetches all enterprise documents with the name attribute equal to the string "Skyrix", result is sorted by the attribute "description".

---

## 6.5 Method:insert

### Documentation

None

---

### Arguments

dict	struct	dictionary
------	--------	------------

### Result

struct	the new enterprise document including the enterprise number.
--------	--

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

dict = {}
dict['description'] = "New Company";
dict['url'] = 'http://www.new-compa.ny'
result = server.enterprise.insert(dict)
```

Creates a new enterprise document with the given attributes.

---

# 6.6 Method:update

## Documentation

None

---

## Arguments

doc	struct	enterprise document
-----	--------	---------------------

## Result

struct	the updated enterprise document.
--------	----------------------------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
enterprise = server.enterprise.fetch(
    "description like '*skyrix*")[0]
enterprise['url'] = 'www.skyrix.minor'
result = server.enterprise.update(enterprise)
```

Updates the enterprise document.

---

## 6.7 Method:delete

### Documentation

None

---

### Arguments

number	string	unique enterprise number
--------	--------	--------------------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

dict = {}
dict = {}
dict['description'] = 'My Enterprise'
enterprise = server.enterprise.insert(dict)
result = server.enterprise.delete(enterprise['number'])
```

Deletes the enterprise.

---

### Arguments

number	enterprise document, has to contain the enterprise number
--------	---

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

dict = {}
dict['description'] = 'My Enterprise'
enterprise = server.enterprise.insert(dict)
result = server.enterprise.delete(enterprise)
```

Deletes the enterprise.

---

## 6.8 Method: get

### Documentation

None

---

### Arguments

number	int	unique enterprise number
--------	-----	--------------------------

### Result

array	array of enterprise documents
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.get(10003)
```

Gets the enterprise with the enterprise number "10003".

---

### Arguments

number	struct	enterprise document
--------	--------	---------------------

### Result

array	array of enterprise documents
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
```



## 6.8 Method:get

---

```
                                login='test', password='system')
result = server.enterprise.get(
    {
        'description' : "*skyrix*",
    }
)
```

Gets the enterprises with the description 'skyrix'.

---

### Arguments

number	string	unique enterprise number
--------	--------	--------------------------

### Result

array	array of enterprise documents
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.enterprise.get("10003")
```

Gets the enterprises with the enterprise number "10003".

---

## 6.9 Method: getPersons

### Documentation

None

---

### Arguments

number	int	unique enterprise number
--------	-----	--------------------------

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.getPersons(10003)
```

Gets the persons belonging to the enterprise matching the given attributes.

---

### Arguments

number	struct	enterprise document
--------	--------	---------------------

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
```

## 6.9 Method: getPersons

---

```
        login='test', password='system')
enterprise = server.enterprise.fetch(
    "description like '*skyrix*')[0]
result = server.enterprise.getPersons(enterprise)
```

Gets the persons belonging to the enterprise matching the given attributes.

---

### Arguments

number	string	unique enterprise number
--------	--------	--------------------------

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
    login='test', password='system')
result = server.enterprise.getPersons("SKY145128")
```

Gets the persons belonging to the enterprise matching the given attributes.

---

## 6.10 Method:fetchPersons

### Documentation

None

---

### Arguments

doc	struct	enterprise document
fSpec	struct	fetch specification

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

fSpec = {}
fSpec['attributes'] = ["name",]
enterprise = server.enterprise.fetch(
    "description like '*skyrix*')[0]
result = server.enterprise.fetchPersons(enterprise, fSpec)
```

Fetches all persons belonging to the enterprise with the description 'Skyrix', only the name attribute is fetched.

---

### Arguments

number	int	unique enterprise number
fSpec	struct	fetch specification

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

fSpec = {}
fSpec['attributes'] = ["name", "nickname"]
result = server.enterprise.fetchPersons(102560, fSpec)
```

Fetches all persons belonging to the enterprise with the enterprise number "102560", only the name attribute is fetched.

---

### Arguments

number	string	Unique enterprise number
fSpec	struct	fetch specification

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

fSpec = {}
fSpec['attributes'] = ["name",]
result = server.enterprise.fetchPersons("SKY145128", fSpec)
```

Fetches all persons belonging to the enterprise with the enterprise number "SKY145128", only the name attribute is fetched.

---

## 6.11 Method: getProjects

### Documentation

None

### Arguments

doc	struct	enterprise document
-----	--------	---------------------

### Result

array	array of project documents
-------	----------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
enterprise = server.enterprise.fetch(
    "description like '*skyrix*')[0]
result = server.enterprise.getProjects(enterprise)
```

Fetches all the projects for the enterprise.

### Arguments

number	int	unique enterprise number
--------	-----	--------------------------

### Result

array	array of project documents
-------	----------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.getProjects(102560)
```

Fetches all the projects for the enterprise with the number "102560".

---

### Arguments

number	string	Unique enterprise number
--------	--------	--------------------------

### Result

array	array of project documents
-------	----------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.getProjects("SKY145128")
```

Fetches all the projects for the enterprise.

---

## 6.12 Method:deletePerson

### Documentation

None

---

### Arguments

entnumber	string	unique enterprise number
pernumber	string	unique person number

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
number = server.person.fetch(
    "name like '*beeblebrox*")[0]['number']
result = server.enterprise.deletePerson("145128",str(number))
```

Deletes the person from the enterprise.

---

### Arguments

entnumber	int	unique enterprise number
pernumber	string	unique person number

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
```



## 6.12 Method:deletePerson

---

```
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
number = server.person.fetch(
    "name like '*Beeblebrox*'" ) [0] ['number']
result = server.enterprise.deletePerson(145128,str(number))
```

Deletes the person from the enterprise.

---

### Arguments

entnumber	string	unique enterprise number
pernumber	int	unique person number

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
number = server.person.fetch(
    "name like '*Beeblebrox*'" ) [0] ['number']
result = server.enterprise.deletePerson("145128",number)
```

Deletes the person from the enterprise.

---

### Arguments

entnumber	int	unique enterprise number
pernumber	int	unique person number

**Result**

Kein Rückgabewert

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
number = server.person.fetch(
    "name like '*Beeblebrox*')[0]['number']
result = server.enterprise.deletePerson(145128,number)
```

Deletes the person from the enterprise.

---

## 6.13 Method:getById

### Documentation

None

---

### Arguments

enterpriseld	string	unique enterprise ID
--------------	--------	----------------------

### Result

struct	enterprise document
--------	---------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.enterprise.getById("145128")
```

Fetches the enterprise with the enterprise ID "145128".

---

## 6.14 Method:insertPerson

### Documentation

None

---

### Arguments

entnumber	string	Unique number of an enterprise
pernumber	string	Unique number of a person

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch(
    "name like '*beeblebrox*')[0]
enterprise = server.enterprise.fetch(
    "description like '*skyrix*')[0]
result = server.enterprise.insertPerson(
    enterprise['number'], person['number'])
```

Inserts the person into the enterprise.

---

## **7 Package: jobs**

## 7.1 Method:markDone

### Documentation

None

---

### Arguments

jobID	string	unique job ID
-------	--------	---------------

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.markDone('123624')
```

Sets the job status of the job with ID '123624' to 'done'.

---

### Arguments

jobID	string	unique job ID
comment	string	comment

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.markDone('123624', 'Finished')
```

## 7.1 Method:markDone

---

Sets the job status of the job with ID '123624' to 'done' and adds the comment 'Finished' to this operation.

---

## 7.2 Method:archiveJob

### Documentation

None

---

### Arguments

jobID	string	unique job ID
-------	--------	---------------

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.archiveJob('123624')
```

Sets the job status of the job with ID '123624' to archived'.

---

### Arguments

jobID	string	unique job ID
comment	string	comment

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.archived('123624', 'Archived')
```



## 7.2 Method:archiveJob

---

Sets the job status of the job with ID '123624' to 'archived' and adds the comment 'Archived' to this operation.

---

## 7.3 Method:acceptJob

### Documentation

None

---

### Arguments

jobID	string	unique job ID
-------	--------	---------------

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.acceptJob('123624')
```

Accepts the job with ID '123624', sets its status to 'processing'.

---

### Arguments

jobID	string	unique job ID
comment	string	comment

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.acceptJob('123624', 'Accepted')
```

### 7.3 Method:acceptJob

---

Accepts the job with ID '123624', sets its status to 'processing' and adds the comment 'Accepted' to this operation.

---

## 7.4 Method:annotateJob

### Documentation

None

---

### Arguments

jobID	string	unique job ID
comment	string	comment

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.annotateJob('123624','Comment')
```

Adds the comment 'Comment' to the job with ID '123624'.

---

## 7.5 Method:rejectJob

### Documentation

None

---

### Arguments

jobID	string	unique job ID
-------	--------	---------------

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.rejectJob('123624')
```

Rejects the job with ID '123624', sets its status to 'rejected'.

---

### Arguments

jobID	string	unique job ID
comment	string	comment

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.rejectJob('123624', 'Not good enough')
```

Rejects the job with ID '123624' and adds the comment 'Not good enough' to this operation.

---

## 7.6 Method:reactivateJob

### Documentation

None

---

### Arguments

jobID	string	unique job ID
-------	--------	---------------

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.reactivateJob('123624')
```

Reactivates the job with ID '123624', sets its status to 'created'.

---

### Arguments

jobID	string	unique job ID
comment	string	comment

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.reactivateJob('123624', 'Reactivated')
```

Reactivates the job with ID '123624', sets its status to 'created' and adds the comment 'Reactivated' to this operation.

---



## 7.7 Method:getToDoList

### Documentation

None

---

### Arguments

no args

### Result

array	array of job documents
-------	------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getToDoList()
```

Gets the todo list of the current user.

---

### Arguments

ids	boolean	IDs and versions flag
-----	---------	-----------------------

### Result

array	array of IDs and versions or array of job documents
-------	---

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getToDoList(1)
```

Gets the IDs and versions of the entries in the todo list of the current user.

---

### Arguments

ids	boolean	IDs and versions flag
personID	string	unique person ID

### Result

array	array of IDs and versions or array of job documents
-------	---

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.getToDoList(0, '123240')
```

Gets the todo list of the user with the ID '123240'.

---

# 7.8 Method:getDelegatedJobs

**Documentation**

None

**Arguments**

no args

**Result**

array	array of job documents
-------	------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getDelegatedJobs()
```

Gets the list of delegated jobs for the current user.

**Arguments**

ids	boolean	IDs and versions flag
-----	---------	-----------------------

**Result**

array	array of IDs and versions or array of job documents
-------	---

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getDelegatedJobs(1)
```

Gets the IDs and versions of the entries in the delegated job list of the current user.

---

### Arguments

ids	boolean	IDs and versions flag
personID	string	unique person ID

### Result

	array	array of IDs and versions or array of job documents
--	-------	---

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.getDelegatedJobs(0, '123240')
```

Gets the delegated job list of the user with the ID '123240'.

---

## 7.9 Method:getArchivedJobs

### Documentation

None

---

### Arguments

no args

### Result

array	array of job documents
-------	------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getArchivedJobs()
```

Gets the list of archived jobs for the current user.

---

### Arguments

ids	boolean	IDs and versions flag
-----	---------	-----------------------

### Result

array	array of IDs and versions or array of job documents
-------	---

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getArchivedJobs(1)
```

Gets the IDs and versions of the entries in the archived job list of the current user.

---

### Arguments

ids	boolean	IDs and versions flag
personID	string	unique person ID

### Result

array	array of IDs and versions or array of job documents
-------	---

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.getArchivedJobs(0, '123240')
```

Gets the archived job list of the user with the ID '123240'.

---

## 7.10 Method:getJobs

### Documentation

None

---

### Arguments

jobids	array	array of job IDs
--------	-------	------------------

### Result

array	array of job documents
-------	------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.getJobs(['123240', '123152'])
```

Gets the jobs with the IDs '123240' and '123152'.

---

## 7.11 Method: getHistory

### Documentation

None

---

### Arguments

jobid	string	unique job Id
-------	--------	---------------

### Result

array	array of job history documents
-------	--------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.getHistory('123152')
```

Gets the job history of the job with ID '123152'.

---



## 7.12 Method:assignToProject

### Documentation

None

---

### Arguments

jobId	string	unique job ID
projectId	string	unique project ID

### Result

boolean	success indicator
---------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.assignToProject('123152','12235')
```

Assign the job with ID '123152' to the project with ID '12235'.

---

### Arguments

jobId	string	unique job ID
projectId	string	unique project ID
logText	string	log text

### Result

boolean	success indicator
---------	-------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.jobs.assignToProject('123152','12235','Assign')
```

Assign the job with ID '123152' to the project with ID '12235' and adds the comment 'Assign' to this operation.

---

# 7.13 Method:detachFromProject

**Documentation**

None

---

**Arguments**

jobId	string	unique job ID
-------	--------	---------------

**Result**

boolean	success indicator
---------	-------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.detachFromProject('123152')
```

Detachs the job with ID '123152' from the currently assigned project.

---

**Arguments**

jobId	string	unique job ID
logText	string	log text

**Result**

boolean	success indicator
---------	-------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.jobs.detachFromProject('123152','Detach')
```

Detachs the job with ID '123152' from the currently assigned project and adds the log text 'Detach' to this operation.

---

## 7.14 Method:getProjects

### Documentation

None

---

### Arguments

doc	struct	PersonDocument, must contain the person number.
-----	--------	---

### Result

array	Array of ProjectDocuments.
-------	----------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch("name like '*beeblebrox*")[0]
result = server.person.getProjects(person)
```

Gets all the projects for the person.

---

### Arguments

doc	string	Unique person number.
-----	--------	-----------------------

### Result

array	Array of ProjectDocuments.
-------	----------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.getProjects('SKY134930')
```

Gets all the projects for the person.

---

# 7.15 Method:fetchJobs

**Documentation**

None

**Arguments**

persondoc	struct	person document
fspec	struct	fetch specification

**Result**

array	Array with job documents.
-------	---------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
person = server.person.fetch("name like '*beeblebrox*")[0]
result = server.person.fetchJobs(person,"type = 'ToDoJob'")
```

Requests the jobs of the person with the person Name "Beeblebrox".

---

## 7.16 Method:fetchJobIds

### Documentation

None

---

### Arguments

persondoc	struct	person document
fspec	struct	fetch specification

### Result

array	Array with job IDs
-------	--------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch(
    "name like '*beeblebrox*")[0]
fSpec = {}
fSpec['qualifier'] = "name like '*Wichtig*'"
fSpec['sortOrdering'] = ['name',]
result = server.person.fetchJobIds(person, fSpec)
```

Requests the job-IDs of the person with the name "Beeblebrox".

---



# 7.17 Method:insertJob

**Documentation**

None

---

**Arguments**

persondoc	struct	person document
jobdoc	struct	job document

**Result**

struct	job document
--------	--------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')

import time, xmlrpclib
person = server.person.fetch(
    "name like '*beeblebrox'")[0]
job = {}
job['startDate'] = xmlrpclib.DateTime(time.time())
job['name'] = 'New Job'
result = server.person.insertJob(person, job)
```

Associates the job with the person.

---

# 7.18 Method:updateJob

## Documentation

None

## Arguments

persondoc	struct	person document
jobdoc	struct	job document

## Result

struct	job document
--------	--------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch("name like '*beebledro*'") [0]
job = server.person.fetchJobs(person,"type = 'ToDoJob'") [0]
job['name'] = "Updated"
result = server.person.updateJob(person, job)
```

Updates the job.

# 7.19 Method:deleteJob

## Documentation

None

---

## Arguments

persondoc	struct	person document
jobdoc	struct	job document

## Result

Kein Rückgabewert

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.person.deleteJob(
    {
        'id' : "12345",
    },
    {
        'id' : "8888",
        'name' : "Aufgabe",
    }
)
```

Deletes the job with ID "8888" associated with the person with person ID "12345".

---



## **8 Package: mail**

# 8.1 Method:createMail

## Documentation

None

---

## Arguments

no args

## Result

string	Key to identify the new mail transaction.
--------	---

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.mail.createMail()
```

This method starts a mail transaction. The return value is a string to identifier the current mail context.

---

## 8.2 Method:setFrom

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
from	string	'From' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setFrom(mailId, 'marvin@heart_of_gold.uni')
```

This method sets the 'from' header field.

---

## 8.3 Method:addTo

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
to	string	'To' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.addTo(mailId, 'marvin@heart_of_gold.uni')
```

This method adds an entry to the 'to' header field.

---



### 8.4 Method:setTo

#### Documentation

None

---

#### Arguments

mid	string	Mail transaction identifier.
to	string	'To' mail header field.

#### Result

Kein Rückgabewert

#### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setTo(mailId, 'marvin@heart_of_gold.uni')
```

This method sets the 'To' header field.

---

## 8.5 Method:addCC

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
cc	string	'CC' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.addCC(mailId, 'marvin@heart_of_gold.uni')
```

This method adds an entry to the 'CC' header field.

---

## 8.6 Method:setCC

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
from	string	'CC' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setCC(mailId, 'marvin@heart_of_gold.uni')
```

This method sets the 'CC' header field.

---

## 8.7 Method:addBCC

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
cc	string	'BCC' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.addBCC(mailId, 'marvin@heart_of_gold.uni')
```

This method adds an entry to the 'BCC' header field.

---

## 8.8 Method:setBCC

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
from	string	'BCC' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setBCC(mailId, 'marvin@heart_of_gold.uni')
```

This method sets the 'BCC' header field.

---

## 8.9 Method:setSubject

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
subject	string	'Subject' mail header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setSubject(mailId, 'Thanks for the fish')
```

This method sets the 'Subject' of a mail.

---

## 8.10 Method:addHeaderValue

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
key	string	The name of the header field.
value	string	The value of the header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.addHeaderValue(mailId, 'return-path', '<marvin@heart_of_gold.uni>')
```

This method adds a value to a key.

---

## 8.11 Method:setHeaderValue

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
key	string	The name of the header field.
value	string	The value of the header field.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setHeaderValue(mailId, 'return-path', '<marvin@heart_of_gold.uni>')
```

This method sets a value for a key.

---



## 8.12 Method:addHeaders

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
headers	struct	Dictionary contains the header entries.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
mailId = server.mail.createMail()
result = server.addHeaders(mailId, { 'return-path': '<marvin@heart_of_gold.uni>' })
```

This method adds header values.

---

## 8.13 Method:setHeaders

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
headers	struct	Dictionary contains the header entries.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setHeaderValue(mailId, 'return-path', '<marvin@heart_of_gold.uni>')
```

This method sets header values.

---

## 8.14 Method:setTextBody

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
body	struct	A string with a plain-text mail body.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setTextBody(mailId, 'The answer is: 42.')
```

This method sets the plain text body of a mail. If the plain text body and the html text body are set, the mail got a multipart/alternative body.

---

## 8.15 Method:setHtmlBody

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
body	struct	A string with a html-text mail body.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.setTextBody(mailId, '<b>The anwser is: 42.</b>')
```

This method sets the html text body of a mail. If the plain text body and the html text body are set, the mail got a multipart/alternative body.

---

## 8.16 Method:addAttachment

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
filename	string	A string with the filename of the attachment. If no blob is set, this must be an absolute filename to an file, which should be attached. The Default 'DeniedAbsolutePathBlobs' denied the use of absolute a path.
contentType	string	The content type of the attachment.
blob	base64	The base64-encoded attachment blob.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
mailId = server.mail.createMail()
blob    = "Ny02Cg=="
result = server.addAttachment(mailId, 'theQuestion.txt', 'plain/text', blob)
```

This method adds an attachment to a mail.

---

## 8.17 Method:send

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
keepOpen	boolean	Whether to keep open or close the mail transaction after send.

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
blob    = "Nyo2Cg=="
result = server.addAttachment(mailId, 'theQuestion.txt', 'plain/text', blob)
result = server.setTo(mailId, 'support@skyrix.com')
result = server.send(mailId, 0)
```

This method sends the mail.

---

## 8.18 Method:getTransactions

### Documentation

None

---

### Arguments

no args

### Result

array	All mail transactions for the current user.
-------	---

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.mail.getTransactions()
```

This method returns the mail transactions for the current user.

---

## 8.19 Method:deleteAllTransactions

### Documentation

None

---

### Arguments

no args

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.mail.deleteAllTransactions()
```

This method deletes all mail transactions.

---



## 8.20 Method:closeTransaction

### Documentation

None

---

### Arguments

mid	string	Mail transaction identifier.
-----	--------	------------------------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
mailId = server.mail.createMail()
result = server.mail.closeTransaction(mailId)
```

This method deletes the given mail transaction.

---



## **9 Package: person**

## 9.1 Method: getById

### Documentation

None

---

### Arguments

personID	string	unique person ID
----------	--------	------------------

### Result

struct	person document
--------	-----------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.getById("134930")
```

Fetches the person with the given id.

---

## 9.2 Method:getByNumber

### Documentation

None

---

### Arguments

personNumber	string	unique person number
--------------	--------	----------------------

### Result

struct	person document
--------	-----------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.getByNumber("SKY134930")
```

Fetches the person with the number "SKY134930".

---

## 9.3 Method:deleteByNumber

### Documentation

None

---

### Arguments

personNumber	string	unique person number
--------------	--------	----------------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.insert({'name' : 'Trillian',})
result = server.person.deleteByNumber(person['number'])
```

Deletes the person.

---

## 9.4 Method:fetchIds

### Documentation

None

---

### Arguments

fSpec	struct	fetch specification
-------	--------	---------------------

### Result

array	Array of person IDs
-------	---------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.fetchIds(
    {
        'qualifier' : "name like '*beeblebrox*",
    }
)
```

Fetches all IDs for persons with a name attribute containing the string "Beeblebrox".

---

## 9.5 Method:fetch

### Documentation

None

---

### Arguments

fspec	struct	fetch specification
-------	--------	---------------------

### Result

array	array of person documents
-------	---------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.fetch(
    {
        'qualifier'      : "name like '*beeblebrox'",
        'sortOrderings' : ["name",],
    }
)
```

Fetches all the persons with a name attribute containing the string "Beeblebrox", sorted by the name attribute.

---



## 9.6 Method:insert

### Documentation

None

---

### Arguments

dict	struct	Dictionary
------	--------	------------

### Result

struct	person document
--------	-----------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.insert(
    {
        'firstname' : "Arthur",
        'name'      : "Dent",
    }
)
```

Inserts a new person document for a person called "Arthur Dent".

---

## 9.7 Method:update

### Documentation

None

---

### Arguments

doc	struct	person document
-----	--------	-----------------

### Result

struct	PersonDocument
--------	----------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch("name like '*beeblebrox*")[0]
person['nickname'] = 'Praesident'
result = server.person.update(person)
```

Updates the person.

---

## 9.8 Method:delete

### Documentation

None

---

### Arguments

doc	struct	person document
-----	--------	-----------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.insert({'name' : 'Trillian',})
result = server.person.delete(person)
```

Fügt eine Person ein und löscht diese anschliessend wieder, dabei wird die Nummer genutzt, die bei der Anlegeoperation innerhalb des Personendokumentes zurückgegeben wurde. Deletes the person.

---

## 9.9 Method: getEnterprises

### Documentation

None

---

### Arguments

doc	struct	person document
-----	--------	-----------------

### Result

array	array of enterprise documents
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch("name like '*beeblebrox*")[0]
result = server.person.getEnterprises(person)
```

Requests all enterprise documents associated with the person.

---

# 9.10 Method:fetchEnterprises

**Documentation**

None

**Arguments**

doc	struct	person document
fspec	struct	fetch specification

**Result**

array	array of enterprise documents
-------	-------------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
person = server.person.fetch("name like '*beebledrox*")[0]
fSpec = {}
fSpec['hints'] = {}
fSpec['hints']['attributes'] = ['description','bank',]
result = server.person.fetchEnterprises(person, fSpec)
```

Fetches only the description and bank attributes of the enterprises associated with the person.

## 9.11 Method:insertEnterprise

### Documentation

None

---

### Arguments

persondoc	struct	person document
enterprisedoc	struct	enterprise document

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
enterprise = server.enterprise.fetch("description like '*skyrix*")[0]
person = server.person.insert({'name' : 'Dent',})
result = server.person.insertEnterprise(person, enterprise)
```

Inserts the person into the enterprise.

---

## 9.12 Method:deleteEnterprise

### Documentation

None

---

### Arguments

persondoc	struct	person document
enterprisedoc	struct	enterprise document

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
enterprise = server.enterprise.fetch("description like '*skyrix*")[0]
person = server.person.insert({'name' : 'Dent',})
server.person.insertEnterprise(person, enterprise)
result = server.person.deleteEnterprise(person, enterprise)
```

Deletes the person from the enterprise.

---

## 9.13 Method: get

### Documentation

None

---

### Arguments

dict	struct	Dictionary containing the query attributes.
------	--------	---

### Result

array	Array with a person document
-------	------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.get(
    {
        'name'      : "Beeblebrox",
    }
)
```

Requests the person with the attribute name equal to "Beeblebrox".

---



## 9.14 Method:fetchJobs

### Documentation

None

---

### Arguments

dict	struct	Dictionary containing the query attributes.
------	--------	---

### Result

array	Array with a person document
-------	------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.get(
    {
        'name'      : "Beeblebrox",
    }
)
```

Requests the person with the attribute name equal to "Beeblebrox".

---

## 9.15 Method:fetchJobIds

### Documentation

None

---

### Arguments

dict	struct	Dictionary containing the query attributes.
------	--------	---

### Result

array	Array with a person document
-------	------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.get(
    {
        'name'      : "Beeblebrox",
    }
)
```

Requests the person with the attribute name equal to "Beeblebrox".

---

## 9.16 Method:insertJob

### Documentation

None

---

### Arguments

dict	struct	Dictionary containing the query attributes.
------	--------	---

### Result

array	Array with a person document
-------	------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.get(
    {
        'name'      : "Beeblebrox",
    }
)
```

Requests the person with the attribute name equal to "Beeblebrox".

---

## 9.17 Method:updateJob

### Documentation

None

---

### Arguments

dict	struct	Dictionary containing the query attributes.
------	--------	---

### Result

array	Array with a person document
-------	------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.get(
    {
        'name'      : "Beeblebrox",
    }
)
```

Requests the person with the attribute name equal to "Beeblebrox".

---

## 9.18 Method:deleteJob

### Documentation

None

---

### Arguments

doc	struct	person document
doc	struct	job document

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.person.deleteJob("134930", "313245")
```

Deletes the job with the ID "313245" from the person with the ID "134930".

---



## **10 Package: project**

## 10.1 Method:fetch

### Documentation

None

---

### Arguments

fspec	struct	FetchSpecification
-------	--------	--------------------

### Result

array	Array of ProjectDocument objects.
-------	-----------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

dict = {}
dict['qualifier'] = "name like '*xmlrpc*"
result = server.project.fetch(dict)
```

Returns all projects with the string "xmlrpc" in their name attribute.

---

### Arguments

qualifier	string	qualifier
-----------	--------	-----------

### Result

array	Array of ProjectDocument objects.
-------	-----------------------------------



### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.fetch("name like '*xmlrpc*')
```

Returns all projects with the string "xmlrpc" in their name attribute.

---

## 10.2 Method:insert

### Documentation

None

---

### Arguments

doc	struct	ProjectDocument
-----	--------	-----------------

### Result

struct	The new ProjectDocument.
--------	--------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

import xmlrpclib, time
startDate = xmlrpclib.DateTime(time.strptime(
                                '2002-01-02', '%Y-%m-%d'))
endDate = xmlrpclib.DateTime(time.strptime(
                              '2002-01-02', '%Y-%m-%d'))

dict = {}
dict['name'] = "Project"
dict['startDate'] = startDate
dict['endDate'] = endDate
result = server.project.insert(dict)
```

Creates a new project document with the name "Project" and the given start and end date.

---

# 10.3 Method:update

**Documentation**

None

**Arguments**

doc	struct	ProjectDocument
-----	--------	-----------------

**Result**

struct	The updated ProjectDocument.
--------	------------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
project = server.project.fetch("name like '*xmlrpc*")[0]
project['name'] = "xmlrpc_"
result = server.project.update(project)
```

Updates the project with the project code "xmlrpc", sets a new name.

---

## 10.4 Method:delete

### Documentation

None

---

### Arguments

doc	struct	ProjectDocument
-----	--------	-----------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')

import xmlrpclib, time
startDate = xmlrpclib.DateTime(time.strptime(
                                '2002-01-02', '%Y-%m-%d'))
endDate = xmlrpclib.DateTime(time.strptime(
                              '2002-01-02', '%Y-%m-%d'))

dict = {}
dict['name'] = "To Be Deleted"
dict['startDate'] = startDate
dict['endDate'] = endDate
project = server.project.insert(dict)
result = server.project.delete(project)
```

Deletes the project.

---

# 10.5 Method:cwd

**Documentation**

None

---

**Arguments**

projectCode	string	Project code.
-------------	--------	---------------

**Result**

string	Current working directory.
--------	----------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.cwd("xmlrpc")
```

Returns the current working directory of the project with the project code "xmlrpc".

---

## 10.6 Method:cd

### Documentation

None

---

### Arguments

projectCode	string	Project code.
path	string	Directory path to change into.

### Result

	boolean	yes or no.
--	---------	------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.cd("xmlrpc", "/tmp")
```

Sets the current working directory of the project with the code "xmlrpc" to the path "/tmp".

---

# 10.7 Method:ls

**Documentation**

None

**Arguments**

projectCode	string	Project code.
args	array	Paths to list.

**Result**

array	Contents of paths as a array.
-------	-------------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.ls("xmlrpc", ["/", "/tmp"])
```

Lists the contents of the paths "/" and "/tmp" in the project with the code "xmlrpc".

**Arguments**

projectCode	string	Project code.
path	string	Path to list.

**Result**

array	Contents of paths as a array.
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.ls("xmlrpc", "/")
```

Lists the contents of the path "/" in the project with the code "xmlrpc".

---

### Arguments

projectCode	string	Project code.
-------------	--------	---------------

### Result

array	Contents of paths as a array.
-------	-------------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.ls("xmlrpc")
```

Lists the contents of the current working directory in the project with the code "xmlrpc".

---



# 10.8 Method:mkdir

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Directory paths to create.

## Result

	boolean	yes or no.
--	---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.mkdir("xmlrpc", ["/tmp", "/home"])
```

Creates the directories "/tmp" and "/home" in the project with the project code "xmlrpc".

---

## Arguments

projectCode	string	Project code.
args	string	Directory path to create.

## Result

	boolean	yes or no.
--	---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.mkdir("xmlrpc", "/tmp")
```

Creates the directory "/tmp" in the project with the project code "xmlrpc".

---

# 10.9 Method:rmdir

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Directories to remove.

## Result

boolean	yes or no.
---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.rmdir("xmlrpc", ["/tmp", "/home"])
```

Removes the directories "/tmp" and "/home" in the project with the project code "xmlrpc".

---

## Arguments

projectCode	string	Project code.
args	string	Directories to remove.

## Result

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.rmdir("xmlrpc", "/tmp")
```

Removes the directory "/tmp" in the project with the project code "xmlrpc".

---

# 10.10 Method:attr

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Files/dirs to show attributes of.

## Result

struct	Attributes.
--------	-------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.attr("xmlrpc", ["file1.txt", "file2.txt"])
```

Returns the attributes of the files "file1.txt" and "file2.txt" in the project with the project code "xmlrpc".

---

## Arguments

projectCode	string	Project code.
args	string	File/dirs to show attributes of.

## Result

struct	Attributes.
--------	-------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.attr("xmlrpc", "file1.txt")
```

Returns the attributes of the file "file1.txt" in the project with the project code "xmlrpc".

---

# 10.11 Method:zip

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Files/dirs to be zipped.

## Result

	base64	Zip file as base64-encoded binary.
--	--------	------------------------------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.zip("xmlrpc", ["file1.txt", "file2.txt"])
```

Returns a zipped binary with the files "file1.txt" and "file2.txt" of the project with the code "xmlrpc" in it.

---

## Arguments

projectCode	string	Project code.
args	string	File/dir to be zipped.

## Result

	base64	Zip file as base64-encoded binary.
--	--------	------------------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.zip("xmlrpc", "file1.txt")
```

Returns a zipped binary with the file "file1.txt" of the project with the code "xmlrpc" in it.

---



# 10.12 Method:rm

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Files to remove.

## Result

	boolean	yes or no.
--	---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.rm("xmlrpc", ["file1.txt", "file2.txt"])
```

Removes the files "file1.txt" and "file2.txt" in the project with the project code "xmlrpc".

---

## Arguments

projectCode	string	Project code.
args	string	File to remove.

## Result

	boolean	yes or no.
--	---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.rm("xmlrpc", "file1.txt")
```

Removes the file "file1.txt" in the project with the project code "xmlrpc".

---

# 10.13 Method:cp

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	1..n-1 -> source paths, n -> target path

## Result

boolean	yes or no.
---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.cp("xmlrpc",
                           ["file1.txt", "file2.txt", "/tmp"])
```

Copies the files "file1.txt" and "file2.txt" to the directory "/tmp" in the project with the project code "xmlrpc".

---

## Arguments

projectCode	string	Project code.
src	string	source poth
dest	string	destination path

## Result

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.cp("xmlrpc", "file1.txt", "/tmp")
```

Copies the file "file1.txt" to the directory "/tmp" in the project with the project code "xmlrpc".

---

## 10.14 Method:mv

### Documentation

None

---

### Arguments

projectCode	string	Project code.
args	array	1..n-1 -> source paths, n -> target path

### Result

boolean	yes or no.
---------	------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.mv("xmlrpc",
                           ["file1.txt", "file2.txt", "/tmp"])
```

Moves the files "file1.txt" and "file2.txt" to the directory "/tmp" in the project with the project code "xmlrpc".

---

### Arguments

projectCode	string	Project code.
src	string	source poth
dest	string	destination path

### Result

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.mv("xmlrpc", "file1.txt", "/tmp")
```

Moves the file "file1.txt" to the directory "/tmp" in the project with the project code "xmlrpc".

---

# 10.15 Method:ln

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	0 -> source path, 1 -> target path

## Result

boolean	yes or no.
---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.ln("xmlrpc",["file1.txt","file1.lnk"])
```

Links the file "file1.txt" to the link "file1.lnk" in the project with the project code "xmlrpc".

---

## Arguments

projectCode	string	Project code.
src	string	source poth
dest	string	destination path

## Result

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.ln("xmlrpc", "file1.txt", "file1.lnk")
```

Links the file "file1.txt" to the link "file1.lnk" in the project with the project code "xmlrpc".

---



# 10.16 Method:exists

**Documentation**

None

---

**Arguments**

projectCode	string	Project code.
args	array	Paths to check for existence.

**Result**

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.exists("xmlrpc", ["file1.txt",])
```

Checks for the existence of file "file1.txt" in the project with the project code "xmlrpc".

---

**Arguments**

projectCode	string	Project code.
args	string	Path to check for existence.

**Result**

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.exists("xmlrpc", "file1.txt")
```

Checks for the existence of file "file1.txt" in the project with the project code "xmlrpc".

---

# 10.17 Method:isdir

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Paths to check for being a directory.

## Result

	boolean	yes or no.
--	---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.isdir("xmlrpc", ["/tmp",])
```

Checks if the path "/tmp" in the project with the project code "xmlrpc" is a directory.

---

## Arguments

projectCode	string	Project code.
args	string	Path to check for being a directory.

## Result

	boolean	yes or no.
--	---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.isdir("xmlrpc", "/tmp")
```

Checks if the path "/tmp" in the project with the project code "xmlrpc" is a directory.

---

# 10.18 Method:islink

## Documentation

None

---

## Arguments

projectCode	string	Project code.
args	array	Paths to check for being a link.

## Result

boolean	yes or no.
---------	------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.islink("xmlrpc", ["file1.lnk",])
```

Checks if the path "file1.lnk" in the project with the project code "xmlrpc" is a link.

---

## Arguments

projectCode	string	Project code.
args	string	Path to check for being a link.

## Result

boolean	yes or no.
---------	------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.islink("xmlrpc", "file1.lnk")
```

Checks if the path "file1.lnk" in the project with the project code "xmlrpc" is a link.

---

## 10.19 Method:flush

### Documentation

None

---

### Arguments

projectCode	string	Project code.
-------------	--------	---------------

### Result

Kein Rückgabewert

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.flush("xmlrpc")
```

Flushes the cache of the project with the project code "xmlrpc".

---

## 10.20 Method:newDocument

### Documentation

None

---

### Arguments

projectCode	int	Project code.
path	string	Path to the new file.
content	string	File content.
attributes	struct	File attributes as dictionary.

### Result

boolean	Return YES or NO.
---------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.newDocument(
    "xmlrpc", "/newfile.txt", "Inhalt",
    {
        'NSFileSubject' : "Titel",
    }
)
```

Creates a file "/file1.txt" in the project with the project code "xmlrpc". The file has the subject attribute set to "Titel" and has the content "Inhalt".

---



# 10.21 Method:loadDocument

**Documentation**

None

---

**Arguments**

projectCode	string	Project code.
path	string	Path to the file to load.

**Result**

	base64	File contenty in binary form (base64 encoded).
--	--------	--

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.project.loadDocument("xmlrpc","newfile.txt")
```

Loads the file "file1.txt" from the project "xmlrpc".

---

## 10.22 Method:saveDocument

### Documentation

None

---

### Arguments

projectCode	int	Project code.
path	string	Path to the file to save.
content	string	Content of the file.

### Result

boolean	yes or no
---------	-----------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.project.saveDocument("xmlrpc","newfile.txt",
                                     "Neuer Inhalt")
```

Saves the file "file1.txt" in the project "xmlrpc" with the content "Inhalt".

---

## **11 Package: resource**

## 11.1 Method:insert

### Documentation

None

---

### Arguments

dict	struct	dictionary
------	--------	------------

### Result

struct	Resource document
--------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.resource.insert({'name' : 'Resource',
                                'category' : 'Cat'})
```

Creates a resource 'Resource' within the category 'Cat'.

---

## 11.2 Method:getById

### Documentation

None

---

### Arguments

resourceID	string	unique resource ID
------------	--------	--------------------

### Result

struct	resource document
--------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.resource.getById('12940')
```

Gets the appointment resource with the ID '12940'.

---

## 11.3 Method: getByName

### Documentation

None

---

### Arguments

resourceName	string	resource name
--------------	--------	---------------

### Result

struct	resource document
--------	-------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.resource.getByName('Resource')
```

Gets the appointment resource with the name 'Resource'.

---

# 11.4 Method:update

**Documentation**

None

---

**Arguments**

dict	struct	dictionary
------	--------	------------

**Result**

struct	Resource document
--------	-------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.resource.update({'name' : 'Updated',
                                'id' : '12940'})
```

Updates the resource with ID '12940', changes the name to 'Updated'.

---

## 11.5 Method:delete

### Documentation

None

---

### Arguments

resourceID	string	unique resource ID
------------	--------	--------------------

### Result

boolean	operation result
---------	------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.resource.delete('12940')
```

Deletes the appointment resource with the ID '12940'.

---



# 11.6 Method:categories

**Documentation**

None

---

**Arguments**

no args

**Result**

array	category names
-------	----------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.resource.categories()
```

Returns all available categories.

---

**Arguments**

category	string	category name
----------	--------	---------------

**Result**

array	category members
-------	------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.resource.categories('Cat')
```

Returns all resources of the category 'Cat'.

---

# 11.7 Method:fetch

**Documentation**

None

---

**Arguments**

fspec	struct	fetch specification
-------	--------	---------------------

**Result**

array	array of resource documents
-------	-----------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.resource.fetch("name like '*Resource*')
```

Fetches all appointment resources with a name attribute containing the string "Resource".

---



## **12 Package: system**

## 12.1 Method: getHostName

### Documentation

None

---

### Arguments

no args

### Result

string	The hostname of the server.
--------	-----------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.system.getHostName()
```

Returns the hostname of the server.

---

## 12.2 Method:getServerTimeZone

### Documentation

None

---

### Arguments

no args

### Result

string	The server's timezone.
--------	------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.system.getServerTimeZone()
```

Returns the server's timezone.

---

## 12.3 Method: getServerTime

### Documentation

None

---

### Arguments

no args

### Result

dateTime.iso8601	Returns the current date and time of the server.
------------------	--

### Examples

#### getservertime.py:

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.system.getServerTime()
```

Returns the current date and time of the server.

---



# 12.4 Method:listMethods

**Documentation**

None

---

**Arguments**

no args

**Result**

array	Array of strings
-------	------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.system.listMethods()
```

Returns the names of all available XML-RPC functions.

---

## 12.5 Method:methodSignature

### Documentation

None

---

### Arguments

functionname	string	Name of a XML-RPC function
--------------	--------	----------------------------

### Result

array	Array of signatures
-------	---------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.system.methodSignature("system.methodSignature")
```

Returns the signatures of the XML-RPC command "system.methodSignature"

---

# 12.6 Method:methodHelp

**Documentation**

None

---

**Arguments**

functionname	string	Name of a XML-RPC function
--------------	--------	----------------------------

**Result**

	string	Help string
--	--------	-------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.system.methodHelp("system.methodSignature")
```

Returns the help of the XML-RPC command "system.methodSignature"

---



## **13 Package: team**

## 13.1 Method: getTeams

### Documentation

None

---

### Arguments

no args

### Result

array	array of Team documents
-------	-------------------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.team.getTeams()
```

Requests all teams.

---

## 13.2 Method:getByLogin

### Documentation

None

---

### Arguments

string	login
--------	-------

### Result

struct	team document
--------	---------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.team.getByLogin('team')
```

Gets the team with the login 'team'.

---

## 13.3 Method: getByNumber

### Documentation

None

---

### Arguments

teamnumber	string	team number
------------	--------	-------------

### Result

struct	team document
--------	---------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.team.getByNumber('12345')
```

Gets the team with the number '12345'.

---



# 13.4 Method:getMembersForTeam

**Documentation**

None

---

**Arguments**

teamnumber	string	team number
------------	--------	-------------

**Result**

array	array of team documents
-------	-------------------------

**Examples**

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                           login='test', password='system')
result = server.team.getMembersForTeam('12345')
```

Gets the members for the team with the number '12345'.

---

## 13.5 Method:insert

### Documentation

None

---

### Arguments

team	struct	team information
------	--------	------------------

### Result

struct	team document
--------	---------------

### Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.team.insert({'description' : 'new_team'})
```

Inserts a new team with the description "new\_team".

---

# 13.6 Method:setMembers

## Documentation

None

---

## Arguments

teamId	string	unique team ID
accountIds	array	array of unique account IDs

## Result

boolean	success indicator
---------	-------------------

## Examples

```
#!/usr/bin/env python
import xmlrpclib
server = xmlrpclib.Server('http://localhost:20000/x/xmlrpc',
                          login='test', password='system')
result = server.team.setMembers('12345', ['12435', '42142'])
```

Sets the accounts with the IDs '12435' and '42142' as members of the team with the ID '12345'.

---