Introduction

The Interspire Knowledge Manager XML API is a remotely accessible service API to allow Interspire Knowledge Manager users to run many of the Interspire Knowledge Manager API functions using XML requests.

The Interspire Knowledge Manager XML API makes it possible to programmatically update and use your system without needing to physically access it. As XML is a general purpose markup language you can use it to communicate between other applications to Interspire Knowledge Manager PHP application base.

For example you could set your system up to add/update/delete users in your knowledgebase and many other functions.

Submitting a Request

All XML POST request should be sent to:

http://location.of.the.knowledgebase/admin/

Possible Requests

This section will describe the different functions that can be used by the Interspire Knowledge Manager XML API.

Requesting group(s) in Knowledgebase via XML API

Requesting a group(s) API allowed users to retrieve a list of selected group(s) or all available group(s) in Interspire Knowledge Manager remotely.

Creating the Request

The XML document structure for submitting groups request is as follows:

- request - (Required)

- (Required) todo - The action for this request: 'GetGroups'.
- (Required) kbuserlogin
  - (Required) username - The username of the knowledgebase user who has the permission to edit groups in Knowledgebase.
  - (Required) password - The password of the username login above.
- (Optional) groups - Please do not specify this tag if the request is to retrieve all the existing groups in the knowledgebase.
  - id - the id of the targeted groups.

Successful Response

Upon submission of a valid group(s) request, a list of groups and number of groups will be returned.

The format is as follows:

- response
  - status - The value of the status field will be "OK" for a successful response.
  - message - The friendly message of the response
  - TotalGroups - The total number of returned groups.
  - GroupDetails - The information of returned groups. (This node will not be returned if there is no group found in the knowledgebase)
    - group
      - groupid - The returned group id.
      - name - The name of returned group.
      - contactable - The flag if the group is contactable. (1 = yes, 0 = no)
      - numberOfUsers - The total number of user(s) in this group.

Unsuccessful Response

Upon submission of an erroneous response due to a missing field, or an invalid value the groups request will failed. A status message will be returned via XML as to why.

The format is as follows:

- response
  - status – Will contain a value of "ERROR" for an unsuccessful request.
  - message – An overall message stating why the message cannot proceed.
  - Errors
    - error – This node will be repeated for each error and can contain the following:
    - (Attribute) code – An error code associated with this error. See Appendix 1 : Error Codes for a full list of error codes
    - (Attribute / Optional) extra – Any additional information that may be useful to help debug this error.
    - Value – The value of the error node contains a friendly error message that can be shown as to why this problem occurred.

Sample XML Request

Request all the groups in the knowledgebase:

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
    <todo>GetGroups</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
</request>
```

Request only a list of selected groups in the knowledgebase:

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
    <todo>GetGroups</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <groups>
        <id>1</id>
        <id>2</id>
    </groups>
</request>
```

Sample PHP Request

The following sample code is written in PHP and makes use of PHP's CURL functionality.

```
<?php
$data = '
<request>
    <todo>GetGroups</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
</request>
```

```
';
$ch = curl_init("http://location.of.the.knowledgebase/admin/");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
"Accept: application/xml",
"Content?Type: application/xml"
));
echo "<pre>";
echo htmlspecialchars(curl_exec($ch));
echo "</pre>";
?>
```

Add a user to Knowledgebase via XML API - [top]

Creating the Request

The XML document structure for submitting add user request is as follows:

- (Required) request
    - (Required) todo - The action for this request: 'SaveNewUser'.
    - (Required) kbuserlogin
        - (Required) username - The username of the knowledgebase user who has the permission to edit groups in Knowledgebase.
        - (Required) password - The password of the username login above.
    - (Required) userdetails
        - (Required) username – The username of the new user.
        - (Required) password – The password of the new user.
        - (Required) email – The email of the new user.
        - (Required) firstname – First name of the new user. (Required)
        - (Required) lastname – Last name of the new user. (Required)
        - (Required - 1=Active, 0=Inactive) status – Status of the new user.
        - (Optional) groups
            - (One or more) group
                - id - the group id to be assigned to the new user.

Successful Response

Upon submission of a valid add user request, a status code and friendly message will be returned.

The format is as follows:

- response
    - status - The value of the status field will be “OK” for a successful response.

- message - The friendly message of the response

Unsuccessful Response

Upon submission of an erroneous response due to a missing field, or an invalid value the add user request will failed. A status message will be returned via XML as to why.

The format is as follows:

- response
  - status – Will contain a value of "ERROR" for an unsuccessful request.
  - message – An overall message stating why the message cannot proceed.
  - Errors
    - error – This node will be repeated for each error and can contain the following:
    - (Attribute) code – An error code associated with this error. See Appendix 1 : Error Codes for a full list of error codes
    - (Attribute / Optional) extra – Any additional information that may be useful to help debug this error.
    - Value – The value of the error node contains a friendly error message that can be shown as to why this problem occurred.

Sample XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
    <todo>SaveNewUser</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <userdetails>
        <username>new_username</username>
        <password>new_user_password</password>
        <email>new_user@example.com</email>
        <firstname>John</firstname>
        <lastname>Smith</lastname>
        <status>1</status>
    </userdetails>
</request>
```

Sample PHP Request

The following sample code is written in PHP and makes use of PHP's CURL functionality.

```
<?php
$data = '
```

```
<request>
    <todo>SaveNewUser</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <userdetails>
        <username>new_username</username>
        <password>new_user_password</password>
        <email>new_user@example.com</email>
        <firstname>John</firstname>
        <lastname>Smith</lastname>
        <status>1</status>
    </userdetails>
</request>
';
$ch = curl_init("http://location.of.the.knowledgebase/admin/");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
"Accept: application/xml",
"Content?Type: application/xml"
));
echo "<pre>";
echo htmlspecialchars(curl_exec($ch));
echo "</pre>";
```

Edit a user in Knowledgebase via XML API

Creating the Request

The XML document structure for submitting edit user request is as follows:

- (Required) request
    - (Required) todo - The action for this request: 'SaveUpdatedUser'.
    - (Required) kbuserlogin
        - (Required) username - The username of the knowledgebase user who has the permission to edit groups in Knowledgebase.
        - (Required) password - The password of the username login above.
    - (Required) userdetails
        - (Required) username – The username of targeted user.
        - (Required) password – The password of targeted user.
        - (Required) email – The email of targeted user.
        - (Required) firstname – First name of targeted user. (Required)
        - (Required) lastname – Last name of targeted user. (Required)

        - (Required - 1=Active, 0=Inactive) status – Status of targeted user.
        - (Optional) groups - these list of specified group will replace all the group assigned to targeted user
            - (One or more) group
                - id - the group id to be assigned to the targeted user.

Successful Response

Upon submission of a valid edit user request, a status code and friendly message will be returned.

The format is as follows:

- response
    - status - The value of the status field will be "OK" for a successful response.
    - message - The friendly message of the response

Unsuccessful Response

Upon submission of an erroneous response due to a missing field, or an invalid value the edit user request will failed. A status message will be returned via XML as to why.

The format is as follows:

- response
    - status – Will contain a value of "ERROR" for an unsuccessful request.
    - message – An overall message stating why the message cannot proceed.
    - Errors
        - error – This node will be repeated for each error and can contain the following:
        - (Attribute) code – An error code associated with this error. See Appendix 1 : Error Codes for a full list of error codes
        - (Attribute / Optional) extra – Any additional information that may be useful to help debug this error.
        - Value – The value of the error node contains a friendly error message that can be shown as to why this problem occurred.

Sample XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
    <todo>SaveUpdatedUser</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <userdetails>
        <userid>2</userid>
```

```
        <username>new_username</username>
        <password>new_user_password</password>
        <email>new_user@example.com</email>
        <firstname>John</firstname>
        <lastname>Smith</lastname>
        <status>1</status>
    </userdetails>
</request>
```

Sample PHP Request

The following sample code is written in PHP and makes use of PHP's CURL functionality.

```
<?php
$data = '
<request>
    <todo>SaveUpdatedUser</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <userdetails>
            <userid>2</userid>
        <username>new_username</username>
        <password>new_user_password</password>
        <email>new_user@example.com</email>
        <firstname>John</firstname>
        <lastname>Smith</lastname>
        <status>1</status>
    </userdetails>
</request>
';
$ch = curl_init("http://location.of.the.knowledgebase/admin/");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
"Accept: application/xml",
"Content?Type: application/xml"
));
echo "<pre>";
echo htmlspecialchars(curl_exec($ch));
echo "</pre>";
```

Delete user(s) in Knowledgebase via XML API

Creating the Request

The XML document structure for submitting delete user request is as follows:

- (Required) request
    - (Required) todo - The action for this request: 'DeleteUser'.
    - (Required) kbuserlogin
        - (Required) username - The username of the knowledgebase user who has the permission to edit groups in Knowledgebase.
        - (Required) password - The password of the username login above.
    - (Required) targetuserdetails
        - (One or more) userid – The user id of targeted user of this request.

Successful Response

Upon submission of a valid delete user(s) request, a status code and friendly message will be returned.

The format is as follows:

- response
    - status - The value of the status field will be "OK" for a successful response.
    - message - The friendly message of the response

Unsuccessful Response

Upon submission of an erroneous response due to a missing field, or an invalid value the delete user request will failed. A status message will be returned via XML as to why.

The format is as follows:

- response
    - status – Will contain a value of "ERROR" for an unsuccessful request.
    - message – An overall message stating why the message cannot proceed.
    - Errors
        - error – This node will be repeated for each error and can contain the following:
        - (Attribute) code – An error code associated with this error. See Appendix 1 : Error Codes for a full list of error codes
        - (Attribute / Optional) extra – Any additional information that may be useful to help debug this error.
        - Value – The value of the error node contains a friendly error message that can be shown as to why this problem occurred.

Sample XML Request

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

```
<request>
    <todo>DeleteUser</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <targetuserdetails>
        <userid>2</userid>
    </targetuserdetails>
</request>
```

Sample PHP Request

The following sample code is written in PHP and makes use of PHP's CURL functionality.

```
<?php
$data = '
<request>
    <todo>DeleteUser</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <targetuserdetails>
        <userid>2</userid>
    </targetuserdetails>
</request>
';
$ch = curl_init("http://location.of.the.knowledgebase/admin/");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
"Accept: application/xml",
"Content?Type: application/xml"
));
echo "<pre>";
echo htmlspecialchars(curl_exec($ch));
echo "</pre>";
```

Get all user(s) details in Knowledgebase via XML API

Creating the Request

- (Required) request
    - (Required) todo - The action for this request: 'GetUsers'.
    - (Required) kbuserlogin
        - (Required) username - The username of the knowledgebase user who has the permission to edit groups in Knowledgebase.
        - (Required) password - The password of the username login above.
    - (Required) requestuserdetails
        - (One or more) value – the user's field to be returned. Could be one of the following: userid, username, firstname, lastname or email.

Successful Response

Upon submission of a valid get user(s) request, a status code and friendly message will be returned.

The format is as follows:

- response
    - status - The value of the status field will be "OK" for a successful response.
    - message - The friendly message of the response
    - TotalUsers - The total number of returned users.
    - UserDetails - The information of returned groups. (This node will not be returned if there is no group found in the knowledgebase)
        - (Zero or more, depends on the number of returned user(s)) user
        - userid – The user id of the returned user.
        - username – The username of the returned user.
        - firstname – The first name of the returned user.
        - lastname – The last name of the returned user.
        - email – The email address of the returned user.

Unsuccessful Response

Upon submission of an erroneous response due to a missing field, or an invalid value the get user(s) request will failed. A status message will be returned via XML as to why.

The format is as follows:

- response
    - status – Will contain a value of "ERROR" for an unsuccessful request.
    - message – An overall message stating why the message cannot proceed.
    - Errors
        - error – This node will be repeated for each error and can contain the following:
        - (Attribute) code – An error code associated with this error. See Appendix 1 : Error Codes for a full list of error codes
        - (Attribute / Optional) extra – Any additional information that may be useful to help debug this error.
        - Value – The value of the error node contains a friendly error message that can be shown as to why this problem occurred.

Sample XML Request

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
    <todo>GetUsers</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>

    </kbuserlogin>
    <requestuserdetails>
        <value>userid</value>
        <value>username</value>
        <value>firstname</value>
        <value>lastname</value>
        <value>email</value>
    </requestuserdetails>
</request>
```

Sample PHP Request

The following sample code is written in PHP and makes use of PHP's CURL functionality.

```
<?php
$data = '
<request>
    <todo>getUsers</todo>
    <kbuserlogin>
        <username>admin</username>
        <password>password</password>
    </kbuserlogin>
    <requestuserdetails>
        <value>userid</value>
        <value>username</value>
        <value>firstname</value>
        <value>lastname</value>
        <value>email</value>
    </requestuserdetails>
</request>
';
$ch = curl_init("http://location.of.the.knowledgebase/admin/");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
"Accept: application/xml",
```

```
"Content?Type: application/xml"
));
echo "<pre>";
echo htmlspecialchars(curl_exec($ch));
echo "</pre>";
```

Appendix 1: Error Codes

The table below outlines all of the possible return values for the error code attribute.

Code	Extra Data	Function	Reason
XMLNoPermission	N/A	ALL	The user login does not have the permission to perform the request.
XMLInvalidAction	N/A	ALL	Invalid value/ No value was specified in Todo tag
XmlBadLogin	N/A	ALL	The login details of the user login were incorrect.
UserCreateError	N/A	Add User	This error occurs normally because of database error or the invalid value is parsed in.
duplicateUsername	N/A	Add User Update User	The specified username already exists.
XMLNoUsersAttb	N/A	Get User(s)	No user attributes are specified.
XMLBadUsersAttb	N/A	Get User(s)	The specified user attributes are invalid.
XMLUserDeleteError	N/A	Delete User(s)	This error occurs normally because of database error or the invalid value is parsed in.
XMLNoUserId	N/A	Delete User(s)	No target user id was specified for deleting.
XMLUserEditError	N/A	Edit User	This error occurs normally because of database error or the invalid value is parsed in.
XMLGetGroupsError	N/A	Get Group(s)	This error occurs normally because of database error or the invalid value is parsed in.