**ОПИСАНИЕ KBD API**

**ДЛЯ ПОЛЬЗОВАТЕЛЕЙ**

**БАНК ДОКУМЕНТОВ 2.8**

**ИЮНЬ 2020**

Оглавление

[ОБЩАЯ ИНФОРМАЦИЯ 4](#_Toc51676915)

[Параметры запросов 4](#_Toc51676916)

[ДОКУМЕНТЫ, ФАЙЛЫ (KBD API\_docs) 6](#_Toc51676917)

[GetCardAttributes 6](#_Toc51676918)

[GetServiceAttributes 7](#_Toc51676919)

[SetServiceAttributes 8](#_Toc51676920)

[GetClassificatorDesc 9](#_Toc51676921)

[GetStatusSettings 10](#_Toc51676922)

[CreateCard 10](#_Toc51676923)

[CheckRequestStatus 12](#_Toc51676924)

[GetDocumentDesc 13](#_Toc51676925)

[GetDocumentItems 14](#_Toc51676926)

[EditCard 15](#_Toc51676927)

[GetEditionDesc 17](#_Toc51676928)

[SetText 18](#_Toc51676929)

[GetLinkedDocsAttributes 19](#_Toc51676930)

[AddAttachment 20](#_Toc51676931)

[GetDocumentAttachmentsDesc 21](#_Toc51676932)

[EditAttachments 22](#_Toc51676933)

[DeleteAttachments 24](#_Toc51676934)

[CreateEdition 25](#_Toc51676935)

[RemoveDocument 26](#_Toc51676936)

[SetDraftModeOn / SetDraftModeOff 27](#_Toc51676937)

[SetKodeksAttachModeOn / SetKodeksAttachModeOff 28](#_Toc51676938)

[SetKodeksStatus 29](#_Toc51676939)

[ТЕРМИНЫ (KBD API\_terms) 31](#_Toc51676940)

[CreateTerm 31](#_Toc51676941)

[CheckRequestStatus 32](#_Toc51676942)

[GetDefDescByGuid 33](#_Toc51676943)

[GetDefDescByLink 34](#_Toc51676944)

[GetDefDescByName 34](#_Toc51676945)

[GetTermDescByGuid 35](#_Toc51676946)

[GetTermDescByName 36](#_Toc51676947)

[GetDefinitionDuplicates 37](#_Toc51676948)

[AddTermDefinition 38](#_Toc51676949)

[AssignTerms 39](#_Toc51676950)

[DeleteTerm 40](#_Toc51676951)

[DeleteTermDefinition 40](#_Toc51676952)

[EditTermDefinition 41](#_Toc51676953)

[EditTermName 42](#_Toc51676954)

[RebindTermDefinition 43](#_Toc51676955)

[ФУНКЦИИ ОБЩЕГО НАЗНАЧЕНИЯ (common) 45](#_Toc51676956)

[SynchroniseVolumes 45](#_Toc51676957)

[СТРУКТУРЫ 47](#_Toc51676958)

[CardAttribute 47](#_Toc51676959)

[ClassificatorDesc 47](#_Toc51676960)

[ClassificatorValue 47](#_Toc51676961)

[StatusSettings 48](#_Toc51676962)

[DocumentDesc 48](#_Toc51676963)

[DocumentItems 49](#_Toc51676964)

[AttachmentItem 49](#_Toc51676965)

[AttachmentDesc 50](#_Toc51676966)

[EditionDesc 51](#_Toc51676967)

[LinkedDocsAttributes 51](#_Toc51676968)

[DefinitionDesc 51](#_Toc51676969)

[FullDefinitionDesc 52](#_Toc51676970)

[DuplicateTermDesc 52](#_Toc51676971)

# ОБЩАЯ ИНФОРМАЦИЯ

Данный сервис предоставляет возможность внешним разработчикам программными средствами получать и записывать информацию в базы данных системы.

**KBD API** представляет собой веб-сервис и определяет набор функций, с помощью которых разработчики могут выполнять запросы добавления, получения, изменения и удаления данных. Взаимодействие происходит по протоколу SOAP поверх HTTP.

Все вызовы функций KBD API – это HTTP-запрос, в заголовке которого есть поле SOAPAction, определяющее вызываемую функцию, а в теле – SOAP-сообщение со значениями параметров функции.

Генерировать подобные запросы на основе WSDL-описания веб-сервиса возможно с помощью SOAP-клиентов, написанных на различных языках программирования.

Формат url запроса для получения wsdl-описания:

*http://<имя сервера>[:<порт>]/<виртуальный каталог КБД>/KBD API\_<группа функций>/?wsdl&use=encoded&api=1*

Для доступа к функциям KBD API требуется авторизация по базе операторов АРМ. Для этого при построении клиента необходима генерация кода взаимодействия с сервером, требующим авторизации.

Пример создания SOAP-клиента на языке PHP для использования API-функций:

|  |
| --- |
| <?php    // url запроса wsdl-описания  $wsdl = 'http://localhost:8020/eda/KBD API\_docs?wsdl&use=encoded&api=1';    // создание SOAP-клиента на основе wsdl-описания с авторизацией  $sclient = new SoapClient($wsdl,  array(  'encoding' => 'UTF-8',  'user\_agent' => 'PHP/SOAP',  'exceptions' => true,  'login' => 'user', *// логин оператора КБД*  'password' => 'password' *// пароль оператора КБД*  )  );    // ...    ?> |

## Параметры запросов

В документации определены параметры для каждой функции. Текстовые значения параметров функции должны передаваться в кодировке UTF-8.

Пример формирования вызова функции веб-сервиса на языке PHP (кодировка php-файла WINDOWS-1251)

|  |
| --- |
| <?php    // url запроса wsdl-описания  $wsdl = 'http://localhost:8020/eda/KBD API\_docs?wsdl&use=encoded&api=1';  // создание SOAP-клиента на основе wsdl-описания  $sclient = new SoapClient($wsdl,  array(  'encoding' => 'UTF-8',  'user\_agent' => 'PHP/SOAP',  'exceptions' => true,  'login' => 'user',  'password' => 'password'  )  );    Try {  $attr = 7774;  echo 'Описание классификатора атрибута ', $attr, ‘<br/><br/>’;  // вызов функции веб-сервиса  $classifDesc = $sclient->GetClassificatorDesc($attr);  // вывод результатов  echo 'Значения:<br/>’;  $i = 1;  if (property\_exists($classifDesc->values, 'item\_ClassificatorValue')) {  foreach ($classifDesc->values->item\_ClassificatorValue as $value) {  echo $i, '. Внутренний номер: ', $value->oid, ‘<br/>’;  echo 'Название: ', iconv('UTF-8', 'WINDOWS-1251', $value->name);  echo ‘<br/><br/>’;  $i++;  }  } else {  echo 'Отсутствуют <br/>’;  }    } catch (SoapFault $e) {  <br/>  echo iconv('UTF-8', 'WINDOWS-1251', $e->faultstring), ‘<br/>’;  }  ?> |

# ДОКУМЕНТЫ, ФАЙЛЫ (KBD API\_docs)

Данная группа предназначена для работы с карточками документов, текстами и вложениями: для получения списка атрибутов карточки и значений классификатора, создания и редактирования карточки документа, загрузки текстов, вложений и др.

Пример запроса WSDL-описания для группы

|  |
| --- |
| <?php    // url запроса wsdl-описания  $wsdl = 'http://localhost:8020/eda/KBD API\_docs?wsdl&use=encoded&api=1';    // ...    ?> |

## GetCardAttributes

Функция получения списка атрибутов для заполнения карточки документа КБД.

Синтаксис

|  |
| --- |
| function GetCardAttributes() : CardAttribute[] |

Параметры

Отсутствуют.

Возвращаемое значение

Функция возвращает массив объектов [CardAttribute](#_CardAttribute).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // получение набора атрибутов карточки документа  try {  $attrslist = $sclient->GetCardAttributes();  $i = 1;  foreach ($attrslist->item\_CardAttribute as $attr) {  echo $i, '. Имя: ', $attr->title, '<br/>';  echo 'Тип: ', $attr->type, '<br/>';  echo 'Номер: ', $attr->attrnum, '<br/><br/>';  $i++;  };  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetServiceAttributes

Функция получения списка служебных атрибутов для документа КБД.

Синтаксис

|  |
| --- |
| function GetServiceAttributes() : CardAttribute[] |

Параметры

Отсутствуют.

Возвращаемое значение

Функция возвращает массив объектов [CardAttribute](#_CardAttribute).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // получение набора атрибутов карточки документа  try {  $attrslist = $sclient-> GetServiceAttributes();  $i = 1;  foreach ($attrslist->item\_CardAttribute as $attr) {  echo $i, '. Имя: ', $attr->title, '<br/>';  echo 'Тип: ', $attr->type, '<br/>';  echo 'Номер: ', $attr->attrnum, '<br/><br/>';  $i++;  };  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## SetServiceAttributes

Функция формирования запроса на заполнение служебных атрибутов документа КБД (не публиковать, ключевые слова, черновик).

Синтаксис

|  |
| --- |
| function SetServiceAttributes (  oid : int,  guid : string,  attrs : string,  format : string,  ) : string |

Параметры

**oid** — внутренний номер документа (число)

**guid** — guid ранее загруженного через KBD API документа (строка)

Обязательно должно быть передано хотя бы одно из значений — oid (приоритетнее) или guid

**attrs** — обязательный (строка), сериализованный в JSON или XML массив объектов (описаний атрибутов карточки для перезаписи значений) атрибутами:

- **attrnum** — номер атрибута

- **value** — значение атрибута (разделитель значений - ';')

**format** — формат, в котором переданы атрибуты, json или xml (строка), по умолчанию json.

Возвращаемое значение

Функция возвращает массив объектов [CardAttribute](#_CardAttribute).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // получение набора атрибутов карточки документа  try {  $attrslist = $sclient->SetServiceAttributes($oid, null, $attrs);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetClassificatorDesc

Функция получения списка значений классификатора.

Синтаксис

|  |
| --- |
| function GetClassificatorDesc(  attrnum : int,  parent\_oid : int,  ) : ClassificatorDesc |

Параметры

**attrnum** — номер атрибута-классификатора, обязательный;

**parent\_oid** — внутренний номер значения верхнего уровня (для иерархических классификаторов; в случае отсутствия - возврат атрибутов верхнего уровня).

Возвращаемое значение

Функция возвращает объект [ClassificatorDesc](#_ClassificatorDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение списка значения классификатора  try {  $classifDesc = $sclient->GetClassificatorDesc('7761', null);  echo 'Тип: ', $classifDesc->type, '<br/><br/>';  echo 'Значения: ', '<br/>';  $i = 1;  foreach ($classifDesc->values->item\_ClassificatorValue as $cvalue) {  echo $i, '. Внутренний номер: ', $cvalue->oid, '<br/>';  echo 'Название: ', $cvalue->name, '<br/>';  $i++;  };  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetStatusSettings

Функция получения настроек статуса (номер атрибута статуса, актуальность, действие, замена и отмена).

Синтаксис

|  |
| --- |
| function GetStatusSettings() : CardAttribute[] |

Параметры

Отсутствуют.

Возвращаемое значение

Функция возвращает объект [StatusSettings](#_StatusSettings).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение списка значения классификатора  try {  $statusset = $sclient->GetStatusSettings();  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## CreateCard

Функция формирования запроса на создание карточки документа КБД.

Синтаксис

|  |
| --- |
| function CreateCard(  attrs : string (  attrnum : int,  value : string  ),  format : string,  ) : string |

Параметры

**attrs** — обязательный, сериализованный в JSON или XML массив объектов (описаний атрибутов карточки) с атрибутами:

- **attrnum** — номер атрибута;

- **value** — значение атрибута (разделитель значений - ';').

**format** — формат, в котором переданы атрибуты, json или xml, по умолчанию json.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| **docsAttributes.php**  <?php  // Классы атрибутов документа  class CardAttribute {  var $attrnum;  var $value;    function \_\_construct($attrnum, $value) {  $this->attrnum = $attrnum;  $this->value = $value;  }  };  class TextFileDesc {  var $filename;    function \_\_construct($filename) {  $this->filename = $filename;  }  };  class TextFileDescBase64 {  var $filename;  var $base64str;    function \_\_construct($filename, $base64str) {  $this->filename = $filename;  $this->base64str = $base64str;  }  };  class AttachmentFileDesc {  var $filename;  var $attachmentname;    function \_\_construct($filename, $attachmentname) {  $this->filename = $filename;  $this->attachmentname = $attachmentname;  }  };  ?>  **CreateCard.php**  <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // классы атрибутов документа  include ' docsAttributes.php';    // создание карточки документа  $attrs[] = new CardAttribute(1, ‘Новый документ’);  $attrs[] = new CardAttribute(7, '01.09.2019');  $GUID = $sclient->CreateCard(json\_encode($attrs));  ...  ?> |

## CheckRequestStatus

Функция проверки состояния запроса.

Синтаксис

|  |
| --- |
| function CheckRequestStatus(  guid : string  ) : ArrayOfstring[] |

Параметры

**guid** — guid ранее загруженного через KBD API документа, обязательный.

Возвращаемое значение

Функция возвращает массив строк ArrayOfstring.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // проверка состояния запроса  try {  $CHKRequest = $sclient->CheckRequestStatus($GUID);  echo 'Состояние запроса для GUID - ', $GUID, ': ', $CHKRequest->string;  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDocumentDesc

Функция получения описания ранее загруженного через KBD API документа КБД.

Синтаксис

|  |
| --- |
| function GetDocumentDesc(  guid : string,  ) : DocumentDesc |

Параметры

**guid** — guid ранее загруженного документа, обязательный.

Возвращаемое значение

Функция возвращает объект [DocumentDesc](#_DocumentDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  try {  $docDesc = $sclient->GetDocumentDesc($GUID);  echo 'Название документа - ', $docDesc->name, '<br/>';  echo 'OID документа - ', $docDesc->oid, '<br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDocumentItems

Функция получения частей документа КБД (текст, примечания, вложения).

Синтаксис

|  |
| --- |
| function GetDocumentItems(  oid : int,  guid : string,  item\_type : string  ) : DocumentItems |

Параметры

**oid** — внутренний номер документа;

**guid** — guid ранее загруженного через KBD API документа;

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

**item\_type** — тип запрашиваемых частей документа: text, notes, attachments; при отсутствии (null) возвращается всё.

Возвращаемое значение

Функция возвращает объект [DocumentItems](#_DocumentItems).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение части документа КБД  try {  $docItem = $sclient->GetDocumentItems($GUID);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## EditCard

Функция формирования запроса на редактирование карточки документа КБД.

Синтаксис

|  |
| --- |
| function EditCard(  oid : int,  attrs : string,  format : string,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**attrs** — обязательный, сериализованный в JSON или XML массив объектов (описаний атрибутов карточки для перезаписи значений) с атрибутами:

- **attrnum** — номер атрибута;

- **value** — значение атрибута (разделитель значений - ';').

**format** — формат, в котором переданы атрибуты, json или xml, по умолчанию json.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| **docsAttributes.php**  <?php  // Классы атрибутов документа  class CardAttribute {  var $attrnum;  var $value;    function \_\_construct($attrnum, $value) {  $this->attrnum = $attrnum;  $this->value = $value;  }  };  class TextFileDesc {  var $filename;    function \_\_construct($filename) {  $this->filename = $filename;  }  };  class TextFileDescBase64 {  var $filename;  var $base64str;    function \_\_construct($filename, $base64str) {  $this->filename = $filename;  $this->base64str = $base64str;  }  };  class AttachmentFileDesc {  var $filename;  var $attachmentname;    function \_\_construct($filename, $attachmentname) {  $this->filename = $filename;  $this->attachmentname = $attachmentname;  }  };  ?>  **EditCard.php**  <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // классы атрибутов документа  include ' docsAttributes.php';  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  ... (см. Пример [GetDocumentDesc](#_GetDocumentDesc))  // редактирование карточки документа  $attrs [] = new CardAttribute(2, '1234-567');  $attrs [] = new CardAttribute(7, '10.05.2015');  try {  $GUID = $sclient->EditCard($OID, json\_encode($attrs));  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## GetEditionDesc

Функция получения описания ранее созданной через KBD API редакции документа.

Синтаксис

|  |
| --- |
| function GetEditionDesc(  guid : string,  ) : EditionDesc |

Параметры

**guid** — guid ранее загруженного документа, обязательный.

Возвращаемое значение

Возвращает объект [EditionDesc](#_EditionDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // редактирование карточки документа  ... (см. Пример [EditCard](#_EditCard_1))    // получение описания редакции документа  $editDesc = $sclient-> GetEditionDesc($GUID);  ...    ?> |

## SetText

Функция формирования запроса на добавление текста в форматах doc(x), rtf, pdf (атрибут 7769) для существующего документа.

Синтаксис

|  |
| --- |
| function SetText(  oid : int,  guid : string,  format : string,  base64str : string,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**guid** — guid ранее загруженного через KBD API документа.

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

**format** — формат (расширение) файла, обязательный.

**base64str** — содержимое файла в формате base64, обязательный.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  ... (см. Пример [GetDocumentDesc](#_GetDocumentDesc))  // добавление текста документа  $path = 'rtftext.rtf';  $fh = fopen($path, 'r');  $base64str = base64\_encode(fread($fh, filesize($path)));  fclose($fh);  try {  $GUID = $sclient->SetText($OID, null, 'rtf', $base64str);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## GetLinkedDocsAttributes

Функция получения номеров атрибутов, настроенных как атрибуты ссылки на заменяющий/заменяемый и отменяющий/отменяемый документы.

Синтаксис

|  |
| --- |
| function GetLinkedDocsAttributes(  status : string,  ) : LinkedDocsAttributes |

Параметры

**status** — статус необходимых атрибутов ссылки, обязательный, возможные значения: 'substitution'/'cancel'.

Возвращаемое значение

Функция возвращает объект [LinkedDocsAttributes](#_LinkedDocsAttributes).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение набора номера атрибута ссылки  try {  echo 'Замененные: ', '<br/>';  $linkAttr = $sclient->GetLinkedDocsAttributes('substitution');  echo 'Номер атрибута ссылки на заменяющий/отменяющий документ: ', $linkAttr->activeAttrNum, '<br/>';  echo 'Номер атрибута ссылки на заменяемый/отменяемый документ: ', $linkAttr->passiveAttrNum, '<br/>';  echo 'Отмененные: ', '<br/>';  $linkAttr = $sclient->GetLinkedDocsAttributes('cancel');  echo 'Номер атрибута ссылки на заменяющий/отменяющий документ: ', $linkAttr->activeAttrNum, '<br/>';  echo 'Номер атрибута ссылки на заменяемый/отменяемый документ: ', $linkAttr->passiveAttrNum, '<br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## AddAttachment

Функция формирования запроса на добавление вложенного файла для документа КБД.

Синтаксис

|  |
| --- |
| function AddAttachment(  oid : int,  guid : string,  filename : string,  base64str : string,  isOriginal : int,  processPdfLinks : int,  needConversion : int,  denyPdfPrint : int,  denyDownload : int  ) : string |

Параметры

**oid** — внутренний номер документа;

**guid** — guid ранее загруженного через KBD API документа;

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

**filename** — отображаемое после загрузки имя файла вместе с расширением, обязательный;

**base64str** — текст в формате base64, обязательный;

**isOriginal** — является ли оригиналом, по умолчанию - 0 (не оригинал), любое другое число, кроме 0 – оригинал;

**processPdfLinks** — обрабатывать ли ссылки в pdf-вложениях, по умолчанию - 0 (не обрабатывать), любое другое число – обрабатывать;

**needConversion** — требуется ли конвертация для обеспечения встроенного просмотра, печати и поиска, по умолчанию - 0 (не требуется), любое другое число – требуется;

**denyPdfPrint** — устанавливать ли защиту от печати для pdf-вложений, по умолчанию - 0 (не устанавливать), любое другое число – устанавливать;

**denyDownload** — запрещать ли скачивание, по умолчанию - 0 (разрешать), любое другое число – запрещать.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // добавление вложенного файла  try {  $path = 'attachment.xls';  $fh = fopen($path, 'r');  $base64str = base64\_encode(fread($fh, filesize($path)));  fclose($fh);  $GUID = $sclient->AddAttachment(null, $GUID, $path, $base64str, 0,0,0,0,0);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDocumentAttachmentsDesc

Функция получения описаний вложений документа.

Синтаксис

|  |
| --- |
| function GetDocumentAttachmentsDesc(  oid : int,  guid : string,  filename : string  ) : AttachmentDesc[] |

Параметры

**oid** — внутренний номер документа;

**guid** — guid ранее загруженного через KBD API документа;

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

**filename** — имя вложения вместе с расширением (строка); если не передано, будут возвращены описания всех вложений.

Возвращаемое значение

Возвращает массив объектов [AttachmentDesc](#_AttachmentDesc) с атрибутами.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение информации о вложенных файлах  try {  $docAttach = $sclient->GetDocumentAttachmentsDesc($GUID);  echo 'Название вложения - ', $docAttach->filename, '<br/>';  echo 'Описание вложения - ', $docAttach->description, '<br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## EditAttachments

Функция формирования запроса на редактирование вложенных файлов для документа КБД.

Синтаксис

|  |
| --- |
| function EditAttachments (  attachOid : int,  description : string,  isScancopy : int,  needConversion : int,  denyDownload : int,  needEncryption : int,  needIndex : int  ) : string |

Параметры

Сериализованный в JSON объект (или массив объектов) с атрибутами:

**attachOid** — внутренний номер вложения, обязательный;

Далее отсутствие атрибута или null-значение означает «оставить без изменений».

**description** — описание вложения, пустая строка ('') - удалить описание;

**isScancopy** — признак скан-копии, 0 - снять, 1 или любое другое число - установить;

**needConversion** — конвертация, 0 - отменить конвертацию, 1 или любое другое число - конвертировать;

**denyDownload** — запрет скачивания, 0 - снять, 1 или любое другое число - установить;

**needEncryption** — шифрование, 0 - дешифровать, 1 или любое другое число - зашифровать;

**needIndex** — индексация, 0 - оставить без изменений, 1 или любое другое число – индексировать.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // объявление классов  class Attachment {  public $attachOid;  public $description;  public $isScancopy;  public $needConversion;  public $denyDownload;  public $needEncryption;  public $needIndex;    function \_\_construct($attachOid, $description, $isScancopy, $needConversion, $denyDownload, $needEncryption, $needIndex) {  $this->attachOid = $attachOid;  $this->description = $description;  $this->isScancopy = $isScancopy;  $this->needConversion = $needConversion;  $this->denyDownload = $denyDownload;  $this->needEncryption = $needEncryption;  $this->needIndex = $needIndex;  }  }  // изменение вложения  try {  $attachmentsParams = new Attachment(816800017, 'Новое описание вложенния', 1, 1, 1, 1, 1);  echo json\_encode($attachmentsParams), '<br/>';    $guid = $sclient->EditAttachments('{"attachOid":816800017,"description":"","isScancopy":0,"needConversion":0,"denyDownload":0,"needEncryption":0,"needIndex":0}');  echo 'Guid=', $guid, '<br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## DeleteAttachments

Функция формирования запроса на удаление вложенных файлов для документа КБД.

Синтаксис

|  |
| --- |
| function DeleteAttachments(  oid : int,  filenames : string,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**filenames** — сериализованный в JSON массив имён вложений для удаления - вместе с расширениями, обязательный.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  ... (см. Пример [GetDocumentDesc](#_GetDocumentDesc))  // удаление вложения документа  try {  $filenames[] = 'attachment.docx';  $GUID = $sclient->DeleteAttachments($OID, json\_encode($filenames));  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## CreateEdition

Функция формирования запроса на создание редакции документа КБД.

Синтаксис

|  |
| --- |
| function CreateEdition(  oid : int,  date\_begin : string,  date\_add : string,  date\_create : string,  isFuture : int,  changingDoc : int,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**date\_begin** — дата начала действия (атрибут 1134), необязательный.

**date\_add** — дата внесения в БД (атрибут 7878), по умолчанию - текущая дата.

**date\_create** — дата редакции (атрибут 12), по умолчанию - текущая дата.

**isFuture** — флаг для будущей редакции, по умолчанию - 0, любое другое число, кроме 0 - будущая редакция.

**changingDoc** — внутренний номер изменяющего документа, необязательный.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  ... (см. Пример [GetDocumentDesc](#_GetDocumentDesc))    // создание редакции документа  try {  $GUID = $sclient->CreateEdition($OID, '01.09.2020', '01.08.2019', '01.07.2019', '1');  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## RemoveDocument

Функция формирования запроса на окончательное удаление документа.

Синтаксис

|  |
| --- |
| function RemoveDocument(  oid : int,  guid : string,  ) : string |

Параметры

**oid** — внутренний номер документа.

**guid** — guid ранее загруженного через KBD API документа.

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  try {  $GUID = $sclient-> RemoveDocument($OID);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## SetDraftModeOn / SetDraftModeOff

Функции формирования запроса на включение/выключение флага "Черновик КБД" для документа КБД.

Синтаксис

|  |
| --- |
| function SetDraftModeOn(  oid : int,  guid : string,  ) : string |

|  |
| --- |
| function SetDraftModeOff(  oid : int,  guid : string,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**guid** — guid ранее загруженного через KBD API документа.

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // включение флага «Черновик КБД»  try {  $GUID\_DM = $sclient->SetDraftModeOn(null, $GUID);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## SetKodeksAttachModeOn / SetKodeksAttachModeOff

Функции формирования запроса на включение/выключение флага "Хранить вложения тиражно" для документа КБД.

Синтаксис

|  |
| --- |
| function SetKodeksAttachOn(  oid : int,  guid : string,  ) : string |

|  |
| --- |
| function SetKodeksAttachOff(  oid : int,  guid : string,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**guid** — guid ранее загруженного через KBD API документа.

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  ... (см. Пример [GetDocumentDesc](#_GetDocumentDesc))  // включение флага « Хранить вложения тиражно»  try {  $GUID\_AM = $sclient->SetKodeksAttachModeOn($OID, null);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

## SetKodeksStatus

Функция формирования запроса на установку тиражного статуса документа КБД (атрибут 14).

Синтаксис

|  |
| --- |
| function SetKodeksStatus(  oid : int,  guid : string,  status : string,  ) : string |

Параметры

**oid** — внутренний номер документа, обязательный.

**guid** — guid ранее загруженного через KBD API документа.

Обязательно должно быть передано хотя бы одно из значений: oid (приоритетнее) или guid.

**status** — статус документа; возможные значения: 'active'/'inactive'; если не передано - статус сбрасывается.

Возвращаемое значение

Функция возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // создание карточки документа  ... (см. Пример [CreateCard](#_CreateCard))  // получение описания документа  ... (см. Пример [GetDocumentDesc](#_GetDocumentDesc))  // установка тиражного статуса документа КБД  try {  $setKStatus = $sclient->SetKodeksStatus($OID, null, 'active');  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

# ТЕРМИНЫ (KBD API\_terms)

Данная группа предназначена для работы с пользовательским словарём: для поиска, импорта, создания, редактирования и удаления терминов и определений.

Пример запроса WSDL-описания для группы

|  |
| --- |
| <?php    // url запроса wsdl-описания  $wsdl = 'http://localhost:8020/eda/KBD API\_term?wsdl&use=encoded&api=1';    // ...    ?> |

## CreateTerm

Функция создания запроса на добавление нового термина и его определения.

Синтаксис

|  |
| --- |
| function CreateTerm(  termname : string,  value : string,  links : string,  section : int,  ) : string |

Параметры

**termname** — имя термина и его первого определения, обязательный.

**value** — значение термина, обязательный.

**links** — внутренние номера устанавливающих документов (разделитель значений - ';').

**section** — номер дополнительного (не КБД) раздела.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  $tName = 'Новый термин';  $tValue = 'Описание нового термина.';  $tLink = '1200015761; 1200141448';  try {  $GUID = $sclient->CreateTerm($tName, $tValue, $tLink);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## CheckRequestStatus

Функция проверки состояния запроса.

Синтаксис

|  |
| --- |
| function CheckRequestStatus(  guid : string,  ) : ArrayOfstring[] |

Параметры

**guid** — guid ранее загруженного термина, обязательный.

Возвращаемое значение

Функция возвращает массив строк ArrayOfstring.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))    // проверка состояния запроса  try {  $requestStatus = $sclient->CheckRequestStatus($GUID);  echo $requestStatus->string;  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDefDescByGuid

Функция получения внутреннего номера и названия ранее загруженного через KBD API определения термина.

Синтаксис

|  |
| --- |
| function GetDefDescByGuid(  guid : string,  ) : DefinitionDesc |

Параметры

**guid** — guid ранее загруженного через KBD API термина, обязательный.

Возвращаемое значение

Функция возвращает объект [DefinitionDesc](#_DefinitionDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))    // получение внутреннего номера и названия определения  try {  $defdesc = $sclient->GetDefDescByGuid($GUID);  echo 'Имя: ', $info->defname, '<br/>';  echo 'DEFOID: ', $info->defoid;  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDefDescByLink

Функция получения описаний определений терминов по заданному oid устанавливающего документа.

Синтаксис

|  |
| --- |
| function GetDefDescByLink(  linkoid : int,  ) : FullDefinitionDesc[] |

Параметры

**linkoid** — внутренний номер устанавливающего документа, обязательный.

Возвращаемое значение

Функция возвращает массив объектов [FullDefinitionDesc](#_FullDefinitionDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение описаний определений терминов по oid  try {  $defdesc = $sclient-> GetDefDescByLink($LinkOID);  echo 'Внутренний номер: ', $defdesc->defoid, '<br/>';  echo 'Название: ', $defdesc->defname, '<br/><br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDefDescByName

Функция получения описаний определений терминов по имени.

Синтаксис

|  |
| --- |
| function GetDefDescByName(  defname : string,  ) : FullDefinitionDesc[] |

Параметры

**defname** — имя определения термина, обязательный.

Возвращаемое значение

Функция возвращает массив объектов [FullDefinitionDesc](#_FullDefinitionDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  try {  $defdesc = $sclient-> GetDefDescByLink($Name);  echo 'Внутренний номер: ', $defdesc->defoid, '<br/>';  echo 'Название: ', $defdesc->defname, '<br/><br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetTermDescByGuid

Функция получения внутреннего номера и названия ранее загруженного через KBD API термина.

Синтаксис

|  |
| --- |
| function GetTermDescByGuid (  guid : string,  ) : TermDesc |

Параметры

**guid** — guid ранее загруженного термина, обязательный.

Возвращаемое значение

Функция возвращает массив объектов [FullDefinitionDesc](#_FullDefinitionDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  try {  $termdesc = $sclient->GetTermDescByGuid($GUID);  echo 'Внутренний номер: ', $defdesc->defoid, '<br/>';  echo 'Название: ', $defdesc->defname, '<br/><br/>';  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetTermDescByName

Функция получения внутреннего номера и названия ранее загруженного через KBD API термина.

Синтаксис

|  |
| --- |
| function GetTermDescByName(  termname : string,  ) : ArrayOfint |

Параметры

**termname** — имя термина, обязательный.

Возвращаемое значение

Функция возвращает массив внутренних номеров найденных документов – терминов (числа).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  try {  $termdesc = $sclient->GetTermDescByName($Name);  echo 'TERMOID: ', $guid->int;  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## GetDefinitionDuplicates

Функция проверки определения термина на существование дубликатов в уже имеющихся терминах.

Синтаксис

|  |
| --- |
| function GetDefinitionDuplicates(  defname : string,  links : string,  ) : DuplicateTermDesc[] |

Параметры

**defname** — имя определения термина, обязательный.

**links** — внутренние номера устанавливающих документов (разделитель значений - ';').

Возвращаемое значение

Функция массив объектов [DuplicateTermDesc](#_DuplicateTermDesc).

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // проверка определения термина на существование дубликатов  try {  $defdup = $sclient->GetDefinitionDuplicates($Name, 1200083232);  echo 'Внутренний номер: ', $value->defoid, '<br/>';  echo 'Название: ', $value->termname, '<br/><br/>'; } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## AddTermDefinition

Функция создания запроса на добавление нового определения для существующего термина.

Синтаксис

|  |
| --- |
| function AddTermDefinition(  termoid : int,  defname : string,  value : string,  links : string,  order : int,  section : int,  ) : string |

Параметры

**termoid** — внутренний номер термина, к которому будет привязано определение, обязательный.

**defname** — имя определения термина, обязательный.

**value** — определение термина, обязательный.

**links** — внутренние номера устанавливающих документов (разделитель значений - ';').

**order** — порядковый номер определения термина, по умолчанию определение добавляется в конец списка).

**section** — номер дополнительного (не КБД) раздела.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  ... (см. Пример [GetTermDescByGuid](#_GetTermDescByGuid))  // добавление нового определения для существующего термина  try {  $guid = $sclient->AddTermDefinition($termoid, $Name, ‘Определение термина’, ‘1200101115; 1200142874’, 3, null);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## AssignTerms

Функция создания запроса на объединение дублирующихся терминов (КБД и “Кодекс” / “Техэксперт”).

Синтаксис

|  |
| --- |
| function AssignTerms() : string |

Параметры

Без параметров.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // получение внутреннего номера и названия определения  try {  $guid = $sclient->AssignTerms();  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## DeleteTerm

Функция создания запроса на удаление термина (со всеми определениями).

Синтаксис

|  |
| --- |
| function DeleteTerm(  termoid : int,  ) : string |

Параметры

**termoid** — внутренний номер удаляемого термина, обязательный.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  ... (см. Пример [GetTermDescByGuid](#_GetTermDescByGuid))  // удаление термина  try {  $guid = $sclient->DeleteTerm($termoid);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## DeleteTermDefinition

Функция создания запроса на удаление определения термина.

Синтаксис

|  |
| --- |
| function DeleteTermDefinition (  defoid : int,  ) : string |

Параметры

**defoid** — внутренний номер определения термина, обязательный.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  ... (см. Пример [GetDefDescByGuid](#_GetDefDescByGuid))  // удаление определения термина  try {  $guid = $sclient->DeleteTermDefinition ($defoid);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## EditTermDefinition

Функция создания запроса на редактирование существующего определения термина.

Синтаксис

|  |
| --- |
| function EditTermDefinition(  defoid : int,  defname : string,  value : string,  links : string,  order : int,  ) : string |

Параметры

**defoid** — внутренний номер определения термина, обязательный.

**defname** — имя определения термина.

**value** — определение термина.

**links** — внутренние номера устанавливающих документов (разделитель значений - ';').

**order** — порядковый номер определения термина.

Обязательно должно быть передано значение как минимум одного из следующих параметров: defname, value, links, order.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  ... (см. Пример [GetDefDescByGuid](#_GetDefDescByGuid))  // редактирование существующего определения  try {  $guid = $sclient-> EditTermDefinition($defoid, null, ‘Определение термина-2’, ‘1200084783’, 7);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## EditTermName

Функция создания запроса на редактирование имени термина.

Синтаксис

|  |
| --- |
| function EditTermName(  termoid : int,  newname : string,  ) : string |

Параметры

**termoid** — внутренний номер документа - термина, обязательный.

**newname** — новое имя термина, обязательный.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  ... (см. Пример [GetTermDescByGuid](#_GetTermDescByGuid))  // редактирование имени термина  try {  $guid = $sclient->EditTermName ($termoid, ‘Наименование термина’);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

## RebindTermDefinition

Функция создания запроса на перепривязку существующего определения термина.

Синтаксис

|  |
| --- |
| function RebindTermDefinition(  defoid : int,  termoid : int,  ) : string |

Параметры

**defoid** — внутренний номер определения термина, обязательный.

**termoid** — внутренний номер документа - термина, обязательный.

Возвращаемое значение

Возвращает guid запроса.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))    // Создание нового термина  ... (см. Пример [CreateTerm](#_CreateTerm))  // получение внутреннего номера и названия определения  ... (см. Пример [GetTermDescByGuid](#_GetTermDescByGuid))  // получение внутреннего номера и названия определения  ... (см. Пример [GetDefDescByGuid](#_GetDefDescByGuid))  // редактирование существующего определения  try {  $guid = $sclient-> RebindTermDefinition($defoid; $termoid);  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...    ?> |

# ФУНКЦИИ ОБЩЕГО НАЗНАЧЕНИЯ (common)

Прочие функции, например, не относящиеся к работе с терминами или документами.

Пример запроса WSDL-описания для группы

|  |
| --- |
| <?php    // url запроса wsdl-описания  $wsdl = 'http://localhost:8020/eda/KBD API\_common?wsdl&use=encoded&api=1';    // ...    ?> |

## SynchroniseVolumes

Функция для запуска синхронизации томов КБД.

Синтаксис

|  |
| --- |
| function SynchroniseVolumes() : string |

Параметры

Отсутствуют.

Возвращаемое значение

Возвращает GUID.

Пример

|  |
| --- |
| <?php    // создание SOAP-клиента на основе wsdl-описания  ... (см. Пример [параметров запросов](#_Параметры_запросов))  // синхронизация томов КБД  try {  $setKStatus = $sclient->SynchroniseVolumes();  } catch (SoapFault $e) {  echo 'ERROR!', '<br/>';  echo $e->faultstring, '<br/>';  }  ...  ?> |

# СТРУКТУРЫ

## CardAttribute

Структура описывает атрибут карточки документа БД «Банк документов».

Синтаксис

|  |
| --- |
| struct CardAttribute {  title : string,  type : int,  attrnum : int  } |

Поля

**title : string** — имя атрибута КБД;

**type : int** — тип атрибута;

**attrnum : int** — номер атрибута.

## ClassificatorDesc

Структура описывает классификаторы карточки документа БД «Банк документов».

Синтаксис

|  |
| --- |
| struct ClassificatorDesc {  type : string,  ClassificatorValue[],  } |

Поля

**type** — тип классификатора (иерархический или линейный).

**values** — массив объектов [ClassificatorValue](#_ClassificatorValue) с атрибутами.

## ClassificatorValue

Структура возвращает атрибуты классификатора карточки документа БД «Банк документов».

Синтаксис

|  |
| --- |
| struct ClassificatorValue {  oid : int,  name : string  } |

Поля

**oid** — внутренний номер документа;

**name** — значение классификатора.

## StatusSettings

Возвращает массив объектов с атрибутами вложений.

Синтаксис

|  |
| --- |
| struct StatusSettings {  statusAttr : int,  actual : ClassificatorValue[],  inactive : ClassificatorValue[],  special : ClassificatorValue[],  substitutive : ClassificatorValue[],  substitutable : ClassificatorValue[],  cancelling : ClassificatorValue[],  cancellable : ClassificatorValue[]  } |

Поля

**statusAttr** — номер атрибута-классификатора, использующегося в качестве атрибут статуса (число)

**actual** — описание статусов актуального документа в виде массива объектов [ClassificatorValue](#_ClassificatorValue);

**inactive** — описание статусов недействующего документа в виде массива объектов [ClassificatorValue](#_ClassificatorValue);

**special** — описание статусов документа с особым применением в виде массива объектов [ClassificatorValue](#_ClassificatorValue);

**substitutive** — описание дополнительных статусов заменяющего документа в виде массива объектов [ClassificatorValue](#_ClassificatorValue);

**substitutable** — описание новых статусов заменяемого документа в виде массива объектов [ClassificatorValue](#_ClassificatorValue);

**cancelling** — описание дополнительных статусов отменяющего документа в виде массива объектов [ClassificatorValue](#_ClassificatorValue);

**cancellable** — описание новых статусов отменяемого документа в виде массива объектов [ClassificatorValue](#_ClassificatorValue).

## DocumentDesc

Структура возвращает описание документа БД «Банк документов».

Синтаксис

|  |
| --- |
| struct DocumentDesc {  oid : int,  name : string  } |

Поля

**oid** — внутренний номер документа;

**name** — значение классификатора.

## DocumentItems

Структура возвращает описание частей документа БД «Банк документов».

Синтаксис

|  |
| --- |
| struct DocumentItems {  text {  type : string,  base64str : string  },  notes {  type : string,  base64str : string  },  attachments[]  } |

Поля

**text** — объект с атрибутами:

- **type** – тип файла: pdf, docx, rtf;

- **base64str** – текст в формате base64.

**notes** — объект с атрибутами:

- **type** – тип файла: pdf, docx, rtf;

- **base64str** – текст в формате base64.

**attachments** — массив объектов [AttachmentItem](#_AttachmentItem).

## AttachmentItem

Возвращает массив объектов с атрибутами вложений.

Синтаксис

|  |
| --- |
| struct AttachmentItem {  filename : string,  base64str : string,  isScanCopy : int,  } |

Поля

**filename** — имя вложения вместе с расширением;

**base64str** – текст в формате base64;

**isScanCopy** — является ли вложение скан-копией, 0 - не является, 1 - является

## AttachmentDesc

Возвращает массив объектов с атрибутами вложений.

Синтаксис

|  |
| --- |
| struct AttachmentDesc {  oid : int,  filename : string,  description : string,  modified : string,  isScanCopy : int,  isDownloadable : int,  isConvertable : int,  isConverted : int,  isEncrypted : int,  hasIndex : int  } |

Поля

**oid** — внутренний номер вложения;

**filename** — имя вложения вместе с расширением;

**description** — описание вложения;

**modified** — дата и время изменения файла вложения;

**isScanCopy** — является ли вложение скан-копией, 0 - не является, 1 - является

**isDownloadable** — разрешено ли скачивание, 0 - не разрешено, 1 - разрешено;

**isConvertable** — конвертируемо ли вложение данного типа, 0 - конвертируемо, 1 - не конвертируемо;

**isConverted** — произведена ли конвертация, 0 - не произведена, 1 - произведена;

**isEncrypted** — зашировано ли вложение, 0 - не зашифровано, 1 - зашифровано;

**hasIndex** — произведена ли индексация, 0 - не произведена, 1 - произведена.

## EditionDesc

Структура возвращает описание редакции документа БД «Банк документов».

Синтаксис

|  |
| --- |
| struct EditionDesc {  oid : int,  } |

Поля

**oid** — внутренний номер документа.

## LinkedDocsAttributes

Структура возвращает номера атрибутов, настроенных как атрибуты ссылки на заменяющий/заменяемый и отменяющий/отменяемый документы БД «Банк документов».

Синтаксис

|  |
| --- |
| struct LinkedDocsAttributes {  activeAttrNum : int,  passiveAttrNum : int,  } |

Поля

**activeAttrNum** — номер атрибута ссылки на заменяющий/отменяющий документ;

**passiveAttrNum** — номер атрибута ссылки на заменяемый/отменяемый документ.

Если какой-либо атрибут ссылки не задан, вместо его номера будет возвращён 0.

## DefinitionDesc

Структура возвращает описание определения термина БД «Банк документов».

Синтаксис

|  |
| --- |
| struct DefinitionDesc {  defoid : int,  defname : string  } |

Поля

**defoid** — внутренний номер документа - определения термина;

**defname** — имя определения термина.

## FullDefinitionDesc

Структура возвращает полное описание определения термина БД «Банк документов».

Синтаксис

|  |
| --- |
| struct FullDefinitionDesc {  defoid : int,  defname : string,  value : string,  links : string,  termoid : int,  termname : string  } |

Поля

**defoid** — внутренний номер документа - определения термина;

**defname** — имя определения термина;

**value** – определение термина;

**links** — внутренние номера устанавливающих документов (разделитель значений - ';');

**termoid** — внутренний номер документа термина, к которому привязано определение;

**termname** — имя термина, к которому привязано определение.

## DuplicateTermDesc

Структура возвращает описание дубликатов определения в имеющихся терминах БД «Банк документов».

Синтаксис

|  |
| --- |
| struct DuplicateTermDesc {  defoid : int,  termname : string  } |

Поля

**defoid** — внутренний номер документа – определения термина.

**termname** — имя термина, у которого уже есть такое определение: с тем же именем и списком устанавливающих документов.