KODWEB API

Поисковая Интеграция

ЯНВАРЬ 2020

ОГЛАВЛЕНИЕ

ОБЩАЯ ИНФОРМАЦИЯ

Kodweb API – набор функций для доступа к функциональности ПСС «Кодекс» и «Техэксперт» без привязки к внутренней инфраструктуре клиентской части системы (скрипты браузера, стили и т.д.).

Для взаимодействия с сервером используется протокол SOAP (simple object access protocol). Иными словами, можно сказать, что Kodweb API это не что иное, как веб-сервис.

Клиентская часть может быть разработана на различных языках программирования (например, PHP, C++, C#, Node.js, 1C). Kodweb поддерживает генерацию WSDL для автоматизации процесса построения "клиентов". URL для получения WSDL-описания Kodweb API имеет следующий вид:

http://<имя сервера>[:<порт>]/<виртуальный каталог>/api?wsdl

Некоторые автоматические построители "клиентов" могут использовать RPC/encoded стиль WSDL, например, PHP. В этом случае URL для получения WSDL-описания Kodweb API имеет следующий вид:

http://<имя сервера>[:<порт>]/<виртуальный каталог>/api?wsdl&use=encoded

Некоторые автоматические построители "клиентов" могут генерировать код взаимодействия с сервером, требующим авторизации. Например, PHP может сгенерировать "клиента" для доступа к виртуальному каталогу с авторизацией по базе пользователей.

```php
<?php

// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl&use=encoded&use=encoded";

// параметры базовой HTTP авторизации
$options = array(
    'login' => 'kodeks',
    'password' => 'kodeks'
);

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient($wsdl, $options);

// теперь можно вызывать методы вебсервиса через одноименные методы
// объекта 'kodweb' (экземпляр 'SoapClient')
$searchAttrs = $kodweb->GetSearchAttributes();

// ...

?>
```

С особенностями организации поиска, "специальным" поиском и структурой документов в ПСС "Кодекс" можно ознакомиться в соответствующих подразделах:

- Поиск;
- Специальный поиск и специальный фильтр;
- Структура документов.

ПОИСК

АТРИБУТНЫЙ ПОИСК

В базах данных «Кодекс» и «Техэксперт» документы находятся в различных областях (таблицах). Однородные данные в документах из разных областей могут находиться в различных атрибутах (столбцах таблицы). Например, номер документа в одной области находится в атрибуте с индексом 2, а в другой - в атрибуте с индексом 5. Для того, чтобы можно было проводить поиск документов во всех областях, как по единой базе данных, в ПСС «Кодекс» и «Техэксперт» введено понятие «поисковый атрибут».

У каждого поискового атрибута есть имя, которое отражает его смысл. Например: 'Номер документа', 'Дата принятия' и т.д. Также поисковые атрибуты разделены на несколько типов:

- строка;
- номер;
- дата;
- классификатор.

Набор поисковых атрибутов в системе зависит от состава БД (подключенных томов). Для получения списка доступных поисковых атрибутов в Kodweb API предназначена функция GetSearchAttributes. Для представления информации о поисковом атрибуте используется структура SearchAttribute.

От типа поискового атрибута зависит, каким образом по нему будет производиться поиск в БД. Для разных типов поисковых атрибутов условия поиска определяются по-разному в соответствующих структурах:

- SearchConditionString;
- SearchConditionNumber;
- SearchConditionDate;
- SearchConditionClassificator.

Что бы сформировать условие поиска для поискового атрибута типа 'классификатор' необходимо знать идентификатор нужного классификатора. Для этого необходимо предварительно получить список классификаторов, относящихся к данному поисковому атрибуту.

ИНТЕЛЛЕКТУАЛЬНЫЙ ПОИСК

Интеллектуальный поиск осуществляется вызовом функции FuzzySearch, единственным параметром которой, является строка с запросом.

СПЕЦИАЛЬНЫЙ ПОИСК И СПЕЦИАЛЬНЫЙ ФИЛЬТР

В ПСС «Кодекс» и «Техэксперт» есть некоторые наборы документов, которые выделены из общего массива документов. Эти документы, как правило, не попадают в результат обычного поиска. Для каждого такого набора внутри системы создаётся специальная именованная поисковая структура. Для поиска в таком наборе документов, а также для наложения фильтра на такие документы, системе необходимо указать имя соответствующей поисковой структуры.

С каждой специальной поисковой структурой связан свой набор поисковых атрибутов. Поисковые атрибуты не содержат в себе информации о том, для какой поисковой структуры они были построены. По этой причине не следует идентификаторы поисковых атрибутов одной поисковой структуры для поиска с использованием другой или для обычного поиска. Результат будет

непредсказуемый. Контроль выполнения этого требования ложится на разработчика клиентской части.

СТРУКТУРА ДОКУМЕНТОВ

Документы в ПСС «Кодекс» и «Техэксперт» помимо текста могут содержать и другую информацию различного рода. По этой причине документы разделены на вкладки, каждая из которых содержит соответствующую ей информацию. Типичными примерами могут быть вкладки "Текст" (содержит текст документа) и "Статус" (содержит дополнительную информацию о самом документе).

ФУНКЦИИ

ATTRSEARCH

Функция **AttrSearch** производит атрибутный поиск по базе документов в соответствии с заданными условиями. Если заданы условия поиска для нескольких поисковых атрибутов, то все они объединяются по 'и'.

Синтаксис

```
function AttrSearch(
    conditions : any[],
    searchName : string,
    search_tabs_index : int,
) : DocListInfo
```

Параметры

conditions — массив условий поиска. Условие поиска – это структура одного из следующих типов: SearchConditionString, SearchConditionNumber, SearchConditionDate и SearchConditionClassificator.

searchName — имя поисковой структуры в случае специального поиска. В случае обычного поиска значение этого параметра должно быть null.

search_tabs_index — индекс поисковой вкладки (района). Значение параметра можно взять из поля id структуры SearchArea. Если параметр не задан, то в результат попадут все найденные документы, что равносильно заданию индекса равному -1. "Все" -1

Возвращаемое значение

Функция возвращает структуру DocListInfo, содержащую информацию о списке найденных документов.

Пример

Предположим, для некоторого состава базы данных в наборе поисковых атрибутов есть следующие поисковые атрибуты:

- Наименование (id: 0; тип: строка)
- Вид документа/материала (id: 1; тип: классификатор)
- Номер (id: 3; тип: номер)
- Дата принятия (id: 4; тип: дата)

Предположим также, что необходимо найти все 'Законы РФ', наименование которых содержит слова 'О праве', номер начинается на '48', принятые в период с 01.12.1992 по 31.05.1993 гг. Следующий пример демонстрирует как можно сформировать условия для поиска, и произвести поиск по всем, вышеперечисленным, поисковым атрибутам:

searchConditions.php

```
<?php
// классы условий поиска
```

```
class SearchConditionString {
   var $id;
   var $value;

   function __construct($id, $value) {
      $this->id = $id;
      $this->value = $value;
   }
};

class SearchConditionNumber {
   var $id;
   var $value;
   var $mode;

   function __construct($id, $value, $mode) {
      $this->id = $id;
      $this->value = $value;
      $this->mode = $mode;
   }
};

class SearchConditionDate {
   var $id;
   var $values;
   var $mode;

   function __construct($id, $values, $mode) {
      $this->id = $id;
      $this->values = $values;
      $this->mode = $mode;
   }
};

class SearchConditionClassificator {
   var $id;
   var $values;
   var $lop;

   function __construct($id, $values, $lop) {
      $this->id = $id;
      $this->values = $values;
      $this->lop = $lop;
   }
};
?>
```

AttrSearch.php

```
<?php
```

```
// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl&use=encoded";

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient($wsdl);

// получение набора поисковых атрибутов
... (см. Пример GetSearchAttributes)

// получение списка классификаторов для
// поискового атрибута "Вид документа/материала"
... (см. Пример GetClassificators)

// классы условий поиска
include 'searchConditions.php';

// формирование массива условий поиска:

// для поискового атрибута 'Наименование'
$conditions[] = new SearchConditionString(
   0,                    // id поискового атрибута
   "О праве"             // строка для поиска
   // Если кодировка по умолчанию не UTF-8 (к примеру, windows-1251), то требуется
преобразование.
   // Можно воспользоваться функцией iconv:
   // iconv("WINDOWS-1251", "UTF-8", "О праве")
);

// для поискового атрибута 'Вид документа/материала'
$conditions[] = new SearchConditionClassificator(
   1,                          // id поискового атрибута
   array(902118100),           // id классификатора 'Закон РФ'
   "or"                        // логическая операция
);

// для поискового атрибута 'Номер'
$conditions[] = new SearchConditionNumber(
   3,                  // id поискового атрибута
   "48",               // строка для поиска
   4                   // режим поиска 'Начинается с'
);

// для поискового атрибута 'Дата принятия'
$conditions[] = new SearchConditionDate(
   4,                      // id поискового атрибута
   array(                  // для выбранного режима поиска нужны два значения
      "01.12.1992",        // дата начала периода
      "31.05.1993"         // дата окончания периода
   ),
   3                       // режим поиска 'Период'
);
```

```
// вызов функции атрибутного поиска.
// в $listInfo вернётся информация о списке найденных документов.
$listInfo = $kodweb->AttrSearch($conditions, NULL);

...

?>
```

FUZZYSEARCH

Функция **FuzzySearch** производит интеллектуальный поиск по базе документов в соответствии с заданными условиями.

Синтаксис

```
function FuzzySearch(
    text : string,
    searchName : string,
    search_tabs_index : int,
    searchVariants : string,
    bparser : string
) : DocListInfo
```

Параметры

text — строка запроса. Слово или число взятое в кавычки (") не будет рассматриваться как атрибут документа. Символ N означает, что следующее слово нужно рассматривать как номер документа.

searchName — имя поисковой структуры в случае специального поиска. В случае обычного поиска значение этого параметра должно быть null.

search_tabs_index — индекс поисковой вкладки (района). Значение параметра можно взять из поля id структуры SearchArea. Если параметр не задан, то в результат попадут все найденные документы, что равносильно заданию индекса равному -1.

searchVariants — строка с вариантными параметрами поиска, разделенными символом &. Имеет следующие параметры:

- **'searchbynames'** - поиск по наименованиям и оглавлениям документов, требуется задание параметра bparser,
- **'archs'** - искать среди прочих.
- **NULL или пустая строка** - в случае обычного поиска.

bparser — результат разбора bparser-ом строки запроса. Параметр требуется для поиска по наименованиям. Нужное значение можно получить с помощью HTTP запроса с параметром */bparser?parse=text*

Возвращаемое значение

Функция возвращает структуру DocListInfo, содержащую информацию о списке найденных документов.

Пример

Поиск по наименованиям и оглавлениям документов:

FuzzySearch.php

```
<?php

// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl&use=encoded";

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient($wsdl);

// получение результата разбора строки запроса bparser-ом
$text = "аренда земельного участка";
$bparser = file_get_contents("http://localhost/kodeks/bparser?parse=" . urlencode($text));

// вызов функции интеллектуального поиска.
// в $listInfo вернётся информация о списке найденных документов.
$listInfo = $kodweb->FuzzySearch($text, NULL, NULL, "searchbynames", $bparser);

...

?>
```

FILTERLIST

Функция **FilterList** производит наложение фильтра на заданный список. Если заданы условия фильтра для нескольких поисковых атрибутов, то все они объединяются по 'и'.

Синтаксис

```
function FilterList(
   conditions : any[],
   list : string,
   search_name : string
) : DocListInfo
```

Параметры

conditions — массив условий фильтра. Условие фильтра – это структура одного из следующих типов: SearchConditionString, SearchConditionNumber, SearchConditionDate, SearchConditionClassificator

list — список внутренних номеров документа, передается текстовой строкой с nd через запятую.

searchName — имя поисковой структуры в случае специального фильтра. В случае обычного фильтра значение этого параметра должно быть null.

Возвращаемое значение

Функция возвращает структуру DocListInfo, содержащую информацию о списке документов после наложения фильтра.

Пример

Предположим, для имеющегося списка документов в наборе обычных поисковых атрибутов есть следующие поисковые атрибуты:

- Наименование (id: 0; тип: строка)
- Вид документа/материала (id: 1; тип: классификатор)
- Номер (id: 3; тип: номер)
- Дата принятия (id: 4; тип: дата)

Предположим также, что необходимо отфильтровать все 'Законы РФ', наименование которых содержит слова 'О праве', номер начинается на '48', принятые в период с 01.12.1992 по 31.05.1993 гг. Следующий пример демонстрирует как можно сформировать условия для фильтра, и наложить его на известный список документов:

searchConditions.php

```
// классы условий поиска
... (см. Пример AttrSearch)
```

FilterList.php

```
<?php

// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl&use=encoded";

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient($wsdl);

// получение набора поисковых атрибутов
... (см. Пример GetSearchAttributes)

// получение списка классификаторов для
// поискового атрибута "Вид документа/материала"
... (см. Пример GetClassificators)

// классы условий фильтра
include 'searchConditions.php';

// формирование массива условий фильтра:

// для поискового атрибута 'Наименование'
$conditions[] = new SearchConditionString(
```

```
    0,                      // id поискового атрибута
    "О праве"               // строка для поиска
    // Если кодировка по умолчанию не UTF-8 (к примеру, windows-1251), то требуется
преобразование.
    // Можно воспользоваться функцией iconv:
    // iconv("WINDOWS-1251", "UTF-8", "О праве")
);

// для поискового атрибута 'Вид документа/материала'
$conditions[] = new SearchConditionClassificator(
    1,                          // id поискового атрибута
    array(902118100),           // id классификатора 'Закон РФ'
    "or"                        // логическая операция
);

// для поискового атрибута 'Номер'
$conditions[] = new SearchConditionNumber(
    3,                      // id поискового атрибута
    "48",                   // строка для поиска
    4                       // режим поиска 'Начинается с'
);

// для поискового атрибута 'Дата принятия'
$conditions[] = new SearchConditionDate(
    4,                          // id поискового атрибута
    array(                      // для выбранного режима поиска нужны два значения
        "01.12.1992",           // дата начала периода
        "31.05.1993"            // дата окончания периода
    ),
    3                           // режим поиска 'Период'
);
// список документов, на которые накладывается фильтр
$list = "551456952,550818249,551782085";
// вызов функции наложения фильтра.
// в $listInfo вернётся информация о списке найденных документов.
$listInfo = $kodweb->FilterList($conditions, $list);

...

?>
```

GETSEARCHAREAS

Функция **GetSearchAreas** достает список вкладок (районов) из поисковой структуры.

Синтаксис

```
function GetSearchAreas (searchName : string) : SearchArea[]
```

Параметры

searchName — имя поисковой структуры для спец.поиска. В случае обычного поиска значение этого параметра должно быть null или "".

Возвращаемое значение

Функция возвращает массив поисковых вкладок (районов) SearchArea поисковой структуры.

Пример

```
<?php

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient("http://localhost/kodeks/api?wsdl&use=encoded");

// получение поисковых вкладок
$searchTab = -1;
$searchTabs = $kodweb->GetSearchAreas(null);

foreach ($searchTabs->item_Area as $tab) {

    // если выбрана вкладка
    if ($tab->id == $searchTab) {
        if (isset($tab->attrs->item_SearchAttribute)) {

            // получаем поисковые атрибуты документов этой вкладки
            $attrsContainer = $tab->attrs->item_SearchAttribute;
                                break;

        }
    }
}

...

?>
```

GETSEARCHLIST

Функция **GetSearchList** возвращает указанную часть списка документов.

Синтаксис

```
function GetSearchList(
    id : string,
    order : int,
    part : int,
    search_Name : string,
    kdocfield : string
) : DocListItem[]
```

Параметры

id — идентификатор списка. Значение, полученное в поле id структуры DocListInfo в результате вызова функций AttrSearch или FuzzySearch.

order — параметр сортировки. Значение может быть:

- NULL - для сортировки по умолчанию;
- элемент массива, полученного с помощью функции GetSearchListOrders,
- номер (нумерация начинается с 0) поискового атрибута, по которому необходимо сортировать список. Сортировка по номеру поискового атрибута неприменима к результатам FuzzySearch. Сортировка по поисковым атрибутам типа 'классификатор' и по поисковому атрибуту "По тексту" не осуществляется. В вышеперечисленных случаях, а так же, если аргумент меньше 0 или больше, либо равен количеству поисковых атрибутов в системе сервер 'выбрасывает' исключение.

part — номер части списка (нумерация начинается с 0).

search_Name — имя поисковой структуры в случае специального поиска. В случае обычного поиска значение этого параметра должно быть null.

kdocfield — Если параметр не задан, то свойство haskdoc элемента возвращаемого массива всегда равно false. Если параметр kdocfield задан, то haskdoc будет равен true, когда у документа есть текст в формате kdoc.

Возвращаемое значение

Функция возвращает массив структур DocListItem.

Пример

```
<?php

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient("http://localhost/kodeks/api?wsdl&use=encoded", array('exceptions' =>
true));

... (см. Пример AttrSearch)

// $listInfo содержит информацию о списке найденных документов.

// получение списка документов
try {
   $documents = $kodweb->GetSearchList(
      $listInfo->id,          // идентификатор списка
      0,                      // сортировка по поисковому атрибуту с индесом 0
      1,                      // запрос второй части списка
      NULL,                   // имя поисковой структуры для специального поиска
      0                       // наличие текста в формате kdoc
   );
} catch (SoapFault $err) {
   echo "Error: ", $err->getMessage(), "\n";
   return;
}

// вывод списка документов
```

```
foreach ($documents as $doc) {
        echo $doc->nd, "\n";
        echo iconv("UTF-8", "cp866", $doc->name), "\n";    // перекодировка для вывода на консоль
        echo iconv("UTF-8", "cp866", $doc->info), "\n\n";
}

...

?>
```

GETCLASSIFICATORS

Функция **GetClassificators** возвращает список классификаторов поискового атрибута.

Синтаксис

```
function GetClassificators(
   attr_Id : int,
   search_tabs_index : int,
   parent_Id : any,
   src_List_Id : string,
   searchName : string,
) : Classificator[]
```

Параметры

attr_Id — идентификатор поискового атрибута, для которого строится список классификаторов. Поисковый атрибут должен иметь тип 'классификатор'.

search_tabs_index — индекс поисковой вкладки. При использовании можно задавать конкретную вкладку.

parent_Id — идентификатор родительского классификатора (для древовидных классификаторов). Для получения списка классификаторов верхнего уровня значение этого параметра должно быть null или 0.

srcList_Id — идентификатор списка документов, для которого строится список классификаторов (в текущей версии API значение этого параметра должно быть null).

search_Name — имя поисковой структуры в случае специального поиска. В случае обычного поиска значение этого параметра должно быть null.

Возвращаемое значение

Функция возвращает массив структур Classificator.

Пример

```
<?php
```

```
// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient("http://localhost/kodeks/api?wsdl&use=encoded");

// получение набора поисковых атрибутов
$searchAttrs = $kodweb->GetSearchAttributes();

// получение списков классификаторов для
// поисковых атрибутов типа 'классификатор'
foreach ($searchAttrs->item_SearchAttribute as $attr) {
   if ($attr->type != 5 /*классификатор*/) {
      continue; // если тип не 'классификатор' - перейдём к следующему элементу
   }

   // полученные списки сохраним как элементы массива classificators
   // с индексами, равными идентификаторам соответствующих поисковых атрибутов
   $classificators[$attr->id] = $kodweb->GetClassificators($attr->id, null, null, null, null);
}

...

?>
```

GETSEARCHATTRIBUTES

Функция **GetSearchAttributes** возвращает список всех поисковых атрибутов в системе. Поисковые атрибуты применяются при задании условий поиска.

Синтаксис

```
function GetSearchAttributes(searchName : string) : SearchAttribute[]
```

Параметры

search_Name — имя поисковой структуры в случае специального поиска. В случае обычного поиска значение этого параметра должно быть null.

Возвращаемое значение

Функция возвращает массив структур SearchAttribute, описывающий поисковые атрибуты.

Пример

```
<?php

// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl&use=encoded";

// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient($wsdl);
```

```
// получение набора поисковых атрибутов
$searchAttrs = $kodweb->GetSearchAttributes();

// далее может следовать код, формирующий, например, пользовательский
// интерфейс поискового запроса

...

?>
```

GETSEARCHLISTORDERS

Функция **GetSearchListOrders** достает перечень возможных сортировок из списка найденных документов.

Синтаксис

```
function GetSearchListOrders(
   id : string,
   search_Name : string
) : OrderListItem[]
```

Параметры

id — идентификатор списка. Значение, полученное в поле id структуры DocListInfo в результате вызова функций AttrSearch, FuzzySearch или FilterList.

search_Name — имя поисковой структуры в случае спец.поиска. В случае обычного поиска значение этого параметра должно быть null.

Возвращаемое значение

Функция возвращает перечень возможных сортировок в виде массива структур OrderListItem. Элементы массива могут использоваться в качестве параметра сортировки order в функции GetSearchList.

Пример

```
<?php

// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl&use=encoded";

... (см. Пример AttrSearch)

// $listInfo содержит информацию о списке найденных документов

// получение списка документов
try {
     $orders = $kodweb->GetSearchListOrders(
```

```
        $listInfo->id,      // идентификатор списка
        NULL                // имя поисковой структуры для специального поиска
    );
    $documents = $kodweb->GetSearchList(
        $listInfo->id,                          // идентификатор списка
        $orders->item_OrderListItem[0],     // сортировка по названию документа
        1,                                      // запрос второй части списка
        NULL,                                   // имя поисковой структуры для специального поиска
        NULL                                    // наличие текста в формате kdoc
    );
} catch (SoapFault $err) {
        echo "Error: ", $err->getMessage(), "\n";
        return;
}

// вывод списка документов
foreach ($documents as $doc) {
        echo $doc->nd, "\n";
        echo iconv("UTF-8", "cp866", $doc->name), "\n";    // перекодировка для вывода на консоль
        echo iconv("UTF-8", "cp866", $doc->info), "\n\n";
}

...

?>
```

GETNEWSLISTS

Функция **GetNewsLists** возвращает список новостных рубрик.

Синтаксис

```
function GetNewsLists() : NewsList[]
```

Параметры

отсутствуют

Возвращаемое значение

Функция возвращает массив структур NewsList, описывающий новостные рубрики.

Пример

```
<?php

// url запроса wsdl-описания
$wsdl = "http://localhost/kodeks/api?wsdl";
```

```
// создание SOAP-клиента на основе wsdl-описания
$kodweb = new SoapClient($wsdl);

// получение набора новостных рубрик
$headings = $kodweb->GetNewsLists();

// далее может следовать код, формирующий, например, пользовательский
// интерфейс показа новостных лент и переходов на страницы с полными текстами
новостей
...

?>
```

СТРУКТУРЫ

CLASSIFICATOR

Структура **Classificator** описывает классификатор.

Синтаксис

```
struct Classificator {
   id :  int,
   name : string,
   hasChildren : boolean
}
```

Поля

id — идентификатор классификатора.

name — имя классификатора.

hasChildren — флаг, указывающий, является ли данный классификатор классификатором верхнего уровня (содержит ли дочерние элементы).

DOCLISTINFO

Структура **DocListInfo** содержит информацию о списке документов, полученном в результате поиска, атрибутного или интеллектуального.

Синтаксис

```
struct DocListInfo {
   id : string,
   size : int,
   parts : int
   dpp : int
}
```

Поля

id — идентификатор списка. Этот идентификатор используется в вызове функции GetSearchList при запросе части списка.

size — количество элементов списка.

parts — количество частей списка.

dpp — количество документов в части списка.

DOCLISTITEM

Структура **DocListItem** содержит информацию о документе из списка документов, возвращенного функцией GetSearchList.

Синтаксис

```
struct DocListItem {
    nd : int,
    r : int,
    type : string,
    name : string,
    info : string,
    haskdoc : boolean
}
```

Поля

nd — внутренний номер документа. Используется в вызове функции GetDocument.

r — номер раздела. Используется в вызове функции GetDocument. Может быть null

type — тип документа. Значение, объединяющее статус (например, действующий/недействующий) и принадлежность документа к категории информации (документы, справки и т.п.).

name — название документа.

info — краткая аннотация к документу.

haskdoc — наличие в документе текста в формате kdoc.

NEWSLIST

NewsList описывает новости определенной рубрики.

Синтаксис

```
struct NewsList {
    id : int,
    news : NewsTape[],
    title : string
}
```

Поля

id — идентификатор. Может использоваться для формирования URL перехода к полному списку новостей рубрики - /d?nd=id

news — массив самых свежих новостей рубрики.

title — название новостной рубрики.

NEWSTAPE

NewsTape описывает элемент списка новостей.

Синтаксис

```
struct NewsTape {
   date : date,
   id : int,
   title : string
}
```

Поля

date — дата новости.

id — идентификатор. Может использоваться для формирования URL перехода к полному тексту новости - /d?nd=id

title — заголовок новости.

ORDERLISTITEM

Структура **OrderListItem** содержит информацию о сортировке документов, полученных в результате поиска.

Синтаксис

```
struct OrderListItem {
   order : string,
   name : string,
   defaultOrder : boolean
}
```

Поля

order — параметры сортировки.

name — название сортировки.

defaultOrder — значение 'истина' означает, что эта сортировка используется по умолчанию в функции GetSearchList.

SEARCHAREA

Структура SearchArea описывает поисковой вкладку (район). Используется в поиске.

Синтаксис

```
struct SearchArea {
   id : int,
   name : string,
   attrs : SearchAttribute[]
}
```

Поля

id — индекс. Это значение присвоенно поисковой вкладке (району) системой. Ниже приведены индексы районов обычного поиска:

-1 - "Все"

0 - "Законодательство России"

1 - "Комментарии, консультации"

2 - "Международное право"

3 - "Нормы, правила, стандарты"

4 - "Образцы и формы"

5 - "Региональное законодательство"

6 - "Справки"

7 - "Судебная практика"

8 - "Техническая документация"

9 - "Электронные публикации"

10 - "Технические описания"

11 - "Корреспонденция счетов"

12 - "Проектная документация"

16 - "Международные стандарты"

17 - "Новости"

18 - "Кодекс/Техэксперт: Банк документов@

Используется в поиске, чтобы задать параметр **search_tabs_index** для получения документов конкретной вкладки (района).

name — имя, присвоенное району системой.

attrs — массив посковых атрибутов документов этого района. Поисковые атрибуты применяются при задании условий поиска.

SEARCHATTRIBUTE

Структура **SearchAttribute** описывает поисковый атрибут. Используется в атрибутном поиске.

Синтаксис

```
struct SearchAttribute {
    id : int,
    type : int,
    name : string,
    hintId : string,
    isGlobal : boolean
}
```

Поля

id — идентификатор поискового атрибута. Это значение, присвоенное поисковому атрибуту системой. Используется при формировании условия поиска по конкретному поисковому атрибуту.

type — тип поискового атрибута. От значения этого поля зависит, какой из типов SearchCondition<XXX> следует использовать для формирования условий поиска по данному поисковому атрибуту. Возможные значения и соответствующие им структуры условий поиска:

1 - Дата; SearchConditionDate

2 - Строка; SearchConditionString

3 - Номер; SearchConditionNumber

5 - Классификатор; SearchConditionClassificator

name — имя поискового атрибута.

hintId — идентификатор для получения контекстной подсказки.

isGlobal — true означает, что атрибут используется на вкладке "Все"; иначе false.

SEARCHCONDITIONCLASSIFICATOR

Структура **SearchConditionClassificator** описывает условие поиска для поискового атрибута типа 'классификатор'.

Синтаксис

```
struct SearchConditionClassificator {
    id : int,
    values : int[],
    lop : string
}
```

Поля

id — идентификатор поискового атрибута.

value — массив значений для поиска, состоящий из идентификаторов классификаторов.

lop — логическая операция. Допустимые значения:

"or" — ИЛИ (объединение); в случае задания двух и более значений классификаторов будут найдены документы, соответствующие хотя бы одному из значений.

"and" — И (пересечение); в случае задания двух и более значений классификаторов будут найдены документы, соответствующие всем значениям сразу. Иными словами, если документ не соответствует, хотя бы одному значению из набора - в результат поиска он не попадёт.

"not" — КРОМЕ (отрицание); будут найдены все документы, кроме соответствующих заданному набору классификаторов. Иначе говоря, если документ соответствует, хотя бы одному значению из набора - в результат поиска он не попадёт.

SEARCHCONDITIONDATE

Структура **SearchConditionDate** описывает условие поиска для поискового атрибута типа 'дата'.

Синтаксис

```
struct SearchConditionDate {
    id : int,
    values : string[],
    mode : int
}
```

Поля

id — идентификатор поискового атрибута.

value — массив значений для поиска, состоящий из одного или двух элементов, в зависимости от режима поиска. Элементами массива должны быть строковые представления дат в формате 'дд.мм.гггг'. Для режимов 0, 1, 2 используется один (0-й) элемент массива, для режимов 3 и 4 используются оба элемента массива.

mode — режим поиска:

0 — Точно; ищутся только точные совпадения дат

1 — По; ищутся даты, до указанной включительно

2 — С; ищутся даты, начиная с указанной

3 — Период; ищутся даты, попадающие в указанный период

4 — Вне периода; ищутся все даты, кроме попадающих в указанный период

SEARCHCONDITIONNUMBER

Структура **SearchConditionNumber** описывает условие поиска для поискового атрибута типа 'номер'.

Синтаксис

```
struct SearchConditionNumber {
   id : int,
   value : string,
   mode : int
}
```

Поля

id — идентификатор поискового атрибута.

value — значение для поиска. Если строка содержит русские буквы, то она должна быть в кодировке UTF-8.

mode — режим поиска:

1 — Точно; ищутся только точные совпадения

2 — Содержит; ищутся номера, которые имеют вхождение заданного значения

4 — Начинается с; ищутся номера, которые начинаются с заданного значения

SEARCHCONDITIONSTRING

Структура **SearchConditionString** описывает условие поиска для поискового атрибута типа 'строка'.

Синтаксис

```
struct SearchConditionString {
   id : int,
   value : string
}
```

Поля

id — идентификатор поискового атрибута.

value — строка для поиска. Если строка содержит русские буквы, то она должна быть в кодировке UTF-8.