**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ КОСТРОМСКОЙ ОБЛАСТИ**

**ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «КОСТРОМСКОЙ ЭНЕРГЕТИЧЕСКИЙ ТЕХНИКУМ ИМ. Ф.В. ЧИЖОВА»**

Химико-технологическое отделение

Специальность 09.02.07 «Информационные системы и программирование»

МДК 05.02. Разработка кода информационных систем

**КУРСОВАЯ РАБОТА**

**Тема: «Разработка кода информационной системы для работы Видеосалона»**

**Выполнил**:

Студент 3 курса, группы 3-1ИС

Савельев Константин Олегович

**Проверил**: Ищук Денис Николаевич

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2024

Оглавление

[Глоссарий 3](#_Toc168251163)

[Введение 4](#_Toc168251164)

[Разработка системы 4](#_Toc168251165)

[Список функционала 4](#_Toc168251166)

[Реализация системы 5](#_Toc168251167)

[Архитектура веб приложения 5](#_Toc168251168)

[Протокол HTTP 5](#_Toc168251169)

[HTTP и RESTful веб-сервисы 6](#_Toc168251170)

[Ресурс 6](#_Toc168251171)

[REST и Ресурсы 6](#_Toc168251172)

[Компоненты HTTP 7](#_Toc168251173)

[Документация API 7](#_Toc168251174)

[Проверка текущей сессии 7](#_Toc168251175)

[Авторизация на сайте 7](#_Toc168251176)

[Регистрация пользователя 7](#_Toc168251177)

[Действия с фильмами 7](#_Toc168251178)

[Источники информации 8](#_Toc168251179)

[Ссылки 8](#_Toc168251180)

# Глоссарий

HTML — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

CSS — это формальный язык, который используется для описания и оформления внешнего вида веб-страниц.

JS — Мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

JQuery — Бесплатная JavaScript-библиотека с открытым исходным кодом. Она заметно упрощает веб-разработку, позволяя писать меньше кода, чем на ванильном JavaScript.

AJAX — «асинхронный JavaScript и XML» — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.

PHP — Интерпретируемый скриптовый язык программирования общего назначения.

API — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.

# Введение

Изначально видеосалоны были созданы для просмотра фильмов на большом экране, но с появлением видеомагнитофонов они превратились в домашние кинозалы. В этих салонах люди могли арендовать или купить видеокассеты с фильмами, а затем смотреть их у себя дома.

Видеосалоны стали популярными среди молодых людей, которые хотели посмотреть новые фильмы без необходимости идти в кинотеатр. Они также были популярны среди тех, кто не мог позволить себе купить домашний видеомагнитофон или кассеты с фильмами.

Однако к середине 1990–х годов видеосалоны начали исчезать из–за появления кабельных и спутниковых телевизионных услуг, которые предлагали широкий выбор фильмов и программ. Кроме того, интернет и онлайн–кинотеатры также стали популярными, что еще больше уменьшило популярность видеосалонов.

Сегодня видеосалоны все еще существуют, но они стали редкими и часто находятся в небольших городах или сельских районах. Многие из них превратились в интернет–кафе или клубы, где люди могут смотреть фильмы

Разработка информационной системы позволит оптимизировать работу видеосалона, сократить временные и финансовые затраты, повысить удовлетворенность клиентов и улучшить качество предоставляемых услуг.

Целью данного проекта является разработка информационной системы для видеосалона, которая будет автоматизировать основные бизнес–процессы и повысить эффективность работы учреждения.

Архитектура системы реализована по Rest API. Для реализации приложения использовались: PHP для серверной части, JavaScript и библиотека JQuery для клиентской части приложения.

## Анализ предметной области

Для анализа своей предметной области я буду использовать метод анализ данных, так как нам нужно идентификация и описание данных, необходимых для функционирования системы.

Для своей предметной области были исследованы следующие сайты:

1. Бургерная–видеосалон [Гамбургеры | chef.ru](https://chef.ru/place/gamburgery/?ysclid=ls1gb64nck297366140)

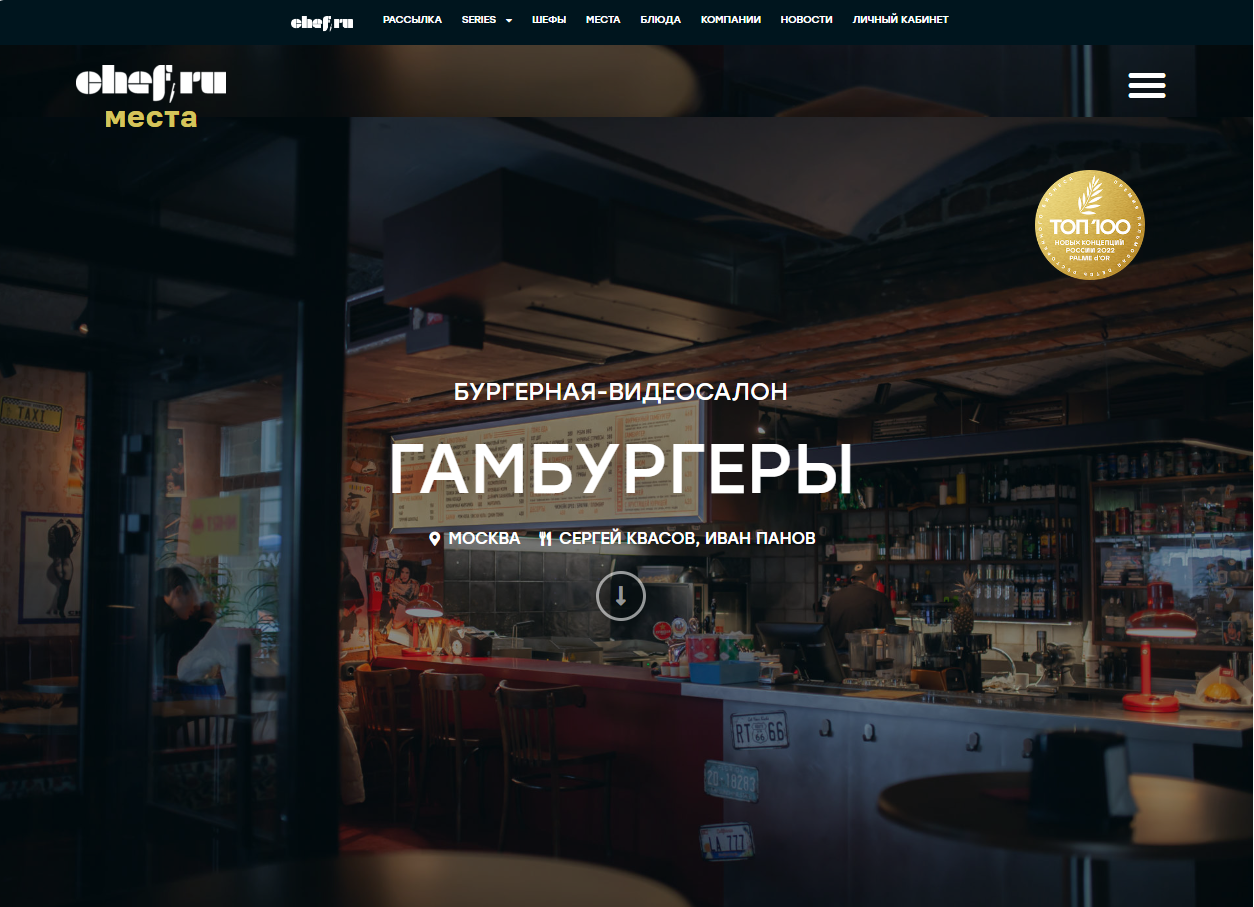


Рис. 1 страница сайта для анализа 1

Достоинства этого видеосалона:

* Сайт встречает приятным и удобным пользовательским интерфейсом.  С первого раза пользователь поймет, что и где расположено.
* Качественные фотографии товаров.
* Красивые карточки товаров.
* Есть регистрация и личный кабинет.
* Сайт адаптирован для телефонов.

Недостатки этого видеосалона:

Некоторые кнопки не подсвечиваются и не понятно можно ли нажать на неё или нет.

1. PSN coll game [PSNCOOLGAME | Цифровой прокат игр для PS5–PS4–VR | Каталог товара](https://psncoolgame.com/?ysclid=ls1gvj4w8g407144618)

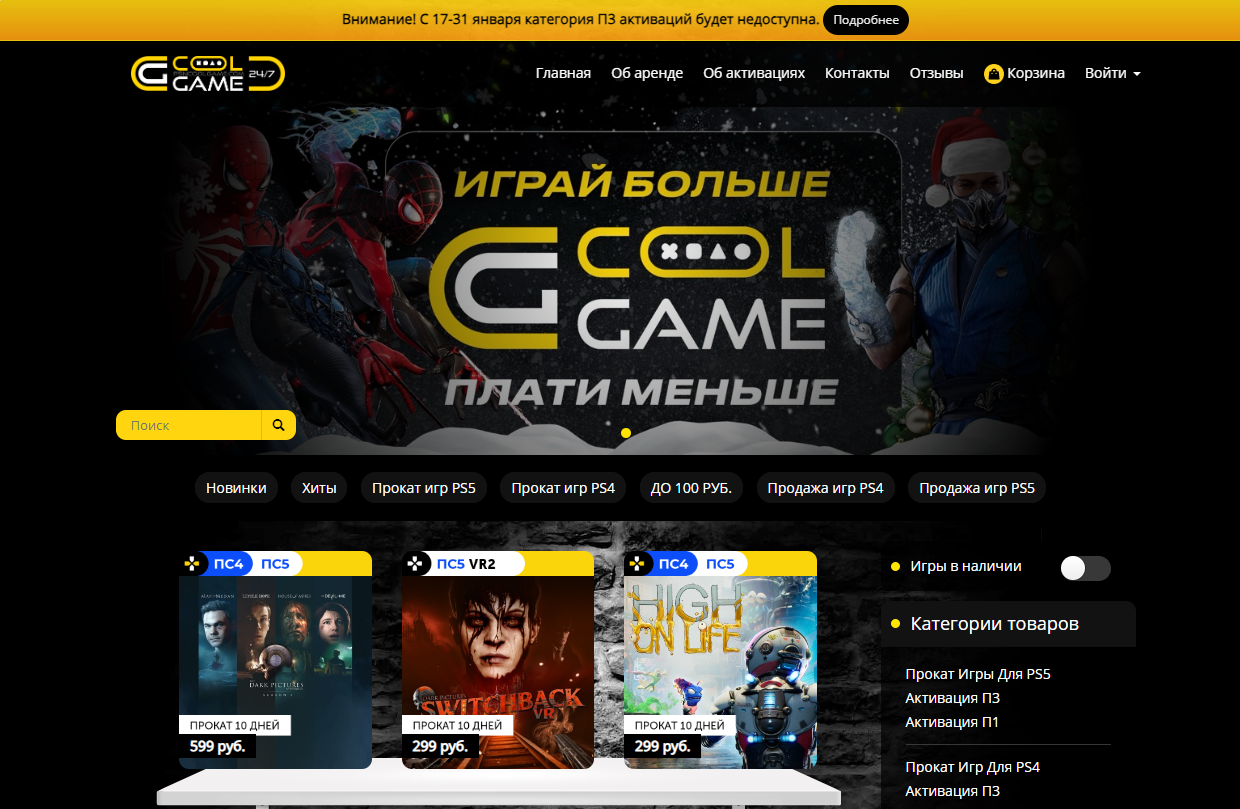


Рис. 2 страница сайта для анализа 2

Достоинства этого видеосалона:

* Сайт встречает приятным и удобным пользовательским интерфейсом.  С первого раза пользователь поймет, что и где расположено.
* Качественные фотографии товаров.
* Красивые карточки товаров.
* Есть регистрация и личный кабинет.
* Сайт адаптирован для телефонов.

Недостатки этого видеосалона:

* много нивидимых гиперссылок

Предметная область видеосалона охватывает услуги по прокату видеофильмов и компьютерных игр. Пользователи могут арендовать фильмы на определенный срок, после чего возвращать их в видеосалон

Анализ предметной области видеосалона может включать следующие аспекты:

* Клиенты: видеосалоны обычно обслуживают широкую аудиторию клиентов, включая людей разных возрастных групп и интересов. Важно понять предпочтения и потребности клиентов, чтобы предложить им наиболее интересные и актуальные фильмы.
* Каталог: видеосалон должен иметь обширный каталог фильмов, включая различные жанры и форматы. Важно постоянно обновлять каталог новыми релизами и предлагать классические фильмы.
* Условия проката: видеосалон может предлагать различные условия проката, такие как аренда на определенное количество дней, возможность продления аренды или возможность получить скидку при аренде нескольких фильмов. Важно установить разумные и привлекательные цены и условия проката.
* Техническое оборудование: видеосалон должен обладать соответствующим техническим оборудованием, чтобы клиенты могли просматривать фильмы. Это включает DVD–плееры.
* Конкурентная среда: важно изучить конкурентов в данной сфере, чтобы определить свои преимущества и выделиться на рынке. Это может включать анализ цен, условий проката, качества обслуживания и доступности определенных фильмов.
* Маркетинг и продвижение: видеосалон может использовать различные маркетинговые стратегии для привлечения клиентов, такие как реклама в местных СМИ, социальные сети, программы лояльности или сотрудничество с другими компаниями. Важно определить наиболее эффективные способы привлечения и удержания клиентов.

# Разработка системы

## Список функционала

Функционал доступный неавторизированному пользователю:

* Авторизация на сайте
* Регистрация на сайте

Функционал доступный авторизированному пользователю:

* Просмотр каталога фильмов доступных на сайте
* Просмотр акций проходящих на данный момент
* Покупка фильма
* Аренда фильма
* Просмотр списка купленных или взятых в аренду фильмов
* Выход из аккаунта

Функционал доступный администратору:

* Добавление нового фильма
* Редактирование уже существующего фильма
* Удаление фильма

## Реализация системы

### Архитектура веб приложения

REST расшифровывается как REpresentational State Transfer. Это был термин, первоначально введен Роем Филдингом (Roy Fielding), который также был одним из создателей протокола HTTP. Отличительной особенностью сервисов REST является то, что они позволяют наилучшим образом использовать протокол HTTP.

Когда мы просматриваем страницы в Интернете, за кулисами происходит много вещей. Ниже приведено упрощенное представление о том, что происходит между браузером и серверами, работающими на посещаемых веб-сайтах:



### Протокол HTTP

Когда вы вводите в браузере URL-адрес, например [www.google.com](https://www.google.com/), на сервер отправляется запрос на веб-сайт, идентифицированный URL-адресом.  
Затем этот сервер формирует и выдает ответ. Важным является формат этих запросов и ответов. Эти форматы определяются протоколом HTTP — Hyper Text Transfer Protocol.

Когда вы набираете URL в браузере, он отправляет запрос GET на указанный сервер. Затем сервер отвечает HTTP-ответом, который содержит данные в формате HTML — Hyper Text Markup Language. Затем браузер получает этот HTML-код и отображает его на экране.

Допустим, вы заполняете форму, присутствующую на веб-странице, со списком элементов. В таком случае, когда вы нажимаете кнопку «Submit» (Отправить), HTTP-запрос POST отправляется на сервер.

### HTTP и RESTful веб-сервисы

HTTP обеспечивает базовый уровень для создания веб-сервисов. Поэтому важно понимать HTTP. Вот несколько ключевых абстракций.

### Ресурс

Ресурс — это ключевая абстракция, на которой концентрируется протокол HTTP. Ресурс — это все, что вы хотите показать внешнему миру через ваше приложение. Например, если мы пишем приложение для управления задачами, экземпляры ресурсов будут следующие:

* Конкретный пользователь
* Конкретная задача
* Список задач

#### URI ресурса

Когда вы разрабатываете RESTful сервисы, вы должны сосредоточить свое внимание на ресурсах приложения. Способ, которым мы идентифицируем ресурс для предоставления, состоит в том, чтобы назначить ему URI — универсальный идентификатор ресурса.

Например:

* Создать пользователя: POST /users
* Удалить пользователя: DELETE /users/1
* Получить всех пользователей: GET /users
* Получить одного пользователя: GET /users/1

### REST и Ресурсы

Важно отметить, что с REST вам нужно думать о приложении с точки зрения ресурсов:

* Определите, какие ресурсы вы хотите открыть для внешнего мира
* Используйте глаголы, уже определенные протоколом HTTP, для выполнения операций с этими ресурсами.

Вот как обычно реализуется служба REST:

* Формат обмена данными: здесь нет никаких ограничений. JSON — очень популярный формат, хотя можно использовать и другие, такие как XML
* Транспорт: всегда HTTP. REST полностью построен на основе HTTP.
* Определение сервиса: не существует стандарта для этого, а REST является гибким. Это может быть недостатком в некоторых сценариях, поскольку потребляющему приложению может быть необходимо понимать форматы запросов и ответов. Однако широко используются такие языки определения веб-приложений, как WADL (Web Application Definition Language) и Swagger.

REST фокусируется на ресурсах и на том, насколько эффективно вы выполняете операции с ними, используя HTTP.

### Компоненты HTTP

HTTP определяет следующую структуру запроса:

* строка запроса (request line) — определяет тип сообщения
* заголовки запроса (header fields) — характеризуют тело сообщения, параметры передачи и прочие сведения
* тело сообщения (body) — необязательное

HTTP определяет следующую структуру ответного сообщения (response):

* строка состояния (status line), включающая код состояния и сообщение о причине
* поля заголовка ответа (header fields)
* дополнительное тело сообщения (body)

#### Методы HTTP-запроса

Метод, используемый в HTTP-запросе, указывает, какое действие вы хотите выполнить с этим запросом. Важные примеры:

* GET: получить подробную информацию о ресурсе
* POST: создать новый ресурс
* PUT: обновить существующий ресурс
* DELETE: Удалить ресурс

#### Код статуса ответа HTTP

Код состояния всегда присутствует в ответе HTTP. Типичные примеры:

* 200 — успех
* 404 — cтраница не найдена

## Документация API

### Проверка текущей сессии

Текущей сессии нет

|  |  |
| --- | --- |
| request | response |
| url:’ http://videosalon/api/session/CurrentUser.php’  dataType: ‘json’ | Header:’ Access-Control-Allow-Origin: \*’  Header: ‘Content-Type: application/json’  Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  status: ’unauthorised’  }  ] |
| GET | 200 OK |

Есть активная сессия

|  |  |
| --- | --- |
| request | response |
| url:’ http://videosalon/api/session/CurrentUser.php’  dataType: ‘json’ | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Id:’2’  Name:’Константин’  Surname: ‘Савельев’  Email: ‘hgtcpvshc@gmail.com’  Password: 1234  Role: ‘admin’  }  ] |
| GET | 200 OK |

### Авторизация на сайте

Успешная авторизация

|  |  |
| --- | --- |
| request | response |
| url:’ http://videosalon/api/session/login.php’  dataType: ‘json’  [  {  email: ’hgtcpvshc@gmail.com’,  password: ’1234’  }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Massage: ’success’  }  ] |
| POST | Status: 201 Accepted |

Неверно введён логин или пароль

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/session/login.php  ’  dataType: ‘json’  [  {  email: ’hgtcpvshc@gmail.com’,  password: ’123’  }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Massage: ’Ошибка неверный логин или пароль’  }  ] |
| POST | Status: 503 |

### Регистрация пользователя

Успешная регистрация

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/session/register.php  ’  dataType: ‘json’  [  {  name: ’Константин’,  surname:’Савельев’,  email: ’hgtcpvshc@gmail.com’,  password: ’1234’,  confirmpassword: ‘1234’  }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Massage: ’Пользователь добавлен’  }  ] |
| POST | Status: 201 Accepted |

Не заполнены поля

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/session/register.php  ’  dataType: ‘json’  [  {  name: ’’,  surname:’Савельев’,  email: ’hgtcpvshc@gmail.com’,  password: ’1234’,  confirmpassword: ‘1234’  }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Massage:’не все поля заполены’  }  ] |
| POST | Status: 400 Bad request |

Пользователь уже существует

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/session/register.php  ’  dataType: ‘json’  [  {  name: ’Константин’,  surname:’Савельев’,  email: ’hgtcpvshc@gmail.com’,  password: ’1234’,  confirmpassword: ‘1234’  }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Massage:пользователь уже существует’  }  ] |
| POST | Status: 400 Bad request |

Пароли не совпадают

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/session/register.php  ’  dataType: ‘json’  [  {  name: ’Константин’,  surname:’Савельев’,  email: ’hgtcpvshc@gmail.com’,  password: ’1234’,  confirmpassword: ‘12343’  }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Massage:пароли не совпадают’  }  ] |
| POST | Status: 400 Bad request |

### Действия с фильмами

Список всех фильмов

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/readFilms.php  ’  dataType: ‘json’ | [      {          "id": "1",          "title": "Криминальное чтиво",          "poster": "Pulp\_Fiction.jpg",          "producer": "Квентин тарантино",          "description": "тут описание",          "rentprice": "100",          "buyprice": "600"      }  ] |
| GET | 200 OK |

Добавление фильма

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/CreateFilms.php  ’  dataType: ‘json’  [      {          "title": "Криминальное чтиво",          "poster": "Pulp\_Fiction.jpg",          "producer": "Квентин тарантино",          "description": "тут описание",          "rentprice": "100",          "buyprice": "600"      }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Messege: фильм добавлен  }  ] |
| POST | 201 Accepted |

Добавление фильма неудачно

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/CreateFilms.php  ’  dataType: ‘json’  [      {          "title": "Криминальное чтиво",          "poster": "Pulp\_Fiction.jpg",          "producer": "Квентин тарантино",          "description": "тут описание",          "rentprice": "100",          "buyprice": "600"      }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Messege: Невозможно добавить фильм  }  ] |
| POST | 400 Bad request |

Удаление фильма

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/DeleteFilm.php  ’  dataType: ‘json’  [      {          ‘id‘:1      }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Messege: фильм удалён  }  ] |
| DELETE | 201 Accepted |

Редактирование фильма

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/UpdateFilm.php  ’  dataType: ‘json’  [      {          "title": "Криминальное чтиво",          "poster": "Pulp\_Fiction.jpg",          "producer": "Квентин тарантино",          "description": "описание изменено",          "rentprice": "100",          "buyprice": "600"      }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Messege: фильм обновлён  }  ] |
| PUT | 201 Accepted |

Аренда фильма

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/RentFilm.php  ’  dataType: ‘json’  [      {          'product':1,  ‘user’:1,  ‘term’:6,  ‘price’:600      }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Messege: фильм арендован  }  ] |
| POST | 201 Accepted |

Покупка фильма

|  |  |
| --- | --- |
| request | response |
| url: ‘http://videosalon/api/entity/RentFilm.php  ’  dataType: ‘json’  [      {          'product':1,  ‘user’:1,  ‘price’:600      }  ] | Header: "Content-Type: application/json; charset=UTF-8"  Header: "Access-Control-Max-Age: 3600"  Header: "Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With"  [  {  Messege: фильм куплен  }  ] |
| POST | 201 Accepted |

# Источники информации

# Ссылки

*AJAX: Шпаргалка по AJAX на jQuery*. (б.д.). Получено из https://habr.com/ru/articles/658139/

*Frontend*. (б.д.). Получено из https://only-to-top.ru/blog/programming/2019-11-11-jquery-ajax-json-php.html

*REST API — RESTful веб-сервисы*. (б.д.). Получено из https://habr.com/en/articles/483202/

*Rest API PHP пошаговое руководство*. (б.д.). Получено из https://only-to-top.ru/blog/programming/2019-11-06-rest-api-php.html

*Документация JQuery*. (б.д.). Получено из https://api.jquery.com/

*Документация PHP*. (б.д.). Получено из https://www.php.net/manual/en/