

# Документация Comexp API

## Содержание

1. Введение
2. Общие сведения о поиске видео
  - 2.1 Параметры видео. Определения и описания
  - 2.2 Параметры видео применительно к API
  - 2.3 Принципы поиска
    - 2.3.1 Основные определения
    - 2.3.2 Описание принципа работы системы
3. REST API
  - 3.1. Описание технологии REST
  - 3.2. Описание Comexp API
    - 3.2.1 Описание параметров
    - 3.2.2 Авторизация
    - 3.2.3 Описание параметров
    - 3.2.4 Работа с видеофрагментами
      - Создание (загрузка) нового видео
      - Получение свойства видеофрагмента
      - Присвоение видеофрагменту нового набора тегов
      - Удаление видео
      - Получение списка видеофрагментов
    - 3.2.5 Работа с корзинами фрагментов
      - Создание новой корзины
      - Получение параметров корзины
      - Изменение параметров корзины
      - Удаление корзины
      - Получение списка корзин
    - 3.2.6 Работа с заданиями на поиск
      - Создание нового задания на поиск
      - Получение параметров задания на поиск
      - Изменение параметров задания на поиск
      - Удаление задания на поиск
      - Получение списка заданий на поиск

## 1. Введение

API компании Comexp предоставляет возможность разработчиками приложений осуществить поиск видео-фрагмента в наборе видео-фрагментов. Для реализации этой возможности в API предусмотрен следующий набор функций:

- загрузка пользовательских видео-файлов;
- определение множеств видео-файлов с помощью интервалов времени и меток;
- создание заданий поиска совпадений между двумя множествами видео-файлов.

## 2. Общие сведения о поиске видео

### 2.1. Параметры видео. Определения и описания

- Количество кадров в секунду (frames per second, fps) — это число неподвижных изображений, сменяющих друг друга при показе одной секунды видеоматериала и создающих эффект движения объектов на экране. Чем больше частота кадров в секунду, тем более плавным и естественным будет казаться движение.
- Чересстрочная развертка. Развертка материала может быть прогрессивной (построчной) или чересстрочной. При прогрессивной развертке все горизонтальные линии (строки) изображения отображаются поочередно, одна за другой. При чересстрочной же развертке показываются попеременно то все четные строки, то нечетные.
- Разрешение. Бывает горизонтальным и вертикальным. Измеряется в пикселях. Записывается как произведение количества точек по горизонтали на количество точек по вертикали (строк). Чем выше разрешение, тем лучше качество видео.
- Соотношение сторон экрана. Соотношение высоты и ширины кадра является важным параметром в любом видеоматериале. Самыми распространенными являются соотношения 4:3, 16:9.
- Количество цветов и цветовое разрешение. Эти параметры описываются цветовыми моделями по аналогии с компьютерными моделями RGB, CMYK. Количество цветов в видеоматериале определяется числом бит, отведенным для кодирования цвета каждого пикселя.
- Битрейт или ширина видеопотока. Ширина, скорость или битрейт видеопотока — это количество обрабатываемых бит информации за секунду времени (бит/с). Чем выше битрейт, тем лучше качество видео. Различают два вида управления шириной потока в видеокодеке: постоянный битрейт (constant bit rate, CBR), переменный битрейт (variable bit rate). Переменный битрейт позволяет сэкономить на занимаемой видеоинформацией памяти путем уменьшения скорости видеопотока на относительно статичных сценах.
- Размеры рабочей области кадра. При выполнении сравнения видео-фрагментов в процессе сравнения может участвовать не вся площадь кадра, а только её часть.

### 2.2. Параметры видео применительно к API

- Количество кадров в секунду. Может быть любым, но с одним условием - количество кадров в секунду у всех видео-фрагментов, участвующих в поиске, должно быть одинаковым.
- Чересстрочная развертка. Используется чересстрочная развертка.
- Разрешение. Минимальное разрешение кадра 100 x 100.

- Соотношение сторон экрана. Может быть любым, хотя параметр и не принимает участия в расчетах.
- Количество цветов и цветовое разрешение. Может быть любым.
- Битрейт. Может быть любым.
- Размеры рабочей области кадра. В поиска участвует часть кадра, ограниченная сверху 12.5%, снизу 12.5%, слева 2.9%, справа 2.9%.

## **2.3. Принципы поиска**

### **2.3.1 Основные определения**

Видео-фрагмент — видео-файл, который будет использоваться в поиске.

Корзина — набор видео-фрагментов. Для помещения видео-фрагментов в корзину используются метки и диапазоны времени.

Задание на поиск — пользовательские задания на поиск совпадений между двумя множествами видео, заданными с помощью корзин.

### **2.3.2 Описание принципа работы системы**

Для поиска видео в API Comexp применяется технология digital fingerprint. Она условно состоит из 2-х принципиально важных алгоритмов: выделение из видео-потока «цифровых отпечатков» (TAPe-формат) и непосредственно поиск по этим отпечаткам. Первый алгоритм автоматически выполняется при загрузке любого видео-фрагмента на сервер api.comexp.ru. Второй реализован по схеме с заданиями на поиск.

После преобразования исходного видео-фрагмента в TAPe-формат загруженный видео-файл уничтожается. Все дальнейшие процедуры работают только с TAPe-данными. Таким образом, нельзя использовать api в качестве хранилища видео.

Каждому видео-фрагменту при загрузке на сервер API присваивается уникальный идентификатор – video\_id. Так же для видео-фрагмента доступны характеристики:

- filename – имя файла;
- ts – дата, время начала видео, а также любое количество произвольных тегов;
- length – длительность фрагмента в секундах. Дополнительная характеристика, доступная только для чтения. Рассчитывается автоматически.

Для манипулирования наборами видео-фрагментов в системе предусмотрены корзины видео. При создании корзины имеется возможность указать несколько полей:

- bin\_name — название корзины;
- ts\_from — дата, время начала видео в корзине;
- ts\_to — дата, время окончания видео в корзине;
- также возможно задать любое количество произвольных тегов.

Так же как и фрагмент, корзина получает при создании уникальный идентификатор – bin\_id.

Условие попадания конкретного видеофрагмента в корзину следующее: одновременно все теги корзины в точности должны совпадать с тегами фрагмента, а также момент времени фрагмента ts (если задан) попадает в интервал времени корзины между ts\_from и ts\_to (если заданы).

Следует обратить внимание, что все фрагменты в корзине в случае использования моментов начала фрагментов ts выстраиваются в одну временную линию. В этом варианте отслеживаются наложения.

Поиск видео построен по схеме, включающей произвольное число заданий. Каждое

задание на поиск содержит следующие обязательные поля:

- `search_name` – название задания;
- `bin_id_1` — корзина фрагментов, в которой ищем;
- `bin_id_2` — корзина фрагментов которые ищем (эталон).

Задание подразумевает поиск ВСЕХ видеофрагментов из корзины `bin_id_2` по ВСЕМ фрагментам из корзины `bin_id_1`. При создании каждому заданию присваивается уникальный идентификатор – `search_id`. В задании есть два параметра доступных для чтения: `state` (статус окончания поиска), `last_update` (дата время последнего изменения результатов поиска). Каждое задание попадает в очередь на исполнение.

После окончания выполнения поиска статус задания устанавливается в значение «`ready`», и становится возможным получить результаты. Результат поиска – таблица, содержащая найденные совпадения видео. Поля:

- `video_id_1` — видео фрагмент, в котором нашлось совпадение;
- `video_id_2` — видео фрагмент, который «нашелся»;
- `start_1` — смещение в секундах относительно начала исходного фрагмента;
- `start_2` — смещение в секундах относительно начала эталона;
- `length` — длительность совпадения, меньше или равная длине эталона.

Важно отметить, что в случае, когда задание на поиск «отработано» и при этом происходит добавление видео в корзину `bin_id_1` и/или `bin_id_2`, то в этом случае статус готовности «сбрасывается» и задание снова попадает в очередь на исполнение. В этом случае происходит дополнительный поиск только по новым видеофрагментам.

## 3. REST API

### 3.1. Описание технологии REST

REST - REpresentational State Transfer. Архитектура REST подразумевает наличие клиент-серверной архитектуры. Клиент инициирует запросы к серверу, сервер обрабатывает их и возвращает ответ. Обычно, если говорят об этой архитектуре, то в рамках HTTP. В общем случае REST применимо на любых уровнях приложений.

REST API определяет набор функций, к которым приложения могут совершать запросы и получать ответы. Взаимодействие происходит по протоколу HTTP. Преимуществом такого подхода является широкое распространение протокола HTTP, поэтому REST API можно использовать практически из любого языка программирования.

Все вызовы методов API – это GET, POST, PUT или DELETE представляют собой HTTP-запросы к URL `http://api.comexp.ru` с некоторым набором параметров. Список возможных методов API, а так же необходимые параметры и возвращаемые значения, приведены в данном документе. Кодировка результата – UTF-8. Формат ответа – XML.

Например, на PHP для осуществления такого запроса можно использовать `cURL`, на Perl — `LWP::Simple`, на Python — `httplib`.

Обратите внимание на готовые библиотеки, возможно, для вашего языка программирования уже существует реализация REST API.

Данные запроса могут передаваться в виде `query`-строки (после знака “?”) при использовании метода GET, либо в теле POST-запроса. Помните, что в случае GET-запроса, параметры должны быть закодированы с помощью URL encoding (потому что для URL доступны только символы латинского алфавита).

## 3.2. Описание Comexp API

### 3.2.1 Авторизация

При взаимодействии с сервером API пользователь идентифицирует себя с помощью уникального ключа.. Все запросы к API должны содержать ключ в HTTP заголовке.

Пример строки HTTP заголовка (указанный ключ недействителен):

X-API-Key: 5A6BB49E098111DFABA000042388EF30

### 3.2.2 Описание параметров

Дополнительные параметры, которые передаются в HTTP-запросе, указаны в разделе «Параметры» описания каждого метода.

Все уникальные идентификаторы (ID) в системе представляют собой GUID. Представление в запросах и ответах – hex-строка ( до 32 символов).

Дата, время – строковое представление в формате ГГГГММДДччммсс (4 цифры года, 2 цифры месяца, 2 цифры дня месяца, 2 цифры часов, 2 цифры минут, 2 цифры секунд).

Тэги – произвольные символьные строки без пробелов. Список тэгов – это строка, где тэги разделены символом слэш «/».

### 3.2.3 Обработка ошибок

При выполнении HTTP-запросов приложение-клиент получает от сервера код ответа.

В случае отсутствия какой либо ошибки этот код будет иметь значение 200.

Коды ответа в ошибочных ситуациях:

Код	Описание
-----	----------

- |     |  |
|-----|--|
| 403 | Invalid API-key - Вы указали неверный ключ разработчика<br>Возвращаемое сообщение «<errors><error><domain>access</domain><code>invalid api key</code></error></errors>»  |
| 404 | Not Found – Вы указали неверное имя API-метода(1) иди идентификатор объекта системы (видеофрагмент, корзина или задание на поиск)(2)<br>Возвращаемые сообщения<br>1 - «<errors><error><domain>http</domain><code>Not Found</code></error></errors>»<br>2 - «<errors><error><domain>file</domain><code>Not Found</code></error></errors>» |
| 500 | Invalid parameter – Вы указали неверный параметр в HTTP-запросе или REST-запросе.  |

### 3.2.4 Работа с видеофрагментами

#### Создание (загрузка) нового видео

Описание	Метод загружает видеофрагмент в систему.
----------	--

Структура REST `http://api.comexp.ru/video/[tag1]/[tag2]/.../[name]@[ts]`  
запроса

1. `tag1,tag2,...` - Тэги, которые присваиваются загружаемому видеофрагменту.
2. `name` - Имя видеофрагмента под которым он будет храниться в системе. Если параметр `name` не указан, то используется имя загружаемого файла. Не должно быть уникальным.
3. `ts` – Дата и время начала видеофрагмента в формате API Comexp.

Метод PUT

Параметры Только API-key

Возвращаемое В случае успешного добавления, метод возвращает уникальный значение идентификатор видеофрагмента в системе.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" --  
upload-file clip.avi http://api.comexp.ru/video/tag1/tag2/  
<video_id>01B9037D24FC816097ED77590985E5AC</video_id>
```

Замечание После загрузки автоматически начинает выполняться процедура выделения «цифровых отпечатков». Для дальнейшей работы видеофрагмент будет доступен после того, как свойство `state` станет равным 5.

### Получение свойств видеофрагмента

Описание Извлекает свойства видеофрагмента из системы.

Структура REST `http://api.comexp.ru/video/<video_id>`  
запроса

1. `video_id` –Идентификатор (ID) видеофрагмента.

Метод GET

Параметры Только API-key

Возвращаемое значение

Свойства видеофрагмента.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" http://api.comexp.ru/video/c4c95b02d19a7ff6395542f205de989b
<video>
<id>C4C95B02D19A7FF6395542F205DE989B</id>
<name>f0.avi</name>
<state>5</state>
<length>90</length>
<tags><tag>tag3</tag><tag>tag4</tag></tags>
</video>
```

Описание свойств видеофрагмента:

**id** Идентификатор видеофрагмента.

**name** Имя видеофрагмента.

**state** Состояние видеофрагмента. state = 5 говорит о готовности видеофрагмента к дальнейшей обработке. Остальные состояния промежуточные.

**length** Дилна видеофрагмента в секундах

**tags** Тэги назначенные видеофрагменту

### Присвоение видеофрагменту нового набора тегов

Описание

Метод предназначен для замены набора тегов у видеофрагмента.

Структура запроса

REST `http://api.comexp.ru/video/<video_id>`  
1. video\_id – уникальный идентификатор (ID) видеофрагмента.

Метод

POST

Параметры

tags – Список новых тэгов, которые присваиваются видеофрагменту. Тэги разделяются слэшем «/». Если список тэгов пуст, то с видеофрагментом не будет связано никаких тэгов.

Возвращаемое значение

Подтверждающее сообщение в случае успешного изменения тэгов.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30"
--data "tags=tag5/tag6" http://api.comexp.ru/video/
c4c95b02d19a7ff6395542f205de989b
<tags>OK</tags>
```

### Удаление видео



Описание	Метод удаляет видеофрагмент из системы
Структура запроса	REST <code>http://api.comexp.ru/video/&lt;video_id&gt;</code> 1. <code>video_id</code> – уникальный идентификатор (ID) видеофрагмента.
Метод	DELETE
Параметры	Только API-key
Возвращаемое значение	Подтверждающее сообщение в случае успешного удаления или сообщение об ошибке, если видеофрагмент не найден.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30"  
-X DELETE http://api.comexp.ru/video/  
01B9037D24FC816097ED77590985E5AC ↵  
<delete>OK</delete>
```

### Получение списка видеофрагментов

Описание	Метод предназначен для получения списка идентификаторов видеофрагментов по заданным тегам и в заданном интервале дат. Все условия (теги и интервал дат) жестко связаны условием «И».
Структура запроса	REST <code>http://api.comexp.ru/video/[tag1]/ [tag2]/.../? ts_from=ts1&amp;ts_to=ts2</code> 1. <code>tag1,tag2,...</code> - произвольное число тегов. 2. <code>ts_from</code> – начало интервала выборки фрагментов. 3. <code>ts_to</code> – конец интервала выборки фрагментов. Дата и время начала и конца интервала в формате Comexp API
Метод	PUT
Параметры	Только API-key
Возвращаемое значение	Список, с описанием полных свойств, видеофрагментов, которые удовлетворяют условиям в запросе. О свойствах видеофрагментов подробнее в описании функции “Получить свойства видеофрагмента”

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30"  
http://api.comexp.ru/video/tag5/ ↵  
<list>  
<video>...</video>  
<video>...</video>  
</list>
```

## **3.2.5 Работа с корзинами фрагментов**

### **Создание новой корзины**

Описание	Метод создает новую корзину видеофрагментов с заданным именем и набором тегов.
Структура запроса	REST <code>http://api.comexp.ru/bins/</code>
Метод	POST
Параметры	<ol style="list-style-type: none"><li>1. <code>name</code> – Имя корзины. Обязательный параметр.</li><li>2. <code>tags</code> – Список тэгов, по которым будут отбираться видеофрагменты в корзину. Тэги разделяются слэшем «/». Необязательный параметр. Если список тэгов пуст, то корзина считается пустой.</li><li>3. <code>ts_from</code> - начало временного интервала (необязательный параметр)</li><li>4. <code>ts_to</code> - конец временного интервала (необязательный параметр)</li></ol>
Возвращаемое значение	В случае успешного создания, метод возвращает уникальный идентификатор корзины в системе.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" --data "name=binname2&tags=tag2" http://api.comexp.ru/bins<bin>01B9037D24FC816097ED77590985E5AC</bin>
```

### Получение параметров корзины

Описание	Извлекает параметры корзины из системы.
Структура запроса	REST <code>http://api.comexp.ru/bins/&lt;bin_id&gt;</code> 1. <code>bin_id</code> – Идентификатор изменяемой корзины.
Метод	GET
Параметры	<ol style="list-style-type: none"><li>1. <code>name</code> – Имя корзины. Обязательный параметр.</li><li>2. <code>tags</code> – Список тэгов, по которым будут отбираться видеофрагменты в корзину. Тэги разделяются слэшем «/». Необязательный параметр. Если список тэгов пуст, то корзина считается пустой.</li><li>3. <code>ts_from</code> - начало временного интервала (необязательный параметр)</li><li>4. <code>ts_to</code> - конец временного интервала (необязательный параметр)</li></ol>
Возвращаемое значение	В случае успешного изменения, метод возвращает уникальный идентификатор корзины в системе.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" --data "name=binname2&tags=tag2" http://api.comexp.ru/bins/01B9037D24FC816097ED77590985E5AC<bin>01B9037D24FC816097ED77590985E5AC</bin>
```

### Изменение параметров корзины

Описание	Метод изменяет параметры корзины.
Структура запроса	REST <code>http://api.comexp.ru/bins/&lt;bin_id&gt;</code> 1. <code>bin_id</code> – Идентификатор изменяемой корзины.
Метод	POST
Параметры	1. <code>name</code> – Имя корзины. Обязательный параметр. 2. <code>tags</code> – Список тэгов, по которым будут отбираться видеофрагменты в корзину. Тэги разделяются слэшем «/». Необязательный параметр. Если список тэгов пуст, то корзина считается пустой. 3. <code>ts_from</code> - начало временного интервала (необязательный параметр). Если параметр не задан, то задается значение по умолчанию (“00000000000000”). 4. <code>ts_to</code> - конец временного интервала (необязательный параметр). Если параметр не задан, то задается значение по умолчанию (“00000000000000”).
Возвращаемое значение	В случае успешного изменения, метод возвращает уникальный идентификатор корзины в системе.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30"
--data "name=binname2&tags=tag2" http://api.comexp.ru/bins/
01B9037D24FC816097ED77590985E5AC
<bin>01B9037D24FC816097ED77590985E5AC</bin>
```

### Удаление корзины

Описание	Метод удаляет корзину из системы.
Структура запроса	REST <code>http://api.comexp.ru/bins/&lt;bin_id&gt;</code> 1. <code>bin_id</code> – Идентификатор удаляемой корзины.
Метод	DELETE
Параметры	Только API-key
Возвращаемое значение	Подтверждающее сообщение в случае успешного удаления.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" -X
DELETE http://api.comexp.ru/bins/4b867761e9016550070a6b301c492306
<delete>ОК</delete>
```

### Получение списка корзин

Описание	Метод предназначен для получения списка корзин.
Структура запроса	REST <code>http://api.comexp.ru/bins/</code>

Метод	GET
Параметры	Только API-key
Возвращаемое значение	<p>Список свойств всех созданных корзин. Подробнее о списках корзин смотрите в описании метода “Получение параметров корзины”</p> <p>Пример:</p> <pre>curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" http://api.comexp.ru/bins/ &lt;list&gt; &lt;bin&gt;...&lt;/bin&gt; &lt;bin&gt;...&lt;/bin&gt; ... &lt;bin&gt;...&lt;/bin&gt; &lt;/list&gt;</pre>

### **3.2.6 Работа с заданиями на поиск**

#### **Создание нового задания на поиск**

Описание	Метод создает новое задание на поиск. Видеофрагменты из второй корзины ищутся в первой корзине.
Структура запроса	REST <code>http://api.comexp.ru/search/</code>
Метод	POST
Параметры	<ol style="list-style-type: none"> <li>1. name – Имя задания на поиск. Необязательный параметр.</li> <li>2. bin1 – Идентификатор корзины в видеофрагментах которой осуществляется поиск. Обязательный параметр.</li> <li>3. bin2 – Идентификатор корзины видеофрагменты которой ищутся в первой корзине. Обязательный параметр.</li> </ol>
Возвращаемое значение	<p>В случае успешного создания, метод возвращает идентификатор задания на поиск.</p> <p>Пример:</p> <pre>curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30" -- data "bin1=e508e1904290391657d8d8a9392b62c0&amp;bin2=ffac89a79e1974c 4116d162c6b08dbc1" http://api.comexp.ru/search/ &lt;search&gt;aed45b98a4bd022195826482b8b0c6a2&lt;/search&gt;</pre>
Ошибки	<pre>&lt;errors&gt;&lt;error&gt;&lt;domain&gt;search&lt;/domain&gt;&lt;code&gt;bin1 2 not found&lt;/ code&gt;&lt;/error&gt;&lt;/errors&gt;</pre> <p>- Корзина с идентификатором bin1 2 не найдена.</p>

**Замечание**                    Вновь созданное задание автоматически запускается на выполнение в асинхронном режиме. Результаты можно узнать с помощью метода “Получение параметров и результатов поиска”  
Если, в следствии загрузки нового видеофрагмента или изменении параметров уже существующего, происходит обновление одно из корзин задания на поиск, то задания автоматически запускается повторно. Результаты поиска сохраняются, даже если видеофрагмент уже не попадает в одну из корзин задания на поиск по каким либо критериям.

**Получение параметров задания на поиск**

**Описание**                    Метод извлекает параметры задания на поиск и результаты поиска.

**Структура запроса**        REST    [http://api.comexp.ru/search/<search\\_id>](http://api.comexp.ru/search/<search_id>)  
1. search\_id - Идентификатор задания на поиск в системе.

**Метод**                        GET

**Параметры**                Только API-key

Возвращаемое значение

Параметры задания на поиск и результаты поиска.

```
<search>
<id>search_id</id>
<name>search_name</name>
<state>search_state</state>
<last_update>date_time</last_update>
<bin_id_1>bin_id1</bin_id_1>
<bin_id_2>bin_id2</bin_id_2>
<error>error_msg</error>
<data>
<match length="3">
<video id="vid1_from_bin1" start="89.880000"/>
<video id="vid1_from_bin2" start="0.000000"/>
</match>
...
</data>
</search>
```

#### Описание параметров задания и результатов поиска

id	Идентификатор задания на поиск
name	Имя задания на поиск
state	Состояние задания на поиск, которое может принимать следующие значения: 1 - Поиск запущен 2 - Поиск завершён. Результаты в блоке <data>. 3 - Поиск завершён с ошибкой. Информация об ошибке в блоке <error>
last_update	Дата и время в формате ComexpAPI последнего запуска задания.
bin_id_1	Идентификатор корзины в видеофрагментах которой осуществляется поиск.
bin_id_2	Идентификатор корзины видеофрагменты которой ищутся в первой корзине.
error	Элемент, который содержит строковое Описание ошибки.
data	Элемент, который содержит результаты поиска.
match	Элемент, который описывает результат нахождения

#### Изменение параметров задания на поиск

Описание	Метод изменяет параметры задания на поиск.
Структура запроса	REST <a href="http://api.comexp.ru/search/&lt;search_id&gt;">http://api.comexp.ru/search/&lt;search_id&gt;</a> 1. search_id - Идентификатор задания на поиск в системе.
Метод	POST

Параметры

1. name – Имя задания на поиск. Необязательный параметр.
2. bin1 – Идентификатор корзины в видеофрагментах которой осуществляется поиск. Обязательный параметр.
3. bin2 – Идентификатор корзины видеофрагменты которой ищутся в первой корзине. Обязательный параметр.

Возвращаемое значение

В случае успешного изменения, метод возвращает уникальный идентификатор задания на поиск в системе.

Пример:

```
<search>01B9037D24FC816097ED77590985E5AC</search>
```

#### **Удаления задания на поиск**

Описание

Метод удаляет задание на поиск из системы.

Структура запроса

REST [http://api.comexp.ru/search/<search\\_id>](http://api.comexp.ru/search/<search_id>)  
1. search\_id – Идентификатор удаляемого задания на поиск.

Метод

DELETE

Параметры

Только API-key

Возвращаемое значение

Подтверждающее сообщение в случае успешного удаления.

Пример:

```
<delete >OK</delete>
```

#### **Получение списка задания на поиск**

Описание

Метод предназначен для получения списка заданий на поиск.

Структура запроса

REST <http://api.comexp.ru/search/>

Метод

GET

Параметры

Только API-key

Возвращаемое значение

Список всех заданий на поиск в системе. Информация о заданиях выдается в краткой форме.

Пример:

```
curl --header "X-API-Key: 5A6BB49E098111DFABA000042388EF30"
```

```
http://api.comexp.ru/search/
```

```
<list>
```

```
<search>
```

```
<id>AED45B98A4BD022195826482B8B0C6A2</id>
```

```
<name></name>
```

```
<state>2</state>
```

```
<last_update>20111130111253</last_update>
```

```
<bin_id_1>E508E1904290391657D8D8A9392B62C0</bin_id_1>
```

```
<bin_id_2>6324FA1381062C626A2F2094CECDDA36</bin_id_2>
```

```
</search>
```

```
...
```

```
</list>
```

Информация о значениях в описании метода “Получение параметров и результатов поиска”