

Class Documentation

AMODWebApp.webservice.AMODWebAPI Class Reference

Public Member Functions

- string **AuthenticateUser** ()
This method is used for authentication in system.
- bool **SetCaseAsExported** (int caseId)
It's marks object as exported to QNT.
- void **SetProcedureType** (int prcId, int prcTypeId)
Sets type of specified AMODprocedure
- void **UpdateCaseField** (int caseId, string fieldName, string fieldValue)
Updates specified case single field value
- void **UpdateCaseFields** (int caseId, string[] fieldNames, string[] fieldsValues)
Updates specified case multiple fields values
- void **UpdateCaseFields2** (int caseId, string[] fieldNames, string[] fieldsValues)
Updates specified case multiple fields values
- void **AddTableFieldToCase** (int caseId, int formProcedureId, string tableStructureXml, string tableFieldName)
Adds table to case
- void **DeleteAllTableFieldRecords** (int procedureWithTableFieldId, string tableFieldName)
Removes all rows from table placed in field with specified name in specified procedure (for every case)
- void **DeleteTableFieldRecordsConnectedToCase** (int procedureWithTableFieldId, string tableFieldName, int caseId)
Removes all rows from table placed in field with specified name in specified procedure (for specified case)
- DataTable **GetProcedureForms** (int prcId)
Gets all procedures connected to specified one, defined as a form
- DataTable **GetProcedureTableProcedures** (int prcId)
Collects all procedures of specified procedure used to store table fields
- string **GetFormByFieldProcedureType** (int caseId, int typeId)
Gets ID of form containing table with specified procedure type id on specified case form
- DataTable **GetProceduresByType** (int typeId)
Used by QNT app in configuration options
- DataSet **GetCasesByTypeId** (int typeId, int scheduleTypeId, string caseStatus, StringCollection fieldMapping, StringCollection fieldsToInclude)
This function is used by QNT for export application.
- string **GetCase** (int caseId)
This method allow to access to single case by Id.
- string **GetCasesByFieldValue** (string fieldName, string fieldValue)
Get case(s) id(s) by value in specified field
- string **GetCaseIdByFieldValue** (string fieldName, string fieldValue)
Gets id of the case by the value of specified field
- string **CloseCase** (int caseId, string dateOfModification)
Closes specified case
- string **UpdateCase** (int caseId, string dateOfModification, int paramId, String value)

Updates parameter with paramId with value in case with caseId

- string **SendCase** (int caseId, int stageId, string dateOfModification, string userDescription, string comment)
Transfers (forwards) case to another user, possibly changing its state and adding a comment
 - string **GetUserCases** ()
This method gets all cases which are assigned to user.
 - string **GetAMODEmail** ()
This method gets AMODIT email address.
 - string **UserEmailExists** (string email)
Check if user with given e-mail address exists.
 - string **GetCurrentUserEmails** ()
Get all (possibly 4) e-mail addresses for the currently authenticated user.
 - string **CaseForward** (int caseId, string email, string comment)
Adds user specified by email address to specified case as CC.
 - string **CaseAddAttachment** (int caseId, string name, byte[] value)
Adds attachment to specified case.
 - string **CaseAddComment** (int caseId, string comText)
This method adds comment to specified case.
 - string **CaseAddUser** (int caseId, string ccEmail, bool addCCnotContrib, string comment)
Adds specified users as CC or Contributors. If needed, creates new user account and sends invite email.
 - string **CreateCase** (string procedureTitle, string startingStateTitle, string caseTitle)
This method creates a new case.
 - byte[] **FindCase** (int caseId, string caseTitle, string procedureTitle, int limit)
This method gets a list of cases matching given filter.
 - string **GetMostUsedProceduresForCurrentUser** (int rowLimit)
This method gets a list of the most used procedures for currently authenticated user.
 - string **GetMostUsedProcedures** (int userId, int rowLimit)
This method gets a list of the most used procedure for specified user.
 - string **GetProcedure** (string title)
This method gets a procedure which name matches specified string.
 - string **GetProcedureSchemaByTitle** (string procedureTitle)
Gets procedure schema for procedure that name matches specified string
 - string **GetProcedureSchema** (int procedureId)
Gets procedure schema by procedure id
 - string **GetProcedures** (string prefixText, int maxProcedures)
This method gets a list of procedure names matching specified string.
 - string **GenerateDefaultTitle** (string prcTitle)
This method returns default case title for given procedure name.
 - string **GetUser** (int userId)
Gets user info
 - string **ForwardCase** (int caseId, string stage, string user, bool ignoreRestrictions, bool sendEmail)
Forwards case to specified user.
 - void **LogToAMODLog** (string message, string callStack)
Logs specified message and callstack to AMOD Logs
-

Member Function Documentation

void AMODWebApp.webservice.AMODWebAPI.AddTableFieldToCase (int *caseId*, int *formProcedureId*, string *tableStructureXml*, string *tableFieldName*)

Adds table to case

Parameters:

<i>caseId</i>	Id of the case
<i>formProcedureId</i>	Form procedure id
<i>tableStructureXml</i>	New table's structure in XML format
<i>tableFieldName</i>	New field's name

string AMODWebApp.webservice.AMODWebAPI.AuthenticateUser ()

This method is used for authentication in system.

Returns:

If authentication is successfully, method returns GUID. If authentication is failure method returns description of error.

string AMODWebApp.webservice.AMODWebAPI.CaseAddAttachment (int *caseId*, string *name*, byte[] *value*)

Adds attachment to specified case.

Parameters:

<i>caseId</i>	Id of the case
<i>name</i>	Name of the attachment
<i>value</i>	Attachment content

Returns:

Empty string if success, otherwise appropriate message

string AMODWebApp.webservice.AMODWebAPI.CaseAddComment (int *caseId*, string *comText*)

This method adds comment to specified case.

Parameters:

<i>caseId</i>	Id of the case
<i>comText</i>	Comment content

Returns:

Empty string if success, otherwise appropriate message

string AMODWebApp.webservice.AMODWebAPI.CaseAddUser (int *caseId*, string *ccEmail*, bool *addCCnotContrib*, string *comment*)

Adds specified users as CC or Contributors. If needed, creates new user account and sends invite email.

Parameters:

<i>caseId</i>	The id of the case
<i>ccEmail</i>	Email list, separated by ','
<i>addCCnotContrib</i>	True if users are to be added as CC, false if as CON
<i>comment</i>	Additional comment

Returns:

Empty string if successful, appropriate message otherwise

string AMODWebApp.webservice.AMODWebAPI.CaseForward (int *caseId*, string *email*, string *comment*)

Adds user specified by email adress to specified case as CC.

Parameters:

<i>caseId</i>	Id of the case
<i>email</i>	Email of user to be added as CC.
<i>comment</i>	Additional comment to be added to the case.

Returns:

Empty string if successful, appropriate message otherwise

string AMODWebApp.webservice.AMODWebAPI.CloseCase (int *caseId*, string *dateOfModification*)

Closes specified case

Parameters:

<i>caseId</i>	Id of the case
<i>dateOfModification</i>	Last date of case modification

Returns:

XML formatted response, indicating wheter operation was successful or not

string AMODWebApp.webservice.AMODWebAPI.CreateCase (string *procedureTitle*, string *startingStateTitle*, string *caseTitle*)

This method creates a new case.

Parameters:

<i>procedureTitle</i>	Name of existing or to be created procedure
<i>startingStateTitle</i>	Name of starting stage

<i>caseTitle</i>	Name of created case
------------------	----------------------

Returns:

XML file with case Id, otherwise appropriate message

void AMODWebApp.webservice.AMODWebAPI.DeleteAllTableFieldRecords (int *procedureWithTableFieldId*, string *tableFieldName*)

Removes all rows from table placed in field with specified name in specified procedure (for every case)

Parameters:

<i>procedureWithTableFieldId</i>	Id of the procedure containing table field
<i>tableFieldName</i>	Name of the field containing table

void AMODWebApp.webservice.AMODWebAPI.DeleteTableFieldRecordsConnectedToCase (int *procedureWithTableFieldId*, string *tableFieldName*, int *caseId*)

Removes all rows from table placed in field with specified name in specified procedure (for specified case)

Parameters:

<i>procedureWithTableFieldId</i>	Id of the procedure containing table field
<i>tableFieldName</i>	Name of the field containing table
<i>caseId</i>	Id of the case containing table

byte [] AMODWebApp.webservice.AMODWebAPI.FindCase (int *caseId*, string *caseTitle*, string *procedureTitle*, int *limit*)

This method gets a list of cases matching given filter.

Parameters:

<i>caseId</i>	case identifier
<i>caseTitle</i>	case title filter
<i>procedureTitle</i>	procedure title filter
<i>limit</i>	maximum number of cases returned

Returns:

Compressed XML file with list of cases.

string AMODWebApp.webservice.AMODWebAPI.ForwardCase (int *caseId*, string *stage*, string *user*, bool *ignoreRestrictions*, bool *sendEmail*)

Forwards case to specified user.

Parameters:

<i>caseId</i>	Id of the case to be forwarded
<i>stage</i>	Stage to be changed to

<i>user</i>	User login to whom the case is to be forwarded to
<i>ignoreRestrictions</i>	True if case should be forwarded despite restrictions, false otherwise
<i>sendEmail</i>	True if notification email should be sent, false otherwise

Returns:

Empty string on success, otherwise an appropriate message

string AMODWebApp.webservice.AMODWebAPI.GenerateDefaultTitle (string *prcTitle*)

This method returns default case title for given procedure name.

Parameters:

<i>prcTitle</i>	procedure title
-----------------	-----------------

Returns:

XML file with default case title.

string AMODWebApp.webservice.AMODWebAPI.GetAMODEmail ()

This method gets AMODIT email address.

Returns:

XML file with an email address.

string AMODWebApp.webservice.AMODWebAPI.GetCase (int *caseId*)

This method allow to access to single case by Id.

Parameters:

<i>caseId</i>	
---------------	--

Returns:

XML file with description of parameters from AMODCase

string AMODWebApp.webservice.AMODWebAPI.GetCaseIdByFieldValue (string *fieldName*, string *fieldValue*)

Gets id of the case by the value of specified field

Parameters:

<i>fieldName</i>	specified field name
<i>fieldValue</i>	specified field value

Returns:

id of first case with the value in the field, when no cases found returns -1

string AMODWebApp.webservice.AMODWebAPI.GetCasesByFieldValue (string *fieldName*, string *fieldValue*)

Get case(s) id(s) by value in specified field

Parameters:

<i>fieldName</i>	Specified field name
<i>fieldValue</i>	value in field

Returns:

string with ids (separated with comma) of cases containing the specified field with the given value.

DataSet AMODWebApp.webservice.AMODWebAPI.GetCasesByTypeId (int *typeId*, int *scheduleTypeId*, string *caseStatus*, StringCollection *fieldMapping*, StringCollection *fieldsToInclude*)

This function is used by QNT for export application.

Parameters:

<i>typeId</i>	
<i>scheduleTypeId</i>	
<i>caseStatus</i>	
<i>fieldMapping</i>	
<i>fieldsToInclude</i>	

Returns:

Dataset object

string AMODWebApp.webservice.AMODWebAPI.GetCurrentUserEmails ()

Get all (possibly 4) e-mail addresses for the currently authenticated user.

Returns:

Xml file with list of e-mail addresses.

string AMODWebApp.webservice.AMODWebAPI.GetFormByFieldProcedureType (int *caseId*, int *typeId*)

Gets ID of form containing table with specified procedure type id on specified case form

Parameters:

<i>caseId</i>	Id of the case to search on
<i>typeId</i>	Id of table form procedure

Returns:

Id of table form procedure, or -1 when not found

string AMODWebApp.webservice.AMODWebAPI.GetMostUsedProcedures (int *userId*, int *rowLimit*)

This method gets a list of the most used procedure for specified user.

Parameters:

<i>userId</i>	The id of the user
<i>rowLimit</i>	Maximum number of returned procedures

Returns:

XML file with list of the most used procedure names, ids and information if user can add new stages.

string AMODWebApp.webservice.AMODWebAPI.GetMostUsedProceduresForCurrentUser (int *rowLimit*)

This method gets a list of the most used procedures for currently authenticated user.

Parameters:

<i>rowLimit</i>	
-----------------	--

Returns:

XML file with list of the most used procedure names, ids and information if user can add new stages.

string AMODWebApp.webservice.AMODWebAPI.GetProcedure (string *title*)

This method gets a procedure which name matches specified string.

Parameters:

<i>title</i>	The title to be matched
--------------	-------------------------

Returns:

XML file with basic procedure parameters.

DataTable AMODWebApp.webservice.AMODWebAPI.GetProcedureForms (int *prcId*)

Gets all procedures connected to specified one, defined as a form

Parameters:

<i>prcId</i>	id of specified procedure
--------------	---------------------------

Returns:

DataTable object containing all form procedures

string AMODWebApp.webservice.AMODWebAPI.GetProcedures (string *prefixText*, int *maxProcedures*)

This method gets a list of procedure names matching specified string.

Parameters:

<i>prefixText</i>	
<i>maxProcedures</i>	Maximum number of returned procedures

Returns:

XML file with list of procedure names and ids.

DataTable AMODWebApp.webservice.AMODWebAPI.GetProceduresByType (int *typeId*)

Used by QNT app in configuration options

Parameters:

<i>typeId</i>	Id of type
---------------	------------

Returns:

DataTable with all active procedures of specified type

string AMODWebApp.webservice.AMODWebAPI.GetProcedureSchema (int *procedureId*)

Gets procedure schema by procedure id

Parameters:

<i>procedureId</i>	The id of the procedure
--------------------	-------------------------

Returns:

XML formatted list of procedure parameters

string AMODWebApp.webservice.AMODWebAPI.GetProcedureSchemaByTitle (string *procedureTitle*)

Gets procedure schema for procedure that name matches specified string

Parameters:

<i>procedureTitle</i>	The name to match
-----------------------	-------------------

Returns:

XML formatted list of procedure parameters

DataTable AMODWebApp.webservice.AMODWebAPI.GetProcedureTableProcedures (int *prId*)

Collects all procedures of specified procedure used to store table fields

Parameters:

<i>prId</i>	specified procedure id
-------------	------------------------

Returns:

DataTable with table procedures

string AMODWebApp.webservice.AMODWebAPI.GetUser (int *userId*)

Gets user info

Parameters:

<i>userId</i>	Id of the user
---------------	----------------

Returns:

XML formatted user property list

string AMODWebApp.webservice.AMODWebAPI.GetUserCases ()

This method gets all cases which are assigned to user.

Returns:

XML file with full description of Cases which are assigned to user. List of all cases with list of comments and descriptions of main form.

void AMODWebApp.webservice.AMODWebAPI.LogToAMODLog (string *message*, string *callStack*)

Logs specified message and callstack to AMOD Logs

Parameters:

<i>message</i>	The message to be logged
<i>callStack</i>	The call stack to be logged

string AMODWebApp.webservice.AMODWebAPI.SendCase (int *caseId*, int *stageId*, string *dateOfModification*, string *userDescription*, string *comment*)

Transfers (forwards) case to another user, possibly changing its state and adding a comment

Parameters:

<i>caseId</i>	Id of the case to be forwarded
<i>stageId</i>	Id of the stage to be changed to
<i>dateOfModification</i>	Last date of modification in "yyyy-MM-dd HH:mm:ss" format
<i>userDescription</i>	Login of the user to whom the case is to be forwarded
<i>comment</i>	Value of the comment to be added

Returns:

Operation summary in xml form

bool AMODWebApp.webservice.AMODWebAPI.SetCaseAsExported (int *caseID*)

It's marks object as exported to QNT.

Parameters:

<i>caseID</i>	Id number of case
---------------	-------------------

Returns:

True if case was set, false otherwise.

void AMODWebApp.webservice.AMODWebAPI.SetProcedureType (int *prcId*, int *prcTypeId*)

Sets type of specified AMODprocedure

Parameters:

<i>prcId</i>	specified procedure id
<i>prcTypeId</i>	specified type id

string AMODWebApp.webservice.AMODWebAPI.UpdateCase (int *caseId*, string *dateOfModification*, int *paramId*, String *value*)

Updates parameter with paramId with value in case with caseId

Parameters:

<i>caseId</i>	Id of the case where the field is to be updated
<i>dateOfModification</i>	Last date of modification in "yyyy-MM-dd HH:mm:ss" format
<i>paramId</i>	Id of parameter to be updated
<i>value</i>	New value to be set

Returns:

Update summary in xml form

void AMODWebApp.webservice.AMODWebAPI.UpdateCaseField (int *caseId*, string *FieldName*, string *fieldValue*)

Updates specified case single field value

Parameters:

<i>caseId</i>	Id of AMOD Case containing field to be updated
<i>FieldName</i>	Name of the field to be updated
<i>fieldValue</i>	New field value

void AMODWebApp.webservice.AMODWebAPI.UpdateCaseFields (int *caseId*, string[] *fieldNames*, string[] *fieldsValues*)

WARNING! This function is a little bit deprecated. Use UpdateCaseFields2() instead.

Updates specified case multiple fields values

Parameters:

<i>caseId</i>	Id of AMOD Case containing field to be updated
<i>fieldNames</i>	Names of the fields to be updated

<i>fieldsValues</i>	New fields values
---------------------	-------------------

void AMODWebApp.webservice.AMODWebAPI.UpdateCaseFields2 (int caseId, string[] fieldNames, string[] fieldsValues)

WARNING! It is highly recommended to use this function.

Updates specified case multiple fields values

Parameters:

<i>caseId</i>	Id of AMOD Case containing field to be updated
<i>fieldNames</i>	Names of the fields to be updated
<i>fieldsValues</i>	New fields values

Returns:

Update status in xml form.

1) On success this method returns:

```
<UpdateCaseFields2Result>
  <Response>success</Response>
</UpdateCaseFields2Result>
```

2) In case of any errors this method returns:

```
<UpdateCaseFields2Result>
  <Response>fail</Response>
  <Errors>
    <UpdateError>
      <ErrorName>incorrectvalue</ErrorName>
      <ErrorField>A document class</ErrorField>
      <ErrorValue>HR document D1</ErrorValue>
    </UpdateError>
  </Errors>
</UpdateCaseFields2Result>
```

A list of returned error names in the <ErrorName/> tag:

- unauthorized: user not logged-in, a reason placed in <ErrorDescription/> tag;
- incorrectvalue: a wrong field value, a field name in <ErrorField/>, a wrong value in <ErrorValue/> tag;
- missingfield: there is no such field on the form, a wrong field name in <ErrorField/> tag;
- casenotopened: there is no such case or user has no rights, a reason placed in <ErrorDescription/> tag;
- updatefailed: an error has occurred while updating case, a reason placed in <ErrorDescription/> tag;
- unexpected: an exception, a reason placed in <ErrorDescription/> tag.

A list of case errors that may appear in <ErrorDescription/> tag:

```
LOCK,
NO_MODIFY_PERMISSIONS,
ACTIVE_ACCEPTANCE,
NEXT_OWNER_NOT_ALLOWED,
NEXT_STATE_NOT_ALLOWED,
NEXT_OWNER_NOT_FOUND,
NEXT_STATUS_NOT_FOUND,
```

REQUIRED_PARAMETER_EMPTY,
REQUIRED_ATTACHMENT_MISSING,
REQUIRED_FORM_MISSING,
RULES_ERROR,
EDIT_LOCK.

Remarks:

If there is a document type field on the case form, you can also send a file with its name and content (Base64 encoded) while using this method. In the fieldsValues parameter you should add the following XML code:

```
<fieldsValues>  
  <string>a file name;Base64 encoded content</string>  
</fieldsValues>
```

string AMODWebApp.webservice.AMODWebAPI.UserEmailExists (string *email*)

Check if user with given e-mail address exists.

Parameters:

<i>email</i>	E-mail address to be searched for
--------------	-----------------------------------

Returns:

"true" if user exists or "false" otherwise