

Требования к API внешней ЭБС, для интеграции с модулем "Библиотека ресурсов ЗКЛ" (материал для разработчиков)

Этот материал предназначен для разработчиков ЭБС и библиотек контента, которые планируют интеграцию своих систем с [элементом курса «Библиотека ресурсов ЗКЛ»](#) в СЭО ЗКЛ. Такая интеграция позволит разработчикам электронных курсов в СЭО ЗКЛ бесшовно использовать материалы из данного внешнего источника без их прямого копирования в СЭО ЗКЛ.

По вопросам интеграции и сотрудничества [обращайтесь в ООО «Открытые технологии»](#).

В данной статье содержатся общие требования к API внешней ЭБС, необходимые для интеграции. Поскольку под каждую внешнюю ЭБС разработчики ООО «Открытые технологии» пишут отдельный драйвер, возможны вариации в реализации API (структуре и названиях запросов, названии переменных и пр.) Главное, чтобы API так или иначе реализовывало следующие возможности: запрос списка подкатегорий заданной категории (если ЭБС вообще поддерживает древовидную иерархию), поиск ресурса по поисковой строке (и, опционально, идентификатору категории), получение контента ресурса для встраивания в одном из трех вариантов: код для встраивания, транслируемый файл (обычно используется для картинок и дополнительных файлов), предавторизованная ссылка для просмотра ресурса конкретным пользователем.

Содержание:

1. [Используемые термины](#)
2. [Введение](#)
3. [Методы интеграции](#)
 - 3.1. [Авторизация](#)
 - 3.1.1. [Общий принцип](#)
 - 3.1.2. [Пример запроса с комментариями по возможностям реализации](#)
 - 3.2. [Получение категорий](#)
 - 3.2.1. [Общий принцип](#)
 - 3.2.2. [Пример запроса с комментариями по возможностям реализации](#)
 - 3.3. [Получение ресурсов по категории](#)
 - 3.3.1. [Пример запроса с комментариями по возможностям реализации](#)
 - 3.4. [Поиск ресурсов](#)
 - 3.4.1. [Пример запроса с комментариями по возможностям реализации](#)
 - 3.5. [Получение контента ресурса](#)
 - 3.5.1. [Доступные режимы отображения](#)
 - 3.5.2. [Пример запроса с комментариями по возможностям реализации](#)
4. [Дополнительная информация](#)

1. Используемые термины

Провайдер — организация, предоставляющая API для интеграции с СЭО ЗКЛ.

Ресурс — книга, статья, журнал, документ, предназначенный для отображения в результате интеграции в модуле «Библиотека ресурсов ЗКЛ».

Источник — экземпляр интеграции, реализующий комплекс возможностей, предоставляемых провайдером своему клиенту для доступа к ресурсам.

Категория — раздел, каталог, предоставляемый источником и содержащий в себе свойственные ему ресурсы.

[-Вернуться к содержанию-](#)

2. Введение

Плагин «Библиотека ресурсов ЗКЛ» системы дистанционного образования ЗКЛ (далее - СДО) предоставляет возможность преподавателю найти ресурс в одном из настроенных администратором среды электронного обучения источников, для последующего отображения ресурса учащимся. Преподаватель выполняет выбор ресурса путём выбора категории и просмотра списка ресурсов, содержащихся в категории, либо путём ввода поискового запроса и просмотра списка результатов поиска.

Ниже описаны методы, необходимые для реализации наиболее распространённого подхода интеграции между плагином «Библиотека ресурсов ЗКЛ» и внешними источниками. С их помощью возможно сформировать базовый набор функций, необходимых для работы с элементом курса.

Приведённые примеры носят рекомендательный характер, не являются обязательными требованиями и содержат перечень данных, необходимых по нашему опыту для реализации комфортной работы с интегрированной системой.

[-Вернуться к содержанию-](#)

3. Методы интеграции

3.1. Авторизация

Реализовывается провайдером при необходимости предоставлять ресурсы ограниченному списку пользователей.

[-Вернуться к содержанию-](#)

3.1.1. Общий принцип

- администратору СДО известны реквизиты интеграции (секретный ключ или пара логин/пароль), которые он настраивает для подключения к источнику;
- перед обращением к источнику, СДО делает запрос в источник по

405

- предоставленному источником API, передавая реквизиты интеграции;
- источник возвращает токен авторизации для использования в последующих запросах/

[-Вернуться к содержанию-](#)

3.1.2. Пример запроса с комментариями по возможностям реализации

Запрос:

ENDPOINT

- /api/auth

POST

- username - логин для авторизации
- password - пароль для авторизации

Ответ:

В ответ приходит объект JSON, содержащий результат обработки запроса и токен авторизации:

- (bool) success - результат обработки запроса
- (string) error - описание ошибки (если результат обработки запроса не был успешным)
- (string) token - токен авторизации для использования в последующих запросах

[-Вернуться к содержанию-](#)

3.2. Получение категорий

Преподаватель в СЭО ЗКЛ производит выбор ресурса, к которому планирует предоставить доступ учащемуся в курсе.

Одним из сценариев выбора ресурса является выбор из списка ресурсов, принадлежащих выбранной категории. Если источник имеет древовидную структуру хранения ресурсов, необходимо реализовать возможность получения списка категорий с её использованием.

[-Вернуться к содержанию-](#)

3.2.1. Общий принцип

- СДО обращается к источнику, запрашивая список категорий, являющихся дочерними для категории, переданной параметром;
- в запросе передается токен авторизации, полученный ранее от

405

- источника (если требуется источником);
- источник возвращает список категорий, дочерних от категории, переданной параметром.

[-Вернуться к содержанию-](#)

3.2.2. Пример запроса с комментариями по возможностям реализации

Запрос:

ENDPOINT

- /api/categories

HEADERS

- token - токен авторизации

GET

- idCategory - идентификатор родительской категории, чьих потомков необходимо получить. Значение null интерпретируется как запрос категорий верхнего уровня (без родителя)

Ответ:

В ответ приходит объект JSON, содержащий результат обработки запроса и список категорий, найденных по запросу:

- (bool) success - результат обработки запроса
- (string) error - описание ошибки (если результат обработки запроса не был успешным)
- (array) items - массив, содержащий объекты со сведениями о найденных категориях (пустой массив, если категорий не было найдено)

Объекты категорий должны содержать:

- идентификатор категории;
- наименование категории;
- сведения о наличии потомков в категории.

Объект категории может содержать любые дополнительные сведения о категории.

[-Вернуться к содержанию-](#)

3.3. Получение ресурсов по категории

405

Преподаватель в СДО производит выбор ресурса, к которому планирует предоставить доступ учащемуся в курсе.

Одним из сценариев выбора ресурса является выбор из списка ресурсов, принадлежащих выбранной категории. Если источник имеет структуру хранения ресурсов, подразумевающую наличие категорий, необходимо реализовать возможность получения списка ресурсов с её использованием.

[-Вернуться к содержанию-](#)

3.3.1. Пример запроса с комментариями по возможностям реализации

Запрос:

ENDPOINT

- /api/resources

HEADERS

- token - токен авторизации

GET

- idCategory - идентификатор категории
- limit - количество ресурсов необходимое для предоставления в результатах запроса;
- offset - количество ресурсов для смещения перед осуществлением выборки; вместо offset также допустимо принимать параметр page (страница) для организации смещения по формуле $((page-1) * limit)$

Ответ:

В ответ приходит объект JSON, содержащий результат обработки запроса и список ресурсов, найденных по запросу:

- (bool) success - результат обработки запроса
- (string) error - описание ошибки (если результат обработки запроса не был успешным)
- (array) items - массив, содержащий объекты со сведениями о найденных ресурсах (пустой массив, если ресурсов не было найдено)
- (int) total - общее количество результатов по запросу (до применения смещения offset и ограничения limit)

Объекты ресурсов должны содержать:

- уникальный идентификатор ресурса;
- заголовок ресурса.

Объекты ресурсов могут содержать любую дополнительную информацию о ресурсе. Наиболее распространенные сведения для примера:

- описание ресурса;
- авторы;
- isbn;
- издательство;
- количество страниц;
- год выпуска;
- url-адрес ресурса на веб-сайте провайдера;
- url-адрес с изображением обложки ресурса.

[-Вернуться к содержанию-](#)

3.4. Поиск ресурсов

Преподаватель в СДО производит выбор ресурса, к которому планирует предоставить доступ учащемуся в курсе.

Одним из сценариев выбора ресурса является выполнение поиска по поисковому запросу.

Если источник имеет возможность предоставить результаты поиска ресурсов по поисковому запросу, необходимо реализовать возможность получения списка ресурсов таким образом.

[-Вернуться к содержанию-](#)

3.4.1. Пример запроса с комментариями по возможностям реализации

Запрос:

<>ENDPOINT

- /api/resources

HEADERS

- token - токен авторизации

GET

- query - поисковая фраза. Провайдеру рекомендуется обрабатывать поисковую фразу не только для полнотекстового поиска по содержимому ресурса, но и для поиска по заголовку ресурса, isbn, автору, идентификатору ресурса; при отсутствии этого параметра запрос воспринимается как попытка получить все ресурсы из категории (раздел [«Получение ресурсов по категории»](#))
- idCategory - идентификатор категории, в пределах которой необходимо

405

выполнять поиск ресурсов по поисковой фразе. Если источник поддерживает древовидную структуру хранения ресурсов, мы рекомендуем реализовать поддержку этого параметра, при отсутствии параметра должен выполняться поиск по всему источнику

- limit - количество ресурсов необходимое для предоставления в результатах запроса
- offset - количество ресурсов для смещения перед осуществлением выборки. Вместо offset также допустимо принимать параметр page (страница) для организации смещения по формуле $((page-1) * limit)$

Ответ:

В ответ приходит объект JSON, содержащий результат обработки запроса и список ресурсов, найденных по запросу:

- (bool) success - результат обработки запроса
- (string) error - описание ошибки (если результат обработки запроса не был успешным)
- (array) items - массив, содержащий объекты со сведениями о найденных ресурсах (пустой массив, если ресурсов не было найдено).
- (int) total - общее количество результатов по запросу (до применения смещения offset и ограничения limit)

Объекты ресурсов должны содержать:

- уникальный идентификатор ресурса;
- заголовок ресурса.

Объекты ресурсов могут содержать любую дополнительную информацию о ресурсе. Наиболее распространенные сведения для примера:

- отрывок текста, в котором найдена поисковая фраза;
- описание ресурса;
- авторы;
- isbn;
- издательство;
- количество страниц;
- год выпуска;
- url-адрес ресурса на веб-сайте провайдера;
- url-адрес с изображением обложки ресурса.

[-Вернуться к содержанию-](#)

3.5. Получение контента ресурса

Преподаватель выбрал ресурс, к которому планирует предоставить доступ учащемуся в курсе.

Учащийся открывает элемент курса «Библиотека ресурсов ЗКЛ» с ресурсом, настроенным ранее преподавателем.

Система должна отобразить ресурс учащемуся.

[-Вернуться к содержанию-](#)

3.5.1. Доступные режимы отображения

- **Встроенный контент.** Элемент курса «Библиотека ресурсов ЗКЛ» отобразит учащемуся плеер. Формирование кода плеера производится либо по образцу от источника (в образец подставляются параметры ресурса), либо источник должен предоставить готовый html-код плеера по запросу. Получившийся код отображается в СДО как есть, соответственно необходимо обеспечить использование только полных ссылок в контенте.
- **Транслированный контент.** Элемент курса «Библиотека ресурсов ЗКЛ» отображает html-код ресурса, предоставленный источником, предварительно заменяя оригинальные ссылки, ведущие на материалы источника (не внешние) содержащиеся в контенте ресурса, на ссылку до специального скрипта, выполняющего маршрутизацию. Скрипт маршрутизации принимает в качестве параметра ссылку до источника и выполняет запрос к источнику на получение контента, передавая токен авторизации. Это позволяет организовать учащимся в СДО доступ к закрытому контенту без публикации ключей авторизации. Доступ предоставляется только пользователям, имеющим доступ к соответствующему элементу курса (авторизованы, подписаны на курс, дошли до изучения ресурса).
- **Ссылка со сквозной авторизацией.** При переходе учащегося к ресурсу, элемент курса «Библиотека ресурсов ЗКЛ» запрашивает у источника предавторизованную (содержащую токен авторизации) ссылку на ресурс по идентификатору ресурса. Минусом данного режима является переход на другой сайт и изменение внешнего вида системы.

[-Вернуться к содержанию-](#)

3.5.2. Пример запроса ресурса с комментариями по возможностям реализации

Запрос:

ENDPOINT

- /api/resource_content

HEADERS

- token - токен авторизации

GET

405

- id - идентификатор ресурса
- user - данные, идентифицирующие пользователя (для статистики)
- course - данные, идентифицирующие курс (для статистики)
- module - данные, идентифицирующие элемент курса (для статистики)
- session - данные, идентифицирующие сессию (для статистики)

Ответ:

В ответ приходит объект JSON, содержащий результат обработки запроса и данные о ресурсе:

- (bool) success - результат обработки запроса
- (string) error - описание ошибки (если результат обработки запроса не был успешным)
- (string) content - либо html-код плеера, либо html-код ресурса (контент), либо предавторизованная ссылка (в зависимости от необходимого режима отображения)

Ответ может также содержать другие данные, если выбран режим отображения, подразумевающий необходимость подставлять в образец кода плеера данные об отображаемом ресурсе.

[-Вернуться к содержанию-](#)

4. Дополнительная информация

Источник может предоставить дополнительные файлы, необходимые html-плееру, например, скрипты javascript или стили css.

Они могут быть переданы двумя способами:

- Источник может вернуть предавторизованные ссылки (могут содержать дополнительные ключи прямо в URL, не должны требовать авторизации), которые плагин «Библиотека ресурсов ЗКЛ» подставит в html-код.
- Источник может предоставить API для запроса этих файлов. Плагин «Библиотека ресурсов ЗКЛ» транслирует эти файлы. В этом случае, система проверит, что пользователь авторизован и имеет доступ к просмотру конкретного ресурса.

[-Вернуться к содержанию-](#)

Уникальный ID ответа: #1483

Опубликовал: : Алексей Дьяченко

Последние обновление: 2025-05-26 11:14