

Processes Online User Guide

Bizproapi & Bizfloapi

1 INTRODUCTION

The terms *class*, and *entity type*, are synonymous. Class names are alphanumeric, with the first character being alphabetic.

At the time of definition of a class, one or more of its attributes would have been defined; these are called *standard attributes*. If any additional attributes are defined for a class, i.e., subsequent to the class definition, then these additional attributes are called *custom attributes*. Attribute names should be alphanumeric, with the first character being alphabetic.

Exactly one of the standard attributes of a class is designated as its *key*. No two objects belonging to a class can have the same key value.

Attributes can be of one of the following datatypes:-

- Integer
- Real
- Money
- Date
- String
- Set
- Email
- Object

Note that all attribute values are *internally* stored in the *Processes Online* database as text strings. *Integer* and *Real* data, once extracted from the database, should be parsed using the methods, `Integer.parseInt(Integer data)` and `Double.parseDouble(Real data)`, respectively. Conversely, before storing Java *int* and *double* data, use the methods, `Integer.toString(int data)` and `Double.toString(double data)`, and then store in the *Processes Online* database. *Money* data is stored as a string of decimal digits, with exactly one decimal point, which must be followed by exactly two decimal digits. *Date* data is in the format, dd-*MMM*-yyyy, e.g., 01-Aug-1908, or 21-Jun-1987. As can be seen in the above list, there is no Boolean datatype; if required you can define a Set datatype, with set elements being (Yes, No). Note that internally values of standard attributes are restricted to 1500 bytes, while those of custom attributes can be upto 1 megabyte.

By default, every object has two attributes of XML datatype; these are simply referred to as DOM1 and DOM2. The maximum permissible size of DOM1 and DOM2 is 1 MB each.

By default again, every object has an unlimited number of attributes of XML datatype; these are simply referred to as DOMn, where n ranges from 3 upwards. The maximum permissible size of DOMn is 1 MB for each value of n.

2 BIZPROAPI

Bizproapi comprises of a number of methods (APIs) as tabulated hereunder. (The complete list of APIs will be found in the corresponding Javadoc). These APIs are used for web application development against the *Processes Online* object-oriented database. **The best way of learning how to use these APIs is to go over *Processes Online* User Guide – Web Application Development Tutorial.**

Type	Method and Description
String	<p>assignCustomAttVals(String entityType, String entityID, String customAttName, String[] vals)</p> <p>This method assigns values to the custom attribute of a given entity of a given entity type</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity customAttName - The name of the custom attribute vals - An array of values for the custom attribute <p>Returns:</p> <ul style="list-style-type: none"> A message heralding successful assignment or otherwise
String	<p>createObject(String entityType, String[] atts, String[] attVals)</p> <p>This method creates an entity of the given entity type</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The given entity type atts - An array of standard attributes of the given entity type, the very first of which is the key attVals - An array of proposed values for the corresponding standard attributes in the array, atts

	<p>Returns:</p> <p>Returns OK if the entity was created and NOTOK if the entity did not get created.</p>
String	<p>deassignCustomAttVals(String entityType, String entityID, String customAttName)</p> <p>This method deassigns the values currently assigned to a custom attribute of an entity</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity customAttName - The name of the custom attribute <p>Returns:</p> <p>A message heralding successful deassignment</p>
String	<p>deleteEntity(String entityType, String entityID)</p> <p>This method deletes an entity of a given entity type</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The given entity type entityID - The entity ID of the entity to be deleted <p>Returns:</p> <p>Returns OK if deletion is successful, returns NOTOK if deletion is not successful</p>
void	<p>deletePastRecords(String entityType, String cutoffDate)</p> <p>Deletes financial transaction records of all objects, or entities, belonging to a given class prior to a given cutoff date; by default, upto 1000 past transaction records will be stored</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The name of the class, or entity type cutoffDate - The cutoff date in dd-MMM-YYYY format, e.g., 01-Feb-2016
void	<p>emailPassword(String entityType, String entityID)</p>

	<p>Emails the current password of an entity to his / her email</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - Should be one of Employee, Customer, Vendor, Faculty, Student, Doctor, Patient, Member entityID - The entity ID of the entity
String[]	<p>getAttVals(String entityType, String attrName2, String attrName1, String attrVal1)</p> <p>This method lists the values of a second attribute of entities of an entity type, given the value of a first attribute</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The given entity type attrName2 - The name of the second attribute attrName1 - The name of the first attribute attrVal1 - The given value of the first attribute <p>Returns:</p> <ul style="list-style-type: none"> An array of values of the second attribute
String	<p>getCurrency()</p> <p>This method returns the home currency symbol of the organization</p>
String	<p>getID(String entityType)</p> <p>This method generates a named series of Long integer, 0,1,2, etc.</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The name of the series, can be one of Member, Customer, Vendor, Student, Faculty, or any custom class name <p>Returns:</p> <ul style="list-style-type: none"> The next member of the series
String[]	<p>getCustomAttVals(String entityType, String entityID, String customAttName)</p> <p>This method returns the values of a custom attribute of an entity of a given entity type</p>

	<p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity customAttName - The name of the custom attribute <p>Returns:</p> <p>An array of values of the custom attribute of the entity</p>
void	<p>initializeDOM(String entityType, String entityID)</p> <p>This method initializes the first XML Document, DOM1, associated with an entity, by way of creating a document with only the root node.</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity
void	<p>initializeDOM2(String entityType, String entityID)</p> <p>This method initializes the second XML Document, DOM2, associated with an entity, by way of creating a document with only the root node.</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity
void	<p>initializeXMLData(String entityType, String entityID, int n)</p> <p>This method initializes the XML Document, DOMn, associated with an entity, where n ranges from 3 upwards, by way of creating a document with only the root node.</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity
org.w3c.dom.Document	<p>getDOM(String entityType, String entityID)</p> <p>This method retrieves the first XML Document, DOM1, associated with an entity</p>

	<p>Parameters: entityType - The entity type of the entity entityID - The entity ID of the entity</p> <p>Returns: Returns the first XML Document associated with the entity</p>
org.w3c.dom.Document	<p>getDOM2(String entityType, String entityID)</p> <p>This method retrieves the second XML document, DOM2, associated with an entity</p> <p>Parameters: entityType - The entity type of the entity entityID - The entity ID of the entity</p> <p>Returns: Returns the second XML document associated with the entity</p>
org.w3c.dom.Document	<p>getXMLData(String entityType, String entityID, int n)</p> <p>This method retrieves the XML document, DOMn, associated with an entity, where n ranges from 3 upwards</p> <p>Parameters: entityType - The entity type of the entity entityID - The entity ID of the entity n – The n-th attribute of datatype XML</p> <p>Returns: Returns the n-th XML document associated with the entity</p>
String	<p>getKey(String entityType)</p> <p>This method retrieves the name of the key attribute of a given entity type</p> <p>Parameters:</p>

	<p>entityType - The given entity type, or class</p> <p>Returns:</p> <p>The key attribute name</p>
String	<p><u>getMyNameFromEmail</u>(String entityType, String email)</p> <p>This method retrieves the full name of an entity given its entity type and email ID</p> <p>Parameters:</p> <p>entityType - The entity type of the entity; must be one of Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient</p> <p>email - The email ID of the entity</p> <p>Returns:</p> <p>Returns the full name, i.e., first name followed by last name</p>
String	<p><u>getMyNameFromID</u>(String entityType, String entityID)</p> <p>This method retrieves the full name of an entity, given its entity type and entity ID</p> <p>Parameters:</p> <p>entityType - The entity type of the entity; must be one of Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient</p> <p>entityID - The entity ID of the entity</p> <p>Returns:</p> <p>The full name of the entity, i.e., first name followed by the last name</p>
String	<p><u>whenCreated</u> (String entityType, String entityID)</p> <p>This method retrieves the date and time at which an entity was created</p> <p>Parameters:</p> <p>entityType - The entity type of the entity</p> <p>entityID - The entity ID of the entity</p>

	<p>Returns:</p> <p>The date and time of creation of the entity in the format dd- MMM-yyyy::HH:mm:ss</p>
String[]	<p>getObjectIDs(String entityType)</p> <p>This method returns the entity IDs of all the entities, or objects, belonging to the given entity type</p> <p>Parameters:</p> <p>entityType - The given entity type</p> <p>Returns:</p> <p>An array of entity IDs; ignore the first element, which is the name of the key attribute of the given entity type</p>
String	<p>getOrgAddress()</p> <p>This method returns the address of the organization</p>
String	<p>getOrgName()</p> <p>This method returns the name of the organization</p>
String[]	<p>getSelectAttVals(String entityType, String[] atts, String pstan)</p> <p>This method lists the values of selected standard and custom attributes of all entities of a given entity type</p> <p>Parameters:</p> <p>entityType - The given entity type</p> <p>atts - An array of attribute names, the first few of which are standard attributes, and the last few are custom attributes</p> <p>pstan - The number of standard attribute names in the array, atts</p> <p>Returns:</p> <p>An array of attribute values, val(1,1), ..., val(1,m),val(2,1), ..., val(2,m), ..., val(n,1), ..., val(n,m), where n = number of entities of the given entity type, and m = length of the array, atts. In case a custom attribute value is returned as --, it means the value is null.</p>
String	<p>getSelection(String entityType)</p>

	<p>This method returns the entity IDs of a given entity type in a format suitable for placement between select tags on a html page</p> <p>Parameters: entityType - The given entity type</p> <p>Returns: A String in the format: <option>EntityID1<option>EntityID2 ...</p>
String	<p>getSelection2(String entityType, String attrName)</p> <p>This method returns the entity IDs, and corresponding values of one attribute, of a given entity type in a format suitable for placement between select tags on a html page</p> <p>Parameters: entityType - The given entity type attrName - The name of one attribute of the given entity type</p> <p>Returns: A String in the format: <option value=" EntityID1"> EntityID1::attrVal1<option value=" EntityID2"> EntityID2::attrVal2...</p>
void	<p>intimateDues(String entityType)</p> <p>Intimates by email the current dues to all entities of the given entity type</p> <p>Parameters: entityType - The given entity type, must be one of Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient</p>
String	<p>makeMoney(double anAmount)</p> <p>This method converts from double to a String in <i>Money</i> format</p> <p>Parameters:</p>

	<p>anAmount</p> <p>Returns:</p> <p>Returns a money formatted String</p> <p>Note: This API is deprecated; instead use the method <code>String.format("%.2f",double anAmount)</code></p>
double	<p>retrieveBalance(String entityType, String entityID)</p> <p>Returns the balance in the account of an entity</p> <p>Parameters:</p> <p>entityType - The entity type of the entity</p> <p>entityID - The entity ID of the entity</p> <p>Returns:</p> <p>The balance in the account of the entity as a real number</p>
String[]	<p>retrievePassbook(String entityType, String entityID)</p> <p>This method retrieves the passbook, or financial statement, of an entity</p> <p>Parameters:</p> <p>entityType - The entity type of the entity</p> <p>entityID - The entity ID of the entity</p> <p>Returns:</p> <p>The passbook in array format The first element of the array is the current balance in the format: xx::yy, where xx is Credit or Debit, yy is the amount in money format Now follow the passbook entries Each passbook entry constitutes 4 elements of the array in the order: Date, Credit or Debit, Amount, Particulars</p>
boolean	<p>roleAuthorized(String entityType, String emailID, String role)</p> <p>Parameters:</p> <p>entityType - The entity type of an entity; must be one of Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient</p> <p>emailID - The email ID of the entity</p>

	<p>role - A custom attribute name</p> <p>Returns:</p> <p>true if the attribute value is Yes</p>
String	<p>sendAdhocMail(String recipient, String subject, String content)</p> <p>This method send an email to a recipient</p> <p>Parameters:</p> <p>recipient - The email ID of the recipient</p> <p>subject - The subject of the email</p> <p>content - The content of the email</p> <p>Returns:</p> <p>A message heralding whether the email was sent or not</p>
String	<p>sendMail(String entityType, String entityID, String subject, String content)</p> <p>This method sends an email to an entity</p> <p>Parameters:</p> <p>entityType - The entity type of the entity, which should be one of Employee, Customer, Vendor, Member, Doctor, Patient, Faculty or Student</p> <p>entityID - The entity ID of the entity</p> <p>subject - The subject of the email</p> <p>content - The content of the email</p> <p>Returns:</p> <p>A message heralding whether the email was sent or not</p>
void	<p>sendStatement(String entityType, String startDate)</p> <p>Sends financial statement by email to all entities of the give class, or entity type</p> <p>Parameters:</p>

	<p>entityType - The given entity type, must be one of Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient</p> <p>startDate - The start date of the financial statement in dd- MMM-YYYY format, e.g., 01-Feb-2016</p>
void	<p>setCurrency(String currency)</p> <p>Sets the home currency symbol of the organization</p> <p>Parameters:</p> <p>currency - The given home currency symbol, which should be exactly one of the following 3-letter symbols: USD (U.S. Dollar), AUD (Australian Dollar), GBP (Pound Sterling), CAD (Canadian Dollar), CZK (Czech Koruna), DKK (Danish Krone), EUR (Euro), HKD (Hong Kong Dollar), HUF (Hungarian Forint), JPY (Japanese Yen), NZD (New Zealand Dollar), NOK (Norwegian Krone), PLN (Polish Zloty), SGD (Singapore Dollar), CHF (Swiss Franc), INR (Indian Rupee), BRL (Brazilian Real), ILS (Israeli New Sheqel), MYR (Malaysian Ringgit), MXN (Mexican Peso), PHP (Philippine Peso), SEK (Swedish Krona), TWD (Taiwan New Dollar), THB (Thai Baht)</p>
String	<p>updateAccount(String entityType, String entityID, String type, String amount, String particulars)</p> <p>This method records a credit or debit made to the account of an entity; it removes the earliest passbook entry, in case their number exceeds 1000</p> <p>Parameters:</p> <p>entityType - The entity type of the entity</p> <p>entityID - The entity ID of the entity</p> <p>type - The type of the transaction, Credit or Debit</p> <p>amount - The amount of the transaction in Money format</p> <p>particulars - The particulars of the transaction</p> <p>Returns:</p> <p>Return is in the format: TypeOfBalanceBefore::BalanceBefore::TypeOfBalanceAfter::BalanceAfter</p>
String	<p>updateAttVal(String entityType, String entityID, String attName, String attVal)</p>

	<p>This method will update the value of a standard attribute of a given entity of a given entity type</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The given entity type entityID - The entity ID of the given entity attName - The name of the attribute attVal - The value to which the attribute is to be updated <p>Returns:</p> <p>Returns OK if the update is successful; returns NOTOK otherwise</p>
String	<p>updateDOM(String entityType, String entityID, org.w3c.dom.Document doc)</p> <p>This method updates the first XML Document, DOM1, associated with an entity</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity doc - The XML Document to be updated <p>Returns:</p> <p>Returns a success message</p>
String	<p>updateDOM2(String entityType, String entityID, org.w3c.dom.Document doc)</p> <p>This method updates the second XML Document, DOM2, associated with an entity</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity doc - The XML Document to be updated <p>Returns:</p>

	Returns a success message
String	<p>updateXMLData(String entityType, String entityID, org.w3c.dom.Document doc, int n)</p> <p>This method updates the XML Document, DOMn, associated with an entity</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity entityID - The entity ID of the entity doc - The XML Document to be updated n – The n-th attribute of datatype, XML; n = 3 upwards <p>Returns:</p> <p>Returns a success message</p>
String	<p>verifyPassword(String entityType, String email, String password)</p> <p>This method verifies the password of a loggable entity</p> <p>Parameters:</p> <ul style="list-style-type: none"> entityType - The entity type of the entity; must be one of Employee, Customer, Vendor, Member, Faculty, Student, Doctor, Patient email - The email ID of the entity password - The password of the entity <p>Returns:</p> <p>Returns OK if verified, returns NOTOK if not verified</p>

3 BIZFLOAPI

Bizfloapi comprises of a number of methods (APIs) as tabulated hereunder. (The complete list of APIs will be found in the corresponding Javadoc). These APIs are used for automating business

processes. **The best way of learning how to use these APIs is to go over Processes Online User Guide – Process Automation Tutorial.** We strongly recommend your doing this, and returning to this document for gaining a better understanding of the APIs.

A business process instance is an entity, or object, belonging to the built-in class, ActiveProcess. When a process instance is archived, it becomes an object belonging to the built-in class, ArchivedProcess.

Type	Method and Description
void	<p>archiveProcess(String objectid)</p> <p>This method archives a business process instance</p> <p>Parameters: objectid - The unique id of the business process instance to be archived</p>
String	<p>updateProcessInstance(String objectid, String pathid, String from, String[] captions, String[] values, String to)</p> <p>This method updates a business process instance with a new segment, and notifies the to entities by email</p> <p>Parameters: objectid - The unique ID of the business process instance pathid – The unique ID of an existing flowpath which is to be updated from - The From part of the new segment captions - The captions part of the new segment content values - The values part of the new segment content to - The To part of the new segment and the new flowpath(s)</p> <p>Returns: A message that the required update has been made</p>
String	<p>updateProcessInstanceSilent(String objectid, String pathid, String from, String[] captions, String[] values, String to)</p> <p>This method updates a business process instance with a new segment, but does not notify the to entities by email</p>

	<p>Parameters:</p> <ul style="list-style-type: none"> objectid - The unique ID of the business process instance pathid – The unique ID of an existing flowpath which is to be updated from - The From part of the new segment captions - The captions part of the new segment content values - The values part of the new segment content to - The To part of the new segment and the new flowpath(s) <p>Returns:</p> <p>A message that the required update has been made</p>
String	<p>getSubject(String objectid)</p> <p>This method retrieves the Subject of a business process instance</p> <p>Parameters:</p> <ul style="list-style-type: none"> objectid - The unique ID of the business process instance <p>Returns:</p> <p>The Subject</p>
String[]	<p>getOriginator(String objectid)</p> <p>This method retrieves the entity type and entity ID of the originator of a business process instance</p> <p>Parameters:</p> <ul style="list-style-type: none"> objectid - The unique ID of the business process instance <p>Returns:</p> <p>A two-element array; first element is entity type; the second element is entity ID</p>
String[]	<p>getPastMessages(String objectid, String pathid)</p> <p>This method returns the messages from previous actors, or entities, along a flowpath of a process instance.</p> <p>Parameters:</p> <ul style="list-style-type: none"> objectid - The unique ID of the process instance pathid - The unique ID of the flowpath <p>Returns:</p>

	<p>An array, each element of which comprises of a sequence of data items separated by :::. The first data item is the unique ID and name of the entity who sent the messages. The second data item is the date and time when the messages were sent. The following data items are the messages themselves. Each message consists of a caption and a value, separated by ~~</p>
String	<p>initiateProcessInstance(String appid, String entityType, String entityID, String subject)</p> <p>This function creates an instance of a given business process</p> <p>Parameters:</p> <ul style="list-style-type: none"> appid - The unique ID of the business process (NOT the process instance, which is yet to be created) entityType - The entity type of the entity initiating the business process entityID - The entity ID of the entity initiating the business process subject - The subject of the business process instance <p>Returns:</p> <p>Returns the unique ID of the initiated process instance</p>
String[]	<p>getPendingParameters(String appid, String entityType, String email)</p> <p>This method returns the parameters of process instances pending for action by a given entity</p> <p>Parameters:</p> <ul style="list-style-type: none"> appid - The id of a given business process entityType - The entity type of the entity; should be one of Employee, Customer, Vendor, Member, Doctor, Patient, Faculty or Student email - The email id of the entity <p>Returns:</p> <p>An array, one element per pending process instance, the format of the element being ProcessInstanceID::FlowpathID::PendingStateID::Subject</p>
String[]	<p>getMessageValues(String objectid, String pathid, String givenCaption)</p>

Parameters:

objectid - The unique ID of a business process instance

pathid - The unique ID of the path along which the process instance has flowed to the current actor

givenCaption - A message caption

Returns:

An array of all message values corresponding to the given caption