

Сервисы CarabiQuery и DoclistStructure

Общие сведения

Сервисы предназначены для формирования динамических SQL-запросов к базе данных документов, получения структурированных списков (доклистов, ссылочных полей), а также вспомогательной информации о колонках и фильтрах.

- **CarabiQuery** – основной сервис для выборки данных из доклистов, референсных списков и справочников типов документов.
- **DoclistStructure** – вспомогательный сервис для получения метаданных колонок и структуры доклистов/рефлистов.

Для запросов, требующих авторизации, в объекте запроса обязательно наличие поля `session` с корректной сессией пользователя (содержит `userId`).

Сервис CarabiQuery

1. `getDoclistDataSql`

Назначение: получить SQL-запрос для выборки данных доклиста без его выполнения.
Права: требует `needAdmin: true` .

Параметры запроса:

Поле	Тип	Обязательное	Описание
<code>docListId</code>	number	да	Идентификатор доклиста
<code>statusList</code>	string	нет	Список ID статусов через запятую (например <code>"1,2,3"</code>)
<code>filter</code>	string	нет	Текстовый фильтр (поиск по контексту или номеру документа)
<code>filterCols</code>	string	нет	JSON-фильтр по колонкам (см. формат в <code>FiltersBuilder</code>)
<code>xmlFilter</code>	string	нет	Дополнительный XML-фильтр, подменяющий <code>XML_SEARCH</code> доклиста

Поле	Тип	Обязательное	Описание
orderBy	string	нет	Имя колонки и направление сортировки (напр. "EVENT_DATE DESC")
resTreeValue	string	нет	Код классификатора для фильтрации по дереву ресурсов

Формат ответа:

```
{
  "SQL": "<сгенерированный SQL-запрос>"
}
```

2. fetchDoclistData

Назначение: получить страницу данных доклиста.

Права: доступно авторизованным пользователям.

Параметры: те же, что у `getDoclistDataSql`, плюс параметры пагинации:

Поле	Тип	По умолчанию	Описание
count	number	10	Количество записей на страницу
offset	number	0	Смещение (начиная с 0)

Ответ:

Массив объектов, соответствующих строкам доклиста. Поля каждой строки включают:

- `document_id` – идентификатор документа
- `doc_status_descr` – описание статуса
- `event_date` – дата события (строка в формате DD.MM.YYYY HH24:MI:SS)
- `doc_status_owner` – ФИО создателя
- `doc_status_modifier` – ФИО изменившего
- `doc_descr` – полное наименование документа
- `doc_status_id`, `doc_status_name` – системные поля статуса
- `doc_status_owner_id` – ID создателя
- `pr_special` – служебное поле
- `attach_count`, `child_count`, `files` – счётчики вложений и дочерних документов
- `PR_<имя_свойства>` – динамические колонки, соответствующие свойствам документа (имена начинаются с `PR_`)

Пример:

```
[
  {
    "document_id": 12345,
    "doc_status_descr": "Утверждён",
    "event_date": "15.01.2025 10:30:00",
    "doc_status_owner": "Иванов Иван Иванович",
    "doc_status_modifier": "Петров Пётр Петрович",
    "doc_descr": "Договор №123 от 10.01.2025",
    "doc_status_id": 2,
    "doc_status_name": "APPROVED",
    "doc_status_owner_id": 101,
    "pr_special": null,
    "attach_count": 3,
    "child_count": 0,
    "files": 2,
    "pr_number": "123",
    "pr_date": "10.01.2025",
    ...
  }
]
```

3. `getDoclistDataCnt`

Назначение: получить общее количество записей в доклисте с учётом фильтров.

Права: авторизованные пользователи.

Параметры: совпадают с `getDoclistDataSql` (без пагинации).

Ответ: число (integer) – количество документов.

4. `fetchDoclistDataRow`

Назначение: получить одну строку доклиста по идентификатору документа.

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
<code>docListId</code>	number	да	ID доклиста
<code>documentId</code>	number	да	ID конкретного документа

Ответ:

Объект с данными строки (такой же, как в массиве `fetchDoclistData`) или `null`, если документ

не найден.

5. `fetchReflistData`

Назначение: получить данные ссылочного поля (референс-листа) для конкретного документа.

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
<code>docKind</code>	string	да	Системное имя типа документа, из которого просматривается ссылка
<code>documentId</code>	number	да	ID текущего документа
<code>docPropId</code>	number	да	ID свойства-ссылки (идентификатор свойства в модели)
<code>parentId</code>	number	нет	ID родительского документа для иерархических ссылок (если дерево)
<code>statusId</code>	string	нет	Фильтр по статусам целевых документов (через запятую)
<code>filter</code>	string	нет	Текстовый фильтр
<code>filterCols</code>	string	нет	JSON-фильтр по колонкам
<code>orderBy</code>	string	нет	Сортировка
<code>count</code>	number	нет (10)	Размер страницы
<code>offset</code>	number	нет (0)	Смещение

Ответ: массив объектов с данными целевых документов (структура аналогична `fetchDoclistData`).

6. `getReflistDataCnt`

Назначение: получить количество документов в ссылочном списке.

Права: авторизованные пользователи.

Параметры: аналогичны `fetchReflistData`, но без `count` / `offset`.

Ответ: число.

7. `fetchReflistDataRow`

Назначение: получить одну строку из ссылочного списка (целевой документ) по его ID.

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
documentId	number	да	ID целевого документа

Ответ: объект строки или `null`.

8. `fetchReflistOptions`

Назначение: получить список возможных документов для назначения в ссылку (диалог выбора).

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
docKind	string	да	Системное имя типа документа-источника
documentId	number	нет	ID текущего документа (для подстановки переменных в XML-фильтр)
docPropId	number	да	ID свойства-ссылки
dockindIdProp	number	да	ID типа документа, доступного для выбора
filter	object	нет	Расширенные фильтры. Состав: <code>xml</code> (XML-фильтр), <code>statusIds</code> , <code>context</code> , <code>columns</code> , <code>classifId</code>
orderBy	string	нет	Сортировка
count	number	нет (10)	Размер страницы
offset	number	нет (0)	Смещение

Ответ: массив объектов доступных документов.

9. `fetchDockindData`

Назначение: получить список документов заданного типа (без привязки к конкретному свойству-ссылке).

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
docKind	string	да	Системное имя типа документа
statusList	string	нет	Список ID статусов через запятую
xmlFilter	string	нет	XML-фильтр
filter	string	нет	Текстовый фильтр
filterCols	string	нет	JSON-фильтр по колонкам
orderBy	string	нет	Сортировка
count	number	нет (10)	Размер страницы
offset	number	нет (0)	Смещение

Ответ: массив объектов документов (формат аналогичен `fetchDoclistData`).

10. `getSqlFromXml`

Назначение: отладочный метод для получения сгенерированного SQL из произвольного XML.

Права: `needAdmin: true` .

Параметры:

Поле	Тип	Обязательное	Описание
xml	string	да	XML-строка с блоком <code><query><formal ...></code>

Ответ: объект с единственным полем `SQL` (строка).

Сервис `DoclistStructure`

1. `getDoclistColumns`

Назначение: получить полную структуру колонок доклиста с учётом пользовательских настроек (ширина, видимость, порядок).

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
doclistId	number	да	ID доклиста

Ответ: массив объектов, каждый из которых описывает одну колонку:

Поле	Тип	Описание
SHOW_ORDER	number	Порядковый номер для отображения
DOCPROP_ID	number	ID свойства модели (для системных колонок < 0)
DOCPROP_NAME	string	Человекочитаемое название
PROP_SYS_NAME	string	Системное имя колонки (напр. DOCUMENT_ID , PR_DATE , DOC_STATUS_DESCR)
DOCPROP_KIND	number	Тип свойства
DOCPROP_OBJECT	string	Код типа данных ('1' - целое, '4' - строка, '32' - дата и т.д.)
MULTI	number	Признак множественного значения
SYS_COLUMN	number	Флаг системной колонки
WIDTH	number	Ширина в пикселях (пользовательская или по умолчанию)
VISIBLE	number	Флаг видимости (1 - показывать, 0 - скрыта)
ORDERBY	string	null
DOCLIST_ID	number	null
IS_FIXED , IS_GROUP и др.	number	Флаги расширенных настроек колонки

2. getReflistStructure

Назначение: получить полную информацию для отображения ссылочного списка, включая колонки, статусы и доступные XML-фильтры.

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
docpropId	number	да	ID свойства-ссылки
dockindId	number	нет	Явное указание типа документов (обычно определяется автоматически)

Ответ: объект с полями:

- `columns` – массив колонок целевого типа документов (формат как в `getDoclistColumns`)
- `info` – объект с базовой информацией о типе (поля `DOCKIND_ID`, `DOCKIND_NAME`, ...)
- `statusColors` – массив объектов статусов с цветовой индикацией
- `xmlFilters` – массив предустановленных XML-фильтров для выбора

3. `getReflistColumns`

Назначение: получить колонки для референс-листа (аналогично `getDoclistColumns`, но без привязки к конкретному доклисту).

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
<code>docpropId</code>	number	нет	ID свойства-ссылки
<code>dockkindId</code>	number	нет	ID типа документов

Ответ: массив колонок (та же структура, что и в `getDoclistColumns`).

4. `getReflistOptionsColumns`

Назначение: получить колонки для диалога выбора документа в ссылку (обычно те же, что и в рефлисте, но могут отличаться правами).

Права: авторизованные пользователи.

Параметры:

Поле	Тип	Обязательное	Описание
<code>docpropId</code>	number	да	ID свойства-ссылки
<code>propDockkindId</code>	number	да	ID типа документов для выбора

Ответ: массив колонок.

Примечания по фильтрации

Текстовый фильтр (`filter`)

- Если переданная строка состоит только из цифр и длиннее 4 символов, сервис сначала пытается найти документ по `DOCUMENT_ID` в указанном типе.
- Если документ не найден или фильтр не числовой, выполняется поиск подстроки без учёта регистра в поле `doc_descr` (через `LIKE '%...%'`).

JSON-фильтр по колонкам (`filterCols`)

Ожидается строка в формате JSON с массивом условий:

```
{
  "filter": [
    {
      "field_name": "PR_NUMBER",
      "value": "123",
      "operator": "="
    }
  ]
}
```

- `field_name` – системное имя колонки (например, `DOCUMENT_ID`, `EVENT_DATE`, `PR_*`).
- `value` – значение для сравнения.
- `operator` – оператор (`=`, `!=`, `<`, `>`, `LIKE`, и т.д.). Если не указан, для строковых полей применяется `LIKE`, для числовых – `=`, для дат – `>=` или `<=`.

XML-фильтр (`xmlFilter`, `xmlSearch`)

Формат соответствует внутреннему представлению условий в подсистеме `LegacyXml2Select`.

Пример:

```
<query>
  <formal dockind_id="123" namevar="MYDOCTYPE" haschild="0" docrefs="1">
    <property docprop_id="100" condition="e" doc_prop_value="Hello"/>
    <reference docprop_id="200" condition="e" referencing="join">
      <status doceventkind_id="1"/>
    </reference>
  </formal>
</query>
```

Для деталей синтаксиса обратитесь к документации по `LegacyXml2Select`.

Аутентификация и сессия

Все запросы (кроме `getDoclistDataSql` и `getSqlFromXml`, требующих `needAdmin`) доступны обычным авторизованным пользователям.

Объект запроса должен содержать поле `session` с пользовательской сессией:

```
{
  "session": {
```

```
"userId": 123,  
...  
,  
"docListId": 42,  
...  
}
```

При использовании HTTP-транспорта сессия обычно передаётся через механизмы платформы Morphcluster (куки, заголовки) – в документации по ядру уточните способ передачи.

Revision #2

Created 21 June 2026 22:42:02 by Admin

Updated 26 June 2026 15:31:30 by Admin