|  |
| --- |
| API КиноПортала |
| Спецификация |
|  |
| |  |  |  |  | | --- | --- | --- | --- | | Дата | Версия | Автор | Изменения | | 07.04.2011 | 1.0 | i.metaljnikov@ftl-consult.ru | Спецификация приведена в соответствие с особенностями реализации | | 07.04.2011 | 1.1 | i.metaljnikov@ftl-consult.ru | Добавлены методы для получения станций метро | | 14.04.2011 | 1.2 | i.metaljnikov@ftl-consult.ru | Добавлено описание процедуры получения URL-ов изображений | | 14.04.2011 | 1.21 | i.metaljnikov@ftl-consult.ru | Добавлен формат sinceNow в методы получения расписаний, небольшие исправления | | 05.05.2011 | 1.22 | i.metaljnikov@ftl-consult.ru | Значение поля priceCategory изменено с id ценовой категории на её текстовое название | | 05.05.2011 | 1.3 | i.metaljnikov@ftl-consult.ru | Добавлена структура paymentInfo в ReservationResponse | | 25.05.2011 | 1.4 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлено поле distance в CinemaInfo(Short)  Добавлены время сеанса и постер в ReservationInfo | | 06.06.2011 | 1.5 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлен метод networks для киносетей. Добавлена фильтрация кинотеатров по киносетям | | 07.06.2011 | 1.6 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены трейлеры | | 08.06.2011 | 1.7 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Партнерозависимые методы (регистрация и аутентификация) перенесены в соответствующие Приложения. | | 15.06.2011 | 1.72 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены: код города, превью трейлеров. | | 16.06.2011 | 1.73 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены координаты в города. | | 17.06.2011 | 1.8 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Изменен формат выдачи трейлеров | | 05.07.2011 | 1.9 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлена поддержка платёжных средств.  Добавлены сценарии поведения клиента в случае различных статусов заказа.  Добавлена ошибка исчерпания лимитов (для метода reservation) | | 06.07.2011 | 1.91 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Убрана поддержка платёжных средств. | | 07.07.2011 | 1.95 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Изменен формат выдачи трейлеров | | 21.07.2011 | 1.96 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены поля в платёжную форму | | 27.07.2011 | 1.97 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлена фильтрация списков кинотеатров и расписаний по метро | | 28.07.2011 | 1.971 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлен фильтр isSale для кинотеатров | | 29.07.2011 | 1.98 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлен метод /cinemas/geo | | 15.08.2011 | 1.99 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены адрес и координаты кинотеатра в ответ на резервирование | | 17.01.2012 | 1.991 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены is3d и isImax в ScheduleInfoShort | | 31.01.2012 | 1.993 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Приложение M1: параметр app более не является обязательным и будет удален в следующих версиях протокола, добавлен параметр apikey | | 06.04.2012 | 1.995 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены методы получения истории заказов и работы с избранными кинотеатрами | |

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Автор | Изменения |
| 09.04.2012 | 2.0 | i.metaljnikov@ftl-consult.ru | Переписан метод /reservation для отказа от использования API ПА |
| 09.04.2012 | 2.001 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Приложения переименованы и добавлены в основной документ, убран параметр app в методе /registration |
| 09.04.2012 | 2.01 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлена передача метода платежа в /reservation и метод для получения методов платежа /payment-types  Исправлены опечатки |
| 12.04.2012 | 2.02 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Убран метод /payment-types (соответствующий функционал перенесён в /reservation). Добавлен метод генерации платёжных ссылок |
| 16.04.2012 | 2.021 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | ДобавленыполяminSum, maxSum, enabledв PaymentType  Добавлено поле fee в ReservationInfo |
| 16.04.2012 | 2.022 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлен параметр filter в метод /movies |
| 17.07.2012 | 2.023 | v.merkushov@kinohod.ru | Добавлен параметр isOrigin в структуру ScheduleInfo(Short) |
| 10.09.2012 | 2.03 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлена информация об оригинале видеоролика в структуру MovieInfo |
| 10.10.2012 | 2.031 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлены qrcodeURL и pkpassURL в структуру Order |
| 07.11.2012 | 2.032 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | ДобавленыqrcodeURLиpkpassURLвструктуруReservationInfo, убран ticketUrl |
| 27.11.2012 | 2.04 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлена сортировка /movies по кассовым сборам в России и дате премьеры в России (revenue и premiere)  Добавлен фильтр по жанрам в /movies(genre) |
| 14.12.2012 | 2.05 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлены параметры ключевой аттрактор attractor, дата премьеры в России premiereDateRussia, кассовые сборы в России grossRevenueRus, кассовые сборы в мире grossRevenueWorld, статус продаж билетов is\_running, `Есть ли фильм в «хочу пойти»` is\_wanted, доступные форматы для фильма formats в структуры MovieInfo / MovieInfoShort.  Добавлен метод вывода жанров, принадлежащим фильмам /genres  Добавлен вариант `wanted` в filter метода /movies  Добавлен модификатор formats для расписания сеансов  Добавлен список метро subway\_station в структуру CinemaInfo / CinemaInfoShort.  Список Добавлен Массив string goodies (in\_shop, in\_cafe, etc.) в структуру CinemaInfo  Добавлен телефон кинотеатр аphones в CinemaInfoShort для cinemas и showtimes.  Добавил isSale и isFav в CinemaInfoShort дляcinemas  Добавлен новый метод доступных форматов для фильма /movie/{id}/formats  Добавлен метод /nearest |
| 17.12.2012 | 2.051 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлено поле shortTitle в СinemaInfoShort |
| 19.12.2012 | 2.052 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | Добавлена расшифровка поля goodies структуры CinemaInfo |
| 20.12.2012 | 2.053 | [i.metaljnikov@ftl-consult.ru](mailto:i.metaljnikov@ftl-consult.ru) | Добавлен цвет линии в структуру SubwayStation |
| 15.01.2013 | 2.054 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | Добавлена структура genres |
| 16.01.2013 | 2.055 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | Изменен ключ текстового описания ошибки в структуре ErrorDesc |
| 17.01.2013 | 2.056 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | В CinemaInfoShort добавлены структура Phones и поле isFav. В метод nearest добавлено поле distance |
| 21.01.2013 | 2.06 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | Добавлен метод Loyalties |
| 22.01.2013 | 3.01 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | Изменен формат вывода жанров с применением структуры genres. |
| 22.01.2013 | 3.02 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Изменен формат рейтинга фильма. |
| 25.01.2013 | 3.03 | [v.finko@kinohod.ru](mailto:v.finko@kinohod.ru) | В метод reserve добавлен учет программы лояльности. В структуру moviesInfoShort добавлено поле firstSeanceStime. |
| 30.01.2013 | 3.04 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлен расчет начисленных единиц валюты по доступным программам лояльности для метода /reservation (поле loyalties в ReservationResponse).  Добавлен метод для проверки валидности номеров карт программ лояльности /loyalty/{id}/card\_check |
| 31.01.2013 | 3.05 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Удален учет лояльности из метода /reservation  Добавлен метод для связки заказа с программой лояльности /orders/<id>/loyalty/<id> |
| 01.02.2013 | 3.06 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлен фильтр “running” для списка жанров /genres  В структуру moviesInfoShort добавлено поле countScreens |
| 05.02.2013 | 3.07 | n.anisimova@kinohod.ru | Добавлено поле subway\_station\_distance в структуру SubwayStation запроса /cinemas. Добавлена структура SubwayStation в запрос /nearest |
| 06.02.2013 | 3.08 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлен алиас alias в структуру LoyaltiesInfo.  Добавлен вывод лояльностей loyalties, закрепленных за заказом в /orders |
| 11.02.2013 | 3.085 | i.metalnikov@kinohod.ru | Добавлено поле fragment в структуру SeatInfo для группировки мест в диваны |
| 14.02.2013 | 3.09 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлен метод DELETE для отмены заказа |
| 19.04.2013 | 4.00 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлено описание авторизации моб. прилож. на сайте «Киноход»  Добавлен метод привязки уч.записи /user/bind  Добавлены методы /user/order и /user/orders |
| 22.04.2013 | 4.01 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Реализован метод отвязки моб. сессии с уч.записью сайта «Киноход» /usr/unbind  Добавлен метод /user/profile  Добавлен метод добавления/изменения карт лояльности /user/profile/loyalty  Переработаны методы **/movies**/{id}/wanted, cinemas/{id}/fav, /orders, /reservation для работы с уч.записью на сайте |
| 30.04.2013 | 4.02 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлены поля subtitle, language в структуры ScheduleInfo, ScheduleInfoShort Добавлены методы /user/cinemas/fav/add, /user/cinemas/fav/delete, /user/movies/wanted/add, /user/movies/wanted/delete |
| 30.07.2013 | 4.03 | n.anisimova@kinohod.ru | Добавлены методы /cancel и /cancel\_order для отмены заказа клиентом |
| 06.08.2013 | 4.04 | i.metalnikov@kinohod.ru | Переработан рейтинг фильма, добавлена структура Rating  Добавлен идентификатор фильма в IMDB |
| 15.10.2013 | 4.05 | [v.elfimov@kinohod.ru](mailto:v.elfimov@kinohod.ru) | Добавлены ссылки на ICS-, VCS-файлы в структуру заказа |

Оглавление

[Термины и обозначения 12](#_Toc367096689)

[Общее описание протокола 13](#_Toc367096690)

[Введение 13](#_Toc367096691)

[Сервер 13](#_Toc367096692)

[Требования к протоколу 13](#_Toc367096693)

[Безопасность 13](#_Toc367096694)

[Версионность 13](#_Toc367096695)

[Сущности протокола 13](#_Toc367096696)

[Ресурсы 13](#_Toc367096697)

[Модификаторы доступа 14](#_Toc367096698)

[Операции 14](#_Toc367096699)

[Неподдерживаемые операции 14](#_Toc367096700)

[Представление данных 15](#_Toc367096701)

[Кэширование 15](#_Toc367096702)

[Использование Cookies 15](#_Toc367096703)

[Ответы сервера 15](#_Toc367096704)

[Безопасность 16](#_Toc367096705)

[Соединение 16](#_Toc367096706)

[Аутентификация 16](#_Toc367096707)

[Фильмы 18](#_Toc367096708)

[URI 18](#_Toc367096709)

[Методы 18](#_Toc367096710)

[Модификаторы URI 18](#_Toc367096711)

[Структуры данных 19](#_Toc367096712)

[Поведение клиента 21](#_Toc367096713)

[Поведение сервера 21](#_Toc367096714)

[Жанры фильмов 21](#_Toc367096715)

[URI 21](#_Toc367096716)

[Методы 21](#_Toc367096717)

[Модификаторы URI 22](#_Toc367096718)

[Поведение клиента 22](#_Toc367096719)

[Поведение сервера 22](#_Toc367096720)

[Форматы фильмов 22](#_Toc367096721)

[URI 22](#_Toc367096722)

[Методы 22](#_Toc367096723)

[Модификаторы URI 22](#_Toc367096724)

[Поведение клиента 23](#_Toc367096725)

[Поведение сервера 23](#_Toc367096726)

[Комментарии и оценки к фильму 23](#_Toc367096727)

[URI 23](#_Toc367096728)

[Методы 23](#_Toc367096729)

[Модификаторы URI 24](#_Toc367096730)

[Структуры данных 24](#_Toc367096731)

[Поведение клиента 24](#_Toc367096732)

[Поведение сервера 24](#_Toc367096733)

[Добавление фильма в “хочу пойти” 25](#_Toc367096734)

[URI 25](#_Toc367096735)

[Методы 25](#_Toc367096736)

[Структуры данных 25](#_Toc367096737)

[Поведение клиента 25](#_Toc367096738)

[Поведение сервера 25](#_Toc367096739)

[Удаление фильма из “хочу пойти” 25](#_Toc367096740)

[URI 26](#_Toc367096741)

[Методы 26](#_Toc367096742)

[Структуры данных 26](#_Toc367096743)

[Поведение клиента 26](#_Toc367096744)

[Поведение сервера 26](#_Toc367096745)

[Фильмы, идущие в кинотеатрах в ближайший час 26](#_Toc367096746)

[URI 26](#_Toc367096747)

[Методы 26](#_Toc367096748)

[Модификаторы URI 27](#_Toc367096749)

[Структуры данных 27](#_Toc367096750)

[Поведение клиента 28](#_Toc367096751)

[Поведение сервера 28](#_Toc367096752)

[Кинотеатры 28](#_Toc367096753)

[URI 28](#_Toc367096754)

[Методы 28](#_Toc367096755)

[Модификаторы URI 28](#_Toc367096756)

[Структуры данных 29](#_Toc367096757)

[Поведение клиента 30](#_Toc367096758)

[Поведение сервера 31](#_Toc367096759)

[Расписание сеансов 31](#_Toc367096760)

[URI 31](#_Toc367096761)

[Методы 31](#_Toc367096762)

[Модификаторы URI 31](#_Toc367096763)

[Структуры данных 32](#_Toc367096764)

[Поведение клиента 33](#_Toc367096765)

[Поведение сервера 33](#_Toc367096766)

[Информация о сеансе 33](#_Toc367096767)

[URI 33](#_Toc367096768)

[Методы 33](#_Toc367096769)

[Модификаторы URI 34](#_Toc367096770)

[Структуры данных 34](#_Toc367096771)

[Поведение клиента 35](#_Toc367096772)

[Поведение сервера 35](#_Toc367096773)

[Информация о зале 35](#_Toc367096774)

[URI 35](#_Toc367096775)

[Методы 35](#_Toc367096776)

[Модификаторы URI 35](#_Toc367096777)

[Структуры данных 35](#_Toc367096778)

[Поведение клиента 36](#_Toc367096779)

[Поведение сервера 37](#_Toc367096780)

[Резервирование мест и оплата 38](#_Toc367096781)

[URI 38](#_Toc367096782)

[Методы 38](#_Toc367096783)

[Модификаторы URI 38](#_Toc367096784)

[Структуры данных 38](#_Toc367096785)

[Поведение клиента 41](#_Toc367096786)

[Поведение сервера 41](#_Toc367096787)

[Возможные сценарии поведения приложения 42](#_Toc367096788)

[Генерация ссылок для платежа 42](#_Toc367096789)

[URI 42](#_Toc367096790)

[Методы 43](#_Toc367096791)

[Структуры данных 43](#_Toc367096792)

[Поведение клиента 43](#_Toc367096793)

[Поведение сервера 43](#_Toc367096794)

[Список городов 44](#_Toc367096795)

[URI 44](#_Toc367096796)

[Методы 44](#_Toc367096797)

[МодификаторыURI 44](#_Toc367096798)

[Структуры данных 44](#_Toc367096799)

[Поведение клиента 45](#_Toc367096800)

[Поведение сервера 45](#_Toc367096801)

[Список станций метро 45](#_Toc367096802)

[URI 45](#_Toc367096803)

[Методы 45](#_Toc367096804)

[МодификаторыURI 45](#_Toc367096805)

[Структуры данных 46](#_Toc367096806)

[Поведение клиента 46](#_Toc367096807)

[Поведение сервера 46](#_Toc367096808)

[Список киносетей 47](#_Toc367096809)

[URI 47](#_Toc367096810)

[Методы 47](#_Toc367096811)

[МодификаторыURI 47](#_Toc367096812)

[Структуры данных 47](#_Toc367096813)

[Поведение клиента 47](#_Toc367096814)

[Поведение сервера 47](#_Toc367096815)

[Координаты кинотеатров 48](#_Toc367096816)

[URI 48](#_Toc367096817)

[Методы 48](#_Toc367096818)

[Модификаторы URI 48](#_Toc367096819)

[Структуры данных 49](#_Toc367096820)

[Поведение клиента 49](#_Toc367096821)

[Поведение сервера 49](#_Toc367096822)

[Список заказов 50](#_Toc367096823)

[URI 50](#_Toc367096824)

[Методы 50](#_Toc367096825)

[Модификаторы URI 50](#_Toc367096826)

[Структуры данных 51](#_Toc367096827)

[Поведение клиента 51](#_Toc367096828)

[Поведение сервера 51](#_Toc367096829)

[Добавление кинотеатра в список избранных 52](#_Toc367096830)

[URI 52](#_Toc367096831)

[Методы 52](#_Toc367096832)

[Структуры данных 52](#_Toc367096833)

[Поведение клиента 52](#_Toc367096834)

[Поведение сервера 52](#_Toc367096835)

[Удаление кинотеатра из списка избранных 53](#_Toc367096836)

[URI 53](#_Toc367096837)

[Метод 53](#_Toc367096838)

[Структуры данных 53](#_Toc367096839)

[Поведение клиента 53](#_Toc367096840)

[Поведение сервера 53](#_Toc367096841)

[Программы лояльности 54](#_Toc367096842)

[URI 54](#_Toc367096843)

[Методы 54](#_Toc367096844)

[Структуры данных 54](#_Toc367096845)

[Поведение клиента 54](#_Toc367096846)

[Поведение сервера 54](#_Toc367096847)

[Расчет суммы баллов для программы лояльности 55](#_Toc367096848)

[URI 55](#_Toc367096849)

[Структуры данных 55](#_Toc367096850)

[Поведение клиента 55](#_Toc367096851)

[Поведение сервера 55](#_Toc367096852)

[Проверка номеров карт программ лояльности 56](#_Toc367096853)

[URI 56](#_Toc367096854)

[Структуры данных 56](#_Toc367096855)

[Поведение клиента 56](#_Toc367096856)

[Поведение сервера 56](#_Toc367096857)

[Привязка заказа к программе лояльности 57](#_Toc367096858)

[URI 57](#_Toc367096859)

[Структуры данных 57](#_Toc367096860)

[Поведение клиента 57](#_Toc367096861)

[Поведение сервера 57](#_Toc367096862)

[Профиль пользователя 58](#_Toc367096863)

[URI 58](#_Toc367096864)

[Методы 58](#_Toc367096865)

[Структуры данных 58](#_Toc367096866)

[Поведение клиента 59](#_Toc367096867)

[Поведение сервера 59](#_Toc367096868)

[Профиль пользователя – изменения номеров карт программ лояльность 60](#_Toc367096869)

[URI 60](#_Toc367096870)

[Методы 60](#_Toc367096871)

[Структуры данных 60](#_Toc367096872)

[Профиль пользователя – Привязка заказов к Удаление кинотеатра из списка избранных 61](#_Toc367096873)

[URI 61](#_Toc367096874)

[Метод 61](#_Toc367096875)

[Структуры данных 61](#_Toc367096876)

[Поведение клиента 61](#_Toc367096877)

[Поведение сервера 61](#_Toc367096878)

[Отмена заказа клиентом 62](#_Toc367096879)

[URI 62](#_Toc367096880)

[Метод 62](#_Toc367096881)

[Структуры данных 62](#_Toc367096882)

[Поведение клиента 64](#_Toc367096883)

[Поведение сервера 64](#_Toc367096884)

[Связка сессии мобильного приложения с учетной записью сайта «Киноход» 65](#_Toc367096885)

[URI 65](#_Toc367096886)

[Структуры данных 65](#_Toc367096887)

[Поведение клиента 65](#_Toc367096888)

[Поведение сервера 65](#_Toc367096889)

[Отвязка сессии мобильного приложения от учетной записью сайта «Киноход» 66](#_Toc367096890)

[URI 66](#_Toc367096891)

[Поведение клиента 66](#_Toc367096892)

[Поведение сервера 66](#_Toc367096893)

[Приложение 1 67](#_Toc367096894)

[Аутентификация приложения 67](#_Toc367096895)

[Управление сессиями 67](#_Toc367096896)

[Регистрация клиента 67](#_Toc367096897)

[URI 67](#_Toc367096898)

[Методы 67](#_Toc367096899)

[Модификаторы URI 68](#_Toc367096900)

[Структуры данных 68](#_Toc367096901)

[Поведение клиента 69](#_Toc367096902)

[Поведение сервера 69](#_Toc367096903)

[Авторизация мобильного приложения с учетной записью «Киноход» 70](#_Toc367096904)

[Приложение 2 71](#_Toc367096905)

[Аутентификация партнера 71](#_Toc367096906)

# Термины и обозначения

Платёжный Агрегатор (ПА)

# Общее описание протокола

## Введение

Данное API определяет язык и протокол для взаимодействия клиентов с сервером Кинопортала (КП). Основной целью данного протокола является его относительная легкая реализация в разных языках программирования на разных платформах.

Протокол оперирует *ресурсами* в виде *объектов* и *коллекций*, а также операциями над ними. Он определен на базе протокола HTTP1.1 (в соответствии с RFC2616) и использует стандартные методы протокола HTTP (GET, POST, PUT, DELETE)для запроса и изменения данных. Клиент взаимодействует с сервером по принципам HTTPв формате запрос-ответ.

## Сервер

Сервер API является веб-сервером и доступен по адресу

http(s)://<HOST>/api/rest

## Требования к протоколу

### Безопасность

Поскольку протокол предполагает обмен чувствительными (sensitive) данными, такими как личные данные пользователя и финансовые операции. Для обеспечения безопасности соединения с сервером осуществляются с использованием защищенного соединения (https) и аутентификации запросов.

### Версионность

В случае изменений протокола, объектной модели сущностей протокола и прочих интерфейсов сервер обеспечивает обратную совместимость с клиентами, ориентированными на прежние версии протокола.

Для версионирования протокола используются разные URI доступа к серверу:

http(s)://<HOST>/api/rest/v1

# Сущности протокола

## Ресурсы

Любой объект в системе в единичном экземпляре является объектом. Примерами объектов могут быть: *пользователь* (user), *транзакция* (transaction), *форма* (form)и т.п. Каждый объект имеет свой id в виде числового идентификатора, уникального среди объектов данного типа.

Множественные объекты объединяются в коллекции. Например, *пользователи* (users), *платежи* (transactions).

Каждый ресурс имеет свой URI в виде пути относительно точки доступа к серверу. Например, все фильмы в системе:

/movies

фильм с id=59:

/movies/59

Объекты и коллекции могут образовывать иерархические структуры. Например, кинотеатры, в которых идет данный фильм:

/movies/{id}/cinemas

## Модификаторы доступа

Модификаторы доступа – дополнительные параметры в URI для изменения результатов выдачи, например, для фильтрации или сортировки.

Пример.

/movies – возвращает полный список фильмов в системе.

С использованием модификаторов этот список можно ограничить фильмами, имеющими сеансы в определенную дату, отсортировать по рейтингу, и выдать первые 20 записей (при постраничном отображении)

/movies?date=01012011sortby=rating&rangeStart=0&limit=20

## Операции

Все операции c ресурсами выполняются стандартными методами протокола HTTP.

|  |  |  |
| --- | --- | --- |
| Метод | Коллекции | Объекты |
| GET | Запрашивает выборку данных из коллекций. Возвращается массив объектов | Возвращается представление объекта |
| PUT | Не применим | Производит обновление данных объекта |
| POST | Вставляет новый объект в коллекцию | Не применим |
| DELETE | Не применим | Удаляет объект |

Для облегчения тестирования, и использования в средах, не поддерживающих перечисленные методы, методы могут перекрываться атрибутом requestMethod в URLзапроса. Самое действие при этом выполняется методом POST или GET.

Например,

POST/movies/59?requestMethod=PUT

эквивалентно

PUT/movies/59 HTTP/1.1

## Неподдерживаемые операции

В случае получения неподдерживаемого запроса (PUT, DELETE для коллекции, POST–для объекта, с учетом переопределения метода параметром requestMethod), сервер отвечает ответом 405 Method Not Allowed. При этом в заголовке Allowed сервер перечисляет разрешенные методы для данного ресурса.

## Представление данных

Для поддержки различных типов возвращаемых данных клиент должен передавать в заголовке Content-type тип возвращаемых данных. На текущий момент поддерживается значение заголовка application/json, что соответствует представлению ресурса в формате JSON ([RFC 4627](http://tools.ietf.org/html/rfc4627)). Даты в ответе имеют формат в соответствии с [ECMA-262](http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf), MIME-тип возвращаемого контента: application/json.

## Кэширование

Для экономии трафика, ускорения обмена данными клиент МОЖЕТ запрашивать данные с использованием HTTP-заголовкаIf-Modified-Since. Серверу СЛЕДУЕТ вернуть ответ 304 Not Modified, если запрашиваемый ресурс не изменился после указанной даты. Однако сервер МОЖЕТ вернуть результат запроса, как если бы он был изменен.

## Использование Cookies

Клиент должен поддерживать механизм cookiesв соответствии с [RFC 2965](http://tools.ietf.org/html/rfc2965). Сервер может сохранять на клиенте любую информацию, например, ближайший или выбранный город, идентификаторы сессий для отслеживания поведения клиента и т.п.

В большинстве случаев передаваемые cookies будут сеансовыми (с неустановленным Max-Age). Клиент должен удалить эти cookies при завершении приложения. На платформах, которые предполагают уничтожение процесса приложения и его последующее восстановление, например, Android, в некоторых сценариях (при входящем звонке), будем считать, что уничтожение и восстановление процесса не является завершением приложения и не приводит к уничтожению cookies.

Клиентские приложения никаким образом не должны устанавливать или изменять cookies, не должны каким-либо образом полагаться на их набор и значения.

## Ответы сервера

Ответы сервера специфичны для каждого запроса и описываются в соответствующих разделах данного документа. В случае, когда параметр ответа помечен, как опциональный, он всё равно включается в ответ со значением null. Следующие ответы сервера не привязаны к конкретному типу запросов и могут быть возвращены клиенту при любом запросе.

|  |  |  |
| --- | --- | --- |
| Ответ | Описание | Действия клиента |
| 503 Service Unavailable | Сервер не доступен в данный момент вследствие перегрузки или сервисных работ | Выдать сообщение клиенту и откатить текущую операцию. Если ответ следует за операцией регистрации, то клиент должен завершить приложение |
| 400 Bad Request | Передаваемая клиентом структура данных не соответствует протоколу | Выдать сообщение клиенту и откатить текущую операцию. Если ответ следует за операцией регистрации, то клиент должен завершить приложение |
| 403 Forbidden | Данное действие запрещено данному клиенту в данный момент времени. Если тело в ответе имеет тип application/json,то ответ содержит структуру ErrorDesc | Если тело в ответе имеет тип application/json, то клиент может опираться на код ошибки из ответа. |
| 500 Internal Error | Произошла ошибка при обращении к внешнему источнику данных | Выдать сообщение клиенту и откатить текущую операцию. Если ответ следует за операцией регистрации, то клиент должен завершить |

Структура ErrorDesc

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| errorCode | int | Да | Код ошибки |
| errorDesc | string | Нет | Текстовое описание ошибки |

Значения errorCode

|  |  |  |
| --- | --- | --- |
| Значение | Описание | Действие клиента |
| 1 | Клиент не зарегистрирован или регистрация истекла | Клиент должен заново инициировать регистрацию на сервере и повторить изначальный запрос. |
| 2 | Бан. Данный клиент заблокирован. Все последующие запросы получат такой же ответ | Выдать сообщение об ошибке |
| 3 | Данный запрос не разрешён для клиента | Выдать сообщение об ошибке и остановить подобные запросы |
| 5 | Произошла ошибка при обращении к внешнему ресурсу | Выдать сообщение об ошибке |
| 6 | Клиент исчерпал попытки обращения к данному методу | Выдать сообщение об ошибке |
| 80 | Моб. сессия клиента не связана с уч.записью на сайте | Связать сессию с уч.записью на сайте с помощью метода /user/bind |
| 99 | Неизвестная ошибка | Выдать сообщение об ошибке |

# Безопасность

## Соединение

Подключение к серверу осуществляется по протоколу HTTPS. В тестовом и отладочном режиме сервер может быть доступен по протоколу HTTP без шифрования.

При попытке подключения к серверу по протоколу HTTP сервер возвращает ответ 426 Upgraderequired.

HTTP/1.1 426 Upgrade Required

Upgrade: TLS/1.0, HTTP/1.1

Connection: Upgrade

## Аутентификация

Конкретные технические детали процедур аутентификации и регистрации различаются для партнеров и мобильных приложений и вынесены в Приложения 1и 2.

# Фильмы

### URI

**/movies**

Обращение к списку фильмов

**/movies/{id}**

Обращение к элементу коллекции фильмов

### Методы

GET

Клиент запрашивает список фильмов.

Ответ

200 OKв случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры MovieInfoShort.

При обращении к элементу коллекции – единичный объект MovieInfo.

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/movies

sort = { name | rating| showcount | revenue| premiere} – сортирует коллекцию в алфавитном порядке| по рейтингу | популярности (числу показов) | сборам в России | дате премьеры в Росии

filter = {soon | premiere| wanted} – выводит список фильмов из категорий “Скоро на экранах” , “Новинки недели” и “Хочу пойти” соответственно.

date=DDMMYYYY - осуществляет выборку фильмов, идущих в прокате в назначенный день. Если данный параметр не передан, осуществляется выборка фильмов идущих в прокате в текущий день.

city=city\_id - осуществляет выборку фильмов, идущих в прокате в указанном городе

rangeStart = N, ограничивает выборку, начиная с N-го элемента (0-basedindex), используется в паре с limit для постраничного отображения.

limit= Y. Ограничивает выборку количеством элементов Y.Используется в паре с rangeStart для постраничного отображения.

search = строка в кодировке UTF-8. Ограничивает выборку элементами, название которых содержит подстроку.

genre = genre\_id – выводит список отфильтрованный по идентификатору жанра.

### Структуры данных

Структура MovieInfo / MovieInfoShort (поля, входящие в MovieInfoShort отмечены знаком «+»)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Short | Поле | Тип | Обяз. | Комментарий |
| + | id | int | Да | ID фильма |
| + | title | string | Да | Название фильма |
| + | duration | numeric | Нет | Продолжительность, минуты |
| + | originalTitle | string | Нет | Название фильма на языке оригинала |
|  | productionYear | string | Нет | Год выпуска |
| + | premiereDateRussia | date | Нет | Дата начала проката в России |
|  | premiereDateWorld | date | Нет | Дата начала проката в мире |
|  | budget | numeric | Нет | Бюджет фильма в долларах |
| + | countries | Массив string | Нет | Страны изготовления |
|  | producers | Массив string | Нет | Продюсеры |
|  | companies | Массив string | Нет | Студии |
|  | directors | Массив string | Нет | Режиссеры |
|  | actors | Массив string | Нет | Актеры |
| + | genres | Массив структур genre | Нет | Жанры |
|  | annotationShort | string | Нет | Краткая аннотация к фильму |
|  | annotationFull | string | Нет | Aннотация к фильму |
| + | ageRestriction | string | Нет | Возрастные ограничения к фильму вида “возраст+”, например “18+” |
| + | ageRating | Массив string | Нет | Рейтинг по классификации изготовителя |
| + | grossRevenueRus | numeric | Нет | Кассовые сборы в России |
| + | grossRevenueWorld | numeric | Нет | Кассовые сборы в мире |
| + | poster | string | Нет | Постер к фильму |
| + | trailers | Массив структур Trailer | Нет | Трейлер к фильму |
|  | images | Массив string | Нет | Изображения (кадры из фильма) |
| + | rating | Структура Rating | Нет | Структура с оценками фильма |
| + | countScreens | int | Нет | Кол-во экранов фильма |
| + | attractor | string | Нет | Ключевой аттрактор |
| + | is\_running | boolean | Нет | Флаг продажи билетов |
| + | Is\_wanted | boolean | Нет | Флаг `Есть ли фильм в «хочу пойти»` |
| + | formats | Массив string | Нет | Доступные форматы для данного фильма |
| + | firstSeanceStime | date | Нет | Дата первого сеанса в городе |
|  | imdb\_id | string | Нет | Идентификатор фильма в IMDB |

Структура Rating

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| rating | int | Нет | Синтетический рейтинг |
| kinopoisk | numeric | Нет | Рейтинг Кинопоиска |
| rtCritics | int | Нет | Рейтинг критиков Rotten Tomatoes |
| rtAudience | int | Нет | Рейтинг аудитории Rotten Tomatoes |

Структура Trailer

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| filename | string | Да | Трейлер к фильму |
| preview | string | Да | Превью трейлера |
| <preset1> | TrailerInfo | Да | Структура TrailerInfo |
| <preset2> | TrailerInfo | Нет | Структура TrailerInfo |
| … | … | … | … |
| <presetN> | TrailerInfo | Нет | Структура TrailerInfo |
| source | TrailerSourceInfo | Нет | Структура TrailerSourceInfo |

Структура TrailerInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| filename | string | Да | Имя файла трейлера |
| duration | int | Да | Продолжительность трейлера в секундах |

Структура TrailerSourceInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| filename | string | Да | Имя файла трейлера |
| contentType | int | Да | Сontent-Type для файла трейлера |

Структура genre

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | Id жанра |
| name | string | Да | Имя жанра |

Значения в полях poster и images представляют собой имя файла изображения или список имён файлов изображений соответственно. Имя файла изображения имеет формат <uuid>.<ext>, где <uuid> формируется согласно rfc4122, а <ext> представляет собой расширение файла и может принимать значения из следующего списка: jpeg, jpg, gif, png.

Поля <trailer1>…<trailerN> представляют собой имена различных представлений данного трейлера (к примеру, настроек конвертации видеоролика). На текущий момент поддерживаются следующие значения наименований этих полей:

1. default: значение по умолчанию
2. mobile\_mp4: представление трейлера для мобильных устройств.

Значение в поле filename структуры Trailer представляет собой имя файла трейлера. Имя файла имеет формат <uuid>\_<preset>.flv, где <uuid> формируется согласно rfc4122.

Значение в поле source представляет собой информацию об исходном файле трейлера и имеет следующие поля:

1. filenameвида <uuid>.<ext>– имя исходного файла трейлера, полный путь к нему формируется как http(s)://<HOST>/o/<ab>/<cd>/<uuid>.<ext>
2. contentType – Content-Type файла трейлера, определяющий его формат;

URL полного изображения формируется следующим образом:  
http(s)://<HOST>/o/<ab>/<cd>/<uuid>.<ext>  
URLизображения предпросмотра формируется следующим образом:  
http(s)://<HOST>/p/<w>x<h>/<ab>/<cd>/<uuid>.<ext>

Где:

* <HOST> -- конфигурируемое значение для хоста с изображениями. На текущий момент такой же, как хост для API;
* <w> и <h> -- ширина и высота изображения соответственно. При несовпадении пропорций по сравнению с оригинальным изображением, оно центрируется, пропорционально масштабируется и обрезается до нужных размеров.
* <ab> и <cd> -- первые 4 символа <uuid>

Поле Rating

В случае отсутствующего рейтинга или равного нулю – отдается -1.

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется

# Жанры фильмов

### URI

**/genres**

Получение списка жанров

### Методы

GET

Клиент запрашивает список жанров фильмов.

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры Genre.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/genres

filter = {soon | premiere | running} – выводит список жанров фильмов из категорий “Скоро на экранах”, “Новинки недели”, “Идущих в кинотеатрах” соответственно.

limit= Y. Ограничивает выборку количеством элементов Y.(По умолчанию – 5).

city=city\_id - осуществляет выборку кинотеатров в указанном городе. Используется только при значении фильтра filter=’soon’

Структура genre

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | Id жанра |
| name | string | Да | Имя жанра |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется

# Форматы фильмов

### URI

**/movies/{id}/formats**

Получение списка форматов фильма

### Методы

GET

Клиент запрашивает список форматов фильмов.

Ответ

200 OKв случае успеха.

При обращении возвращается массив String форматов для фильма, либо пустой массив – при наличии только формата 2D.

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/movie/{id}/formats

city=city\_id - осуществляет выборку форматов фильма, доступных в указанном городе.

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется

# Комментарии и оценки к фильму

### URI

**/movies/{id}/comments**

Пользовательские комментарии и оценки.

**/movies/{id}/reviews**

Рецензии к фильму.

### Методы

GET

Клиент запрашивает список комментариев/оценок и рецензий к фильму

Ответ

При обращении к коллекции comments сервер возвращает массив структур Comment.

При обращении к коллекции reviews сервер возвращает массив структур Review.

200 OK в случае успеха.

404 Not Found, при обращении к элементу коллекции с несуществующим ID.

POST

Публикация комментариев применима только к коллекции comments. Публикация в коллекцию reviews не поддерживается (сервер возвращает 405).

Клиент публикует оценку/комментарий

Ответ

200 OK / 201 Created в случае успеха.

404 Not Found, при обращении к элементу коллекции с несуществующим ID.

DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/movies/{id}/comments  
/movies/{id}/reviews

rangeStart = N, ограничивает выборку, начиная с N-го элемента (0-based index), используется в паре с limit для постраничного отображения.

limit= Y. Ограничивает выборку количеством элементов Y.Используется в паре с rangeStart для постраничного отображения.

/movies/{id}/comments

sortby = {date | rating }, сортирует выборку по дате | оценке.

sortorder = { reverse }в обратном порядке.

### Структуры данных

Структура Comment

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| user | Структура User | Да | Пользователь |
| rating | int | Нет | Оценка пользователя (-1/+1) |
| comment | string | Нет | Комментарий пользователя |

Структура User

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| user | string | Да | Имя пользователя (никнейм, логин) |
| displayName | string | Нет | Отображаемое имя пользователя (если предусмотрено в социальной сети) |
| network | string | Да | Имя социальной сети, в которой зарегистрирован пользователь:  - vkontakte  - facebook  - twitter |

Структура Review

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| author | string | Нет | Имя рецензента |
| authorTitle | string | Нет | Должность, звание, титул. Например, «Главный редактор журнала «Афиша» |
| comment | string | Да | Текст комментария |

### Поведение клиента

Клиент может опубликовать либо только оценку, либо только комментарий, либо оба сразу. В зависимости от выбора пользователя заполняются соответствующие структуры

### Поведение сервера

Аутентификация в соц. сетях – TBD, требует изучения.

# Добавление фильма в “хочу пойти”

Метод требует предварительной «связки» моб.сессии с уч.записью на сайте «Киноход» с помощью метода /user/bind. В противном случае, возвращается ошибка с HTTP-статусом 412 и errorCode = 80

### URI

**/user/movies/wanted/add**

Добавление фильма в «хочу пойти»

### Методы

POST

Добавление фильма в список «хочу пойти»

Клиент передает в теле структуру UserMoviesWantedRequest

Ответ

200 OK/ 201 Created в случае успеха.

GET, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура UserMoviesWantedRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| movies | Array | Да | Массив ID фильмов |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Удаление фильма из “хочу пойти”

Метод требует предварительной «связки» моб.сессии с уч.записью на сайте «Киноход» с помощью метода /user/bind. В противном случае, возвращается ошибка с HTTP-статусом 412 и errorCode = 80

### URI

**/user/movies/wanted/delete**

Удаление фильма из «хочу пойти»

### Методы

POST

Удаление фильма из списка «хочу пойти»

Клиент передает в теле структуру UserMoviesWantedRequest

Ответ

200 OK/ 201 Created в случае успеха.

GET, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура UserMoviesWantedRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| movies | Array | Да | Массив ID фильмов |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Фильмы, идущие в кинотеатрах в ближайший час

### URI

**/nearest**

Получение списка фильмов, идущих в кинотеатрах в ближайший час (от 31 до 60 минут)

### Методы

GET

Клиент запрашивает список фильмов, идущих в кинотеатрах в ближайший час

Ответ

200 OKв случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры Nearest.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/nearest

city=city\_id - осуществляет выборку фильмов по кинотеатрам в указанном городе

latitude = число. Географическая широта пользователя. Используется в паре с longitude при сортировке кинотеатров по удаленности.

longitude = число. Географическая долгота пользователя. Используется в паре с latitude при сортировке кинотеатров по удаленности.

### Структуры данных

Структура Nearest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID сеанса |
| startTime | datetime | Да | Фактическое время показа, с указанием смещения UTC. Это время может отличаться от даты показа. Например показы с началом в 1 час ночи относятся к предыдущему дню, хотя фактически идут в следующие сутки. |
| is3d | boolean | Нет | Флаг 3D |
| isImax | boolean | Нет | Флаг IMAX |
| isSaleAllowed | boolean | Нет | Флаг возможности покупки билетов |
| cinemaId | int | Да | ID кинотеатра |
| cinemaName | string | Да | Название кинотеатра |
| movieId | int | Да | ID фильма |
| movieName | string | Да | Название фильма |
| distance | int | Нет | Расстояние до кинотеарта |
| subway\_stations | Структура SubwayStation | Нет | Список станции метро рядом с кинотеатром |

Структура SubwayStation

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | numeric | Да | ID метро |
| name | string | Да | Название станции метро |
| subway\_station\_distance | int | Нет | Расстояние от кинотеатра до метро |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется

# Кинотеатры

### URI

**/cinemas**

Обращение к списку кинотеатров

**/cinemas/{id}**

Обращение к элементу коллекции кинотеатров

### Методы

GET

Клиент запрашивает список кинотеатров

Ответ

200 OKв случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры CinemaInfoShort. При обращении к элементу коллекции – единичный объект CinemaInfo.

404 Not Found, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/cinemas

sort = { name| distance} – сортирует коллекцию в алфавитном порядке | по удаленности.

city=city\_id - осуществляет выборку кинотеатров в указанном городе

network = network\_id – осуществляет выборку кинотеатров для указанной киносети.

rangeStart = N, ограничивает выборку, начиная с N-го элемента (0-based index), используется в паре с limit для постраничного отображения.

limit= Y. Ограничивает выборку количеством элементов Y.Используется в паре с rangeStart для постраничного отображения.

search = строка в кодировке UTF-8. Ограничивает выборку элементами, название которых содержит подстроку.

latitude = число. Географическая широта пользователя. Используется в паре с longitude при сортировке кинотеатров по удаленности.

longitude = число. Географическая долгота пользователя. Используется в паре с latitude при сортировке кинотеатров по удаленности.

subwayStation = IDстанции метро. Используется при сортировке по удаленности от метро.

subwayFilter = IDстанции метро. Используется при фильтрации списка по метро.

isSale = {0|1} Флаг продаваемости билетов в кинотеатре

isFav = { 0 | 1 } - флаг избранных кинотеатров. В случае 1 выдаёт только избранные кинотеатры пользователя.

/cinemas/<id>

latitude = число. Географическая широта пользователя. Используется в паре с longitude для получения расстояния до кинотеатра.

longitude = число. Географическая долгота пользователя. Используется в паре с latitude при сортировке кинотеатров по удаленности.

### Структуры данных

Структура CinemaInfo / CinemaInfoShort (поля входящие в CinemaInfoShort отмечены знаком «+»)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Short | Поле | Тип | Обяз. | Комментарий |
| + | id | int | Да | ID кинотеатра |
| + | title | string | Да | Название кинотеатра |
| + | shortTitle | string | Нет | Короткое название кинотеатра |
|  | description | string | Нет | Описание кинотеатра |
|  | website | string | Нет | Веб-сайт кинотеатра |
| + | phones | Массив структур Phone | Нет | Телефоны кинотеатра |
|  | city | int | Да | ID города кинотеатра |
| + | address | string | Нет | Адрес кинотеатра |
| + | location | Структура Location | Нет | Географическое положение кинотеатра |
|  | photo | Массив string | Нет | Фото кинотеатра, залов |
| + | distance | int | Нет | Расстояние до кинотеатра в метрах (при условии передачи в параметрах значений latitude и longitude) |
| + | network | int | Нет | Идентификатор киносети |
|  | isSale | boolean | Нет | Флаг продаваемости билетов в кинотеатре |
| + | isFav | boolean | Нет | Флаг нахождения кинотеатра в “Избранном” пользователя |
| + | subway\_stations | Структура SubwayStation | Нет | Список станции метро рядом с кинотеатром |
| + | goodies | Массив string | Нет | Список дополнительных сервисов(goodies) для кинотеатра. Каждый элемент массива может принимать одно из значений указанных в столбце «значение» таблицы «Расшифровка значений Goodies» |
| + | isSale | boolean | Нет | Флаг наличия продаж для кинотеатра |
| + | isFav | boolean | Нет | Флаг нахождения кинотеатра в избранных |

Структура Phone

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| number | string | Да | Телефон |
| description | string | Нет | Описание. Например, «Автоинформатор», «Служба бронирования» и т.п. |

Структура Location

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| latitude | numeric | Да | Широта |
| longitude | numeric | Да | Долгота |

Структура SubwayStation

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | numeric | Да | ID метро |
| name | string | Да | Название станции метро |
| subway\_station\_distance | Int | Нет | Расстояние от кинотеатра до метро |

Расшифровка значений Goodies

|  |  |
| --- | --- |
| Значение | Расшифровка |
| have\_bar | Бар |
| in\_shop | Расположен в ТЦ |
| have\_cafe | Кафе |
| have\_play\_area | Игровая зона |
| have\_wifi\_free | Бесплатный Wi-Fi |
| have\_dvd\_shop | Магазин DVD |
| have\_karaoke | Караоке |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Расписание сеансов

### URI

**/cinemas/{id}/schedules**

Расписание сеансов кинотеатра

**/movies/{id}/schedules**

Расписание сеансов фильма

### Методы

GET

Клиент запрашивает список сеансов

Ответ

200 OKв случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры Showtimes

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/cinemas/{id}/schedules

rangeStart = N, ограничивает выборку, начиная с N-го элемента (0-based index), используется в паре с limit для постраничного отображения.

limit= Y. Ограничивает выборку количеством элементов Y.Используется в паре с rangeStart для постраничного отображения.

date=DDMMYYYY - осуществляет выборку фильмов, идущих в прокате в назначенный день. Если данный параметр не передан, осуществляется выборка фильмов идущих в прокате в текущий день.

search = строка в кодировке UTF-8. Ограничивает выборку элементами, название которых содержит подстроку.

sinceNow = 1/0 – выводит с текущего момента / за весь день. Применяется только если не задан параметр date. По умолчанию 0.

format = "is\_imax" / "is\_3d" / "is\_imax,is\_3d" – ограничивает выборку расписания только с определенными форматами фильмов.

/movies/{id}/schedules

sort = { distance} – сортирует коллекцию в алфавитном порядке | по удаленности.

city=city\_id - осуществляет выборку кинотеатров в указанном городе.

network = network\_id – осуществляет выборку кинотеатров для указанной киносети.

rangeStart = N, ограничивает выборку, начиная с N-го элемента (0-based index), используется в паре с limit для постраничного отображения.

limit= Y. Ограничивает выборку количеством элементов Y.Используется в паре с rangeStart для постраничного отображения.

date=DDMMYYYY - осуществляет выборку фильмов, идущих в прокате в назначенный день. Если данный параметр не передан, осуществляется выборка фильмов идущих в прокате в текущий день.

search = строка в кодировке UTF-8. Ограничивает выборку элементами, название которых содержит подстроку.

latitude = число. Географическая широта пользователя. Используется в паре с longitude при сортировке кинотеатров по удаленности.

longitude = число. Географическая долгота пользователя. Используется в паре с latitude при сортировке кинотеатров по удаленности.

subwayStation = IDстанции метро. Используется при сортировке по удаленности от метро.

subwayFilter = IDстанции метро. Используется при фильтрации списка по метро.

sinceNow = 1/0 – выводит с текущего момента / за весь день. Применяется только если не задан параметр date. По умолчанию 0.

### Структуры данных

Структура Showtimes

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| movie | Структура MovieInfoShort | Да | Фильм |
| cinema | Структура CinemaInfoShort | Да | Кинотеатр |
| date | date | Да | Дата |
| schedules | Массив структур ScheduleInfoShort | Да | Показы данного фильма |

Структура ScheduleInfoShort

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID сеанса |
| time | string | Да | Время показа в формате “hh:mm” |
| startTime | datetime | Да | Фактическое время показа, с указанием смещения UTC. Это время может отличаться от даты показа. Например, показы с началом в 1 час ночи относятся к предыдущему дню, хотя фактически идут в следующие сутки. |
| isSaleAllowed | boolean | Да | Разрешена продажа билетов на этот сеанс |
| minPrice | numeric | Нет | Минимальная цена на билет |
| maxPrice | numeric | Нет | Максимальная цена на билет |
| maxSeatsInOrder | int | Да | Максимальное количество мест в заказе |
| is3d | boolean | Нет | Флаг 3D |
| isImax | boolean | Нет | Флаг IMAX |
| isOrigin | boolean | Нет | Признак показа фильма на языке оригинала |
| subtitle | Структура SubtitleInfo | Да | Информация о субтитрах фильма |
| language | Структура LanguageInfo | Да | Информация о языке фильма |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Информация о сеансе

### URI

**/schedules**

Все сеансы

**/schedules/{id}**

Информация о сеансе

### Методы

GET

Клиент запрашивает информацию о сеансе

Ответ

При обращении к /schedules возвращается ответ 403 Forbidden, errorCode = 3 (линейный список огромен и не несет пользы для клиента).

200 OKв случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры ScheduleInfo

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

Не применяются

### Структуры данных

Структура ScheduleInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| movie | Структура MovieInfoShort | Да | Фильм |
| cinema | Структура CinemaInfoShort | Да | Кинотеатр |
| date | Date | Да | Дата. |
| time | String | Да | Время показа в формате “hh:mm” |
| startTime | datetime | Да | Фактическое время показа, с указанием смещения UTC. Это время может отличаться от даты показа (date). Например, показы с началом в 1 час ночи относятся к предыдущему дню, хотя фактически идут в следующие сутки. |
| hall | String | Нет | Название зала, в котором проходит сеанс. Например, «Зал 2», «Малый зал» |
| is3d | boolean | Да | Флаг 3D |
| isImax | boolean | Да | Флаг IMAX |
| isOrigin | boolean | Да | Признак показа фильма на языке оригинала |
| isSaleAllowed | boolean | Да | Разрешена продажа билетов на этот сеанс |
| minPrice | numeric | Нет | Минимальная цена на билет |
| maxPrice | numeric | Нет | Максимальная цена на билет |
| seatCategories | Массив структур SeatCategory | Нет | Описание категорий мест в зале |
| maxSeatsInOrder | Int | Да | Максимальное число мест в заказе |
| subtitle | Структура SubtitleInfo | Да | Информация о субтитрах фильма |
| language | Структура LanguageInfo | Да | Информация о языке фильма |

Структура SeatCategory

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | Int | Да | ID категории |
| name | String | Да | Название категории |
| price | numeric | Да | Цена данной категории в рублях |

Структура LanguageInfo, структура SubtitleInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | Int | Да | ID категории |
| name | String | Да | Название языка |
| pred\_name | String | Да | Название языка (предл.падеж) |
| orig\_name | String | Да | Название языка на ориг.языке |
| short\_name | String | Да | Сокр. назв. языка на ориг.языке |
| greeting | String | Да | Приветствие на ориг. языке |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Информация о зале

### URI

**/schedules/{id}/hallscheme**

Информация о схеме зала и занятости мест

### Методы

GET

Ответ

200 OKв случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры HallScheme

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

Не применяются

### Структуры данных

Структура HallScheme

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | Int | Да | ID зала |
| name | String | Нет | Название зала. Например, «Зал 2», «Малый зал» |
| width | Int | Да | Ширина зала |
| height | Int | Да | Высота (длина) зала |
| screen\_X | Int | Да | Координата Х экрана |
| screen\_Y | Int | Да | Координата Y экрана |
| screen\_W | Int | Да | Ширина экрана |
| screen\_H | Int | Да | Высота экрана |
| seatCount | Int | Да | Количество мест в зале |
| sections | Массив структур Section | Да | Секции в зале |

Структура Section

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | Int | Да | ID секции |
| name | String | Нет | Название секции. |
| width | Int | Да | Ширина секции |
| height | Int | Да | Высота секции |
| seatCount | Int | Да | Количество мест в секции |
| seats | Массив структур SeatInfo | Да | Места в секции |

Структура SeatInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | Int | Да | ID места |
| x | Int | Да | координата Х места |
| y | Int | Да | координата Y мета |
| width | Int | Да | Ширина секции |
| height | Int | Да | Высота секции |
| status | String | Да | Статус места:  vacant – свободное  occupied–занятое |
| row | String | Да | Номер ряда («1», «2», «3», «A», «B», «С») |
| number | String | Да | Номер места в ряду («1», «2», «3», «A», «B», «С») |
| priceCategory | String | Да | Название ценовой категории |
| price | Int | Да | Цена в рублях |
| fragment | Int | Нет | Идентификатор группы мест. Все места с непустым и одинаковым значением этого поля должны вести себя как одно место при выборе |

### Поведение клиента

Описание структуры зала.

На данный момент структура зала двухуровневая - зал и секция. Как правило, в зале одна секция (исключение - Пушкинский). Все параметры ширин, высот и координат заданы в условных единицах (пикселях). Все координаты x и y объектов (мест, экрана) заданы относительно верхнего левого угла вышерасположенного в иерархии объекта.

Отрисовка:

Для зала необходимо отрисовать границы области по ширине и высоте

- рисуем экран, позиционируя в соответствии с x и y

- по списку секций рисуем границы секции

- по списку мест рисуем места, позиционируя их в соответствии с x и y

### Поведение сервера

Описание не требуется.

# Резервирование мест и оплата

Метод недоступен в API Партнера.

### URI

**/reservation  
/reservation/{id}**

### Методы

GET

Клиент запрашивает информацию о статусе резервирования

Ответ

При обращении к /reservation возвращается ответ 403 Forbidden, errorCode = 3 (линейный список огромен и не несет пользы для клиента)

200 OK в случае успеха, информация о бронировании ReservationInfo

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST

Клиент инициирует резервирование мест

Запрос

Клиент передает в теле структуру ReservationRequest

Ответ

200 OK и информация о статусе резервирования ReservationResponse. Статус 200OK не означает успешного резервирования. Он лишь означает успешность выполнения запроса.

403, errorCode=6 при исчерпании клиентом суточных лимитов на покупку билетов.

DELETE

Клиент инициирует отмену заказа

Ответ

200 OK в случае успеха, информация о бронировании ReservationInfo

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

### Модификаторы URI

Не применяются

### Структуры данных

Структура ReservationRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| scheduleId | int | Да | ID сеанса |
| seats | Массив id мест из структуры SeatInfo | Да | Места для резервирования |
| phone | string | Нет | Контактный телефон клиента |
| email | string | Нет | E-mail покупателя. |

Структура ReservationResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | string | Нет | IDрезервирования. null, в случае неудачи |
| status | int | Да | Результат операции:  0 – операция завершена успешно (резервирование совершено, оплата подтверждена), поле id содержит ID резервирования  Ненулевые значения – неуспешное резервирование.  1 – не удалось зарезервировать, указанные места заняты. Клиент должен запросит повторно схему зала и статус мест  7 – программа лояльности не найдена  8 – номер карты программы лояльности не прошел верификацию |
| statusText | string | Нет | Текстовое сообщение для пользователя |
| reservation | Структура ReservationInfo | Нет | Информация о резервировании, если оно успешно произведено. |
| paymentTypes | Массив структур PaymentType | Да | Средства платежа доступные для данного заказа |
| loyalties | Массив структур LoyaltiesInfo | Да | Список доступных программ лояльностей для данного резерва |

Структура PaymentType

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID платежного средства |
| name | string | Да | Название платежного средства (например “Банковская карта”) |
| nameInstrumental | string | Да | Название платежного средства в творительном падеже (например “Банковской картой”) |
| minSum | numeric | Да | Минимальная сумма заказа по данному платежу |
| maxSum | numeric | Да | Максимальная сумма заказа по данному платежу |
| enabled | boolean | Да | Флаг доступности данного платежа для оплаты данного заказа |

Структура LoyaltiesInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID программы лояльности |
| name | string | Да | Имя программы лояльности |
| amount | string | Да | Сумма начисляемых единиц валюты программы лояльности |
| currency | Структура Currency | Да | Валюта программы лояльности |

Структура ReservationInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | string | Нет | ID резервирования |
| reservationStatus | string | Да | Статус резервирования  reserved – зарезервировано  cancelled–отменено  approved – резервирование подтверждено после оплаты |
| paymentStatus | string | Да | Статус оплаты резервирования:  new – только создано, не было запроса на оплату  not\_paid– не было оплачено  paid – оплачено processed – находится в процессе обработки |
| price | numeric | Нет | Общая стоимость резервирования |
| fee | numeric | Нет | Размер комиссии |
| movieId | int | Да | IDфильма (фильм может более не существовать в системе) |
| movieTitle | string | Да | Название фильма |
| cinemaId | int | Да | IDкинотеатра (данный кинотеатр может более не существовать в системе) |
| cinemaName | string | Да | Название кинотеатра |
| cinemaAddress | string | Да | Адрес кинотеатра |
| cinemaLocation | структура Location | Да | Координаты кинотеатра |
| scheduleId | int | Да | IDсеанса. Информация по данному ID может не существовать в системе |
| seats | Массив структур ReservedSeat | Да | Информация о зарезервированных местах. |
| phone | string | Нет | Контактный телефон клиента |
| email | string | Нет | Email клиента |
| transactionId | string | Нет | Код транзакции платежного агрегатора, в случае успешной оплаты |
| reservationCode | string | Нет | Код резервирования в случае успеха |
| startTime | datetime | Да | Фактическое время показа, с указанием смещения UTC. Это время может отличаться от даты показа (date). Например показы с началом в 1 час ночи относятся к предыдущему дню, хотя фактически идут в следующие сутки. |
| poster | string | Нет | Постер к фильму |
| qrcodeURL | string | Нет | Ссылка на изображение qrcode’а билета |
| pkpassURL | string | Нет | Ссылка на PassBook файл |

Структура ReservedSeat

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| seatCaption | string | Нет | Строка вида «Зал 2, секция 3, ряд 2, место F». |
| seatPrice | numeric | Нет | Цена места. |
| hallId | int | Да | IDзала |
| hallName | string | Да | Название зала |
| sectionId | int | Да | ID секции в зале |
| sectionName | string | Да | Название секции |
| seatId | int | Да | ID места |
| row | string | Да | Номер ряда |
| number | string | Да | Номер места в ряду |

### Поведение клиента

1. Клиент инициирует резервирование, отправляя структуру ReservationRequest методом POST
2. Результат резервирования сообщается в структуре ReservationResponse
   1. В случае неудачи выдается сообщение пользователю. Если резервирование не удалось из-за конфликта мест, клиент запрашивает повторно схему зала и статусы мест и начинает с пункта 1.
3. В случае успешного резервирования, клиент получает информацию о резервировании (код, сумму к оплате) одновременно с результатом резервирования в структуре ReservationInfo;
4. Клиент открывает платежную форму, по URL, переданному в параметре paymentURL;
5. После завершения оплаты клиент закрывает форму по факту редиректа на адрес, переданный в параметре closeURL;
6. Клиент периодически запрашивает /reservation методом GET. Логика работы клиента регламентируется следующей таблицей в пункте “Возможные сценарии поведения приложения”.
7. Клиент отображает пользователю соответствующую информацию.

### Поведение сервера

1. Получив запрос ReservationRequest, сервер инициирует резервирование в киносети или получает необходимую информацию о статусе заказа (при вызове методом POST или GET соответственно);
2. Результат резервирования возвращается клиенту в структуре ReservationResponse.

Возможные сценарии поведения приложения

Возможны ситуации, когда не удаётся с первого раза подтвердить статус платежа в ПА и/или резервирования в киносети. В случае, если пользователь возвращается в приложение, когда происходит подобная ситуация, клиент должен в течение определенного периода времени с определенной периодичностью запрашивать статус заказа (оплаты и резервирования) на сервере. На вебе периодичность опросов равна 6-и секундам, период времени, в течение которого осуществляются запросы, равен 2-ум минутам.

При запросе статуса резервирования возможны следующие сочетания статусов оплаты (paymentStatus) и резервирования (reservationStatus).

|  |  |  |  |
| --- | --- | --- | --- |
| paymentStatus | reservationStatus | Описание ситуации | Действия клиента |
| new | reserved | Пользователь по какой-то причине не был перенапрален на ПА для оплаты | Перенаправить пользователя на оплату |
| new | cancelled | Не удалось подтвердить статус платежа, истёк таймаут на подтверждение статуса в кинотеатре. | Сообщить пользователю об ошибке резервирования и сроках возврата средств за неуспешную покупку. |
| paid | reserved | Платёж прошёл, не удалось подтвердить резервирование мест в кинотеатре. | Продолжать запрашивать статус заказа. |
| paid | approved | Платёж прошёл, статус в кинотеатре подтверждён. | Показать экран успешной покупки. |
| paid | cancelled | Платёж прошёл, истёк таймаут на подтверждение статуса в кинотеатре. | Сообщить пользователю об ошибке резервирования и сроках возврата средств за неуспешную покупку. |
| not\_paid | reserved | Платёж не прошёл (возможно, не хватило средств). | Сообщить пользователю об ошибке. |
| not\_paid | cancelled | Платёж не прошёл (возможно, не хватило средств), истёк таймаут на подтверждение статуса в кинотеатре. | Сообщить пользователю об ошибке. |
| processed | reserved | Не удалось подтвердить статус платежа. | Продолжать запрашивать статус заказа. |

По истечении отведенного на подтверждение статуса заказа времени клиент должен показать пользователю информацию о невозможности подтвердить статус за отведенное время с предложением дождаться извещения о покупке посредством email или SMS.

### Генерация ссылок для платежа

### URI

/reservation/{id заказа}/payment/{id средства платежа}

### Методы

GET

Клиент запрашиваетURL’ы для совершения платежа

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры Payment

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура Payment

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| paymentURL | string | Да | URL платежной формы |
| closeURL | string | Да | URL закрытия платежной формы |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Список городов

Доступ к справочной информации по городам

### URI

**/cities  
/cities/{id}**

Обращение к коллекции городов

### Методы

GET

Клиент запрашивает информацию о городах

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры City. При обращении к элементу коллекции – единичный объектCity.

404 Not Found, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### МодификаторыURI

/cities

sort = { name| distance } – сортирует коллекцию в алфавитном порядке либо по удаленности от заданных координат

rangeStart= X- осуществляет выборку, начиная с элемента X (нумерация с 0). Используется совместно с limit.

limit= Y. Ограничивает выборку количеством элементов Y.

latitude = L1 – географическая широта заданной точки в виде числа с точкой, используется в паре с longitudeпри сортировке по расстоянию.

longitude = L2 – географическая долгота заданной точки в виде числа с точкой, используется в паре с latitudeпри сортировке по расстоянию.

### Структуры данных

Структура City

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID города |
| name | string | Да | Название города |
| alias | string | Нет | Код города |
| utcOffset | int | Да | Часовой пояс города в виде смещения от UTCпо зимнему времени |
| location | структура Location | Да | Координаты города |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Список станций метро

Доступ к справочной информации по станциям городского метро

### URI

**/cities/{id}/subway**

Обращение к коллекции станций метро в указанном городе

**/subway/{id}**

Обращение к элементу коллекции с id = {id}

### Методы

GET

Клиент запрашивает информацию о станциях метро

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры SubwayStation. При обращении к элементу коллекции – единичный объект SubwayStation.

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### МодификаторыURI

/cities/{id}/subway

sort = { name| distance } – сортирует коллекцию в алфавитном порядке либо по удаленности от заданных координат

rangeStart= X- осуществляет выборку, начиная с элемента X (нумерация с 0). Используется совместно с limit.

limit= Y. Ограничивает выборку количеством элементов Y.

latitude = L1 – географическая широта заданной точки в виде числа с точкой, используется в паре с longitudeпри сортировке по расстоянию.

longitude = L2 – географическая долгота заданной точки в виде числа с точкой, используется в паре с latitudeпри сортировке по расстоянию.

### Структуры данных

Структура SubwayStation

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID станции |
| name | string | Да | Название станции |
| line | String | Да | Название линии |
| color | String | Да | hex-код цвета линии (без ‘#’) |
| location | структура Location | Нет | Координаты станции |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Список киносетей

Получение списка киносетей

### URI

**/networks/**

Обращение к коллекции киносетей

**/networks/{id}**

Обращение к элементу коллекции с id = {id}

### Методы

GET

Клиент запрашивает информацию о киносетях

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры CinemaNetwork. При обращении к элементу коллекции – единичный объект CinemaNetwork.

404 Not Found, при обращении к элементу коллекции с несуществующим ID.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### МодификаторыURI

/networks

rangeStart= X- осуществляет выборку, начиная с элемента X (нумерация с 0). Используется совместно с limit.

limit= Y. Ограничивает выборку количеством элементов Y.

### Структуры данных

Структура CinemaNetwork

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID киносети |
| name | string | Да | Название киносети |
| isSale | boolean | Да | Есть продажа билетов |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Координаты кинотеатров

### URI

**/cinemas/geo**

Обращение к списку кинотеатров с координатами. (Является оптимизированным вариантом метода **/cinemas**, предназначенным для отрисовки кинотеатров на карте)

### Методы

GET

Клиент запрашивает список кинотеатровc координатами

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры CinemaGeoInfo.

POST, DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/cinemas/geo

sort = { name| distance} – сортирует коллекцию в алфавитном порядке | по удаленности.

city=city\_id - осуществляет выборку кинотеатров в указанном городе

network = network\_id – осуществляет выборку кинотеатров для указанной киносети.

rangeStart = N, ограничивает выборку, начиная с N-го элемента (0-based index), используется в паре с limit для постраничного отображения.

limit= Y. Ограничивает выборку количеством элементов Y.Используется в паре с rangeStart для постраничного отображения.

search = строка в кодировке UTF-8. Ограничивает выборку элементами, название которых содержит подстроку.

latitude = число. Географическая широта пользователя. Используется в паре с longitude при сортировке кинотеатров по удаленности.

longitude = число. Географическая долгота пользователя. Используется в паре с latitude при сортировке кинотеатров по удаленности.

subwayStation = IDстанции метро. Используется при сортировке по удаленности от метро.

subwayFilter = ID станции метро. Используется при фильтрации списка по метро.

isSale = {0|1} Флаг продаваемости билетов в кинотеатре

### Структуры данных

Структура CinemaGeoInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID кинотеатра |
| title | string | Да | Название кинотеатра |
| latitude | numeric | Да | Широта кинотеатра |
| longitude | numeric | Да | Долгота кинотеатра |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Список заказов

Получение списка заказов

### URI

/orders

Обращение к коллекции успешных заказов клиента

/orders/{id}

Обращение к элементу коллекции с id = {id}

### Методы

GET

Клиент запрашивает список заказов

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры Order. При обращении к элементу коллекции – единичный объект Order.

404 NotFound, при обращении к элементу коллекции с несуществующим ID.

POST

Клиент привязывает заказы к аккаунту на сайте

Ответ

200 OK в случае успеха.

DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

/orders

rangeStart = X- осуществляет выборку, начиная с элемента X (нумерация с 0). Используется совместно с limit.

limit = Y. Ограничивает выборку количеством элементов Y.

dateFrom – дата в формате DDMMYYYY ограничивающая список заказов снизу

dateTo – дата в формате DDMMYYYY ограничивающая список заказов сверху

### Структуры данных

Структура Order

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID заказа |
| ticket | string | Да | Номер билета |
| movie | string | Да | Название фильма |
| cinema | string | Да | Название кинотеатра |
| ctime | datetime | Да | Время продажи |
| startTime | datetime | Да | Фактическое время показа, с указанием смещения UTC. Это время может отличаться от даты показа. Например, показы с началом в 1 час ночи относятся к предыдущему дню, хотя фактически идут в следующие сутки. |
| barcodeToken | string | Нет | Уникальный токен для генерации QR-кода и pkpass-файла для PassBook |
| qrcodeURL | string | Нет | Ссылка на изображение qrcode’а билета |
| pkpassURL | string | Нет | Ссылка на PassBook файл |
| icsURL | string | Нет | Ссылка на ICS-файл для IPhone |
| vcsURL | string | Нет | Ссылка на VCS-файл для Android |
| placeCount | int | Да | Количество мест в заказе |
| poster | string | Нет | Имя файла постера |
| loyalties | Массив структур Loyalty | Нет | Программы лояльности, примененная к данному заказу |

### Поведение клиента

Описание не требуется

### Поведение сервера

Описание не требуется

# Добавление кинотеатра в список избранных

Метод требует предварительной «связки» моб.сессии с уч.записью на сайте «Киноход» с помощью метода /user/bind. В противном случае, возвращается ошибка с HTTP-статусом 412 и errorCode = 80

### URI

/user/cinemas/fav/add

Пометка кинотеатра как избранного для текущего пользователя приложения

### Методы

POST

Пометка кинотеатра как избранного для текущего пользователя приложения

Клиент передает в теле структуру UserCinemasFavRequest

Ответ

200 OK/ 201 Created в случае успеха.

GET, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура UserCinemasFavRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| cinemas | Array | Да | Массив ID кинотеатров |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Удаление кинотеатра из списка избранных

Метод требует предварительной «связки» моб.сессии с уч.записью на сайте «Киноход» с помощью метода /user/bind. В противном случае, возвращается ошибка с HTTP-статусом 412 и errorCode = 80

### URI

/user/cinemas/fav/delete

Удаление пометки избранного у кинотеатров для текущего пользователя приложения

### Метод

POST

Удаление пометки избранного у кинотеатров для текущего пользователя приложения

Клиент передает в теле структуру UserCinemasFavRequest

Ответ

200 OK/ 201 Created в случае успеха.

GET, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура UserCinemasFavRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| cinemas | Array | Да | Массив ID кинотеатров |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Программы лояльности

### URI

/loyalties/

Обращение к коллекции доступных программ лояльности

### Методы

1. GET

Клиент запрашивает список программ лояльности

Ответ

200 OK в случае успеха.

При обращении к коллекции возвращается массив JSON-объектов структуры Loyalty.

PUT, DELETE, POST

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура Loyalty

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID программы лояльности |
| alias | string | Да | Алиас программы лояльности |
| name | string | Да | Имя программы лояльности |
| currency | Currency | Нет | Валюта программы лояльности |

Структура Currency

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID валюты |
| name | string | Да | Имя валюты (просклоненное) |
| name\_plural | string | Да | Имя валюты во множественном числе |
| rate | double | Да | Курс |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Расчет суммы баллов для программы лояльности

### URI

/loyalty/{id}/calc

POST

Клиент запрашивает число баллов получаемых по программе лояльности

Тело запроса

JSON-объект структуры LoyaltyCalcRequest

Ответ

200 OK и JSON-объект структуры LoyaltyCalcResponse, в случае успеха.

404 Not Found в случае если программа лояльности не найдена

GET, DELETE, POST

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура LoyaltyCalcRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| sum | double | Да | Сумма заказа |

Структура LoyaltyCalcResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| points | int | Да | Число очков по программе лояльности |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Проверка номеров карт программ лояльности

### URI

/loyalty/card\_check

POST

Клиент запрашивает проверку номера карты программы лояльности

Тело запроса

Клиент передает в теле структуру LoyaltyCardCheckRequest

Ответ

200 OK и JSON-объект структуры LoyaltyCardCheckResponse

GET, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура LoyaltyCardCheckRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| loyalty\_id | int | Да | ID программы лояльности |
| cardnumber | string | Да | Номер карты для проверки |

Структура LoyaltyCardCheckResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| result | double | Да | Результат проверки номера карты |

Значения result

|  |  |
| --- | --- |
| 0 | Номер неправильный |
| 1 | Номер правильный |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Привязка заказа к программе лояльности

### URI

/orders/{order\_id}/loyalty/{loyalty\_id}

POST

Клиент привязывает заказ к программе лояльности

Тело запроса

Клиент передает в теле структуру OrderLoyaltyBindRequest

Ответ

200 OK и JSON-объект структуры OrderLoyaltyBindResponse

404 Not Found – При неверных order\_id или loyalty\_id

500 Internal Error – При прочих ошибках

GET, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура OrderLoyaltyBindRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| token | string | Да | Номер карты |

Структура OrderLoyaltyBindResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| result | int | Да | Результат привязки номера карты |

Значения result

|  |  |
| --- | --- |
| 1 | Успешно |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Профиль пользователя

### URI

/user/profile

Обращение за профилем пользователя

### Методы

1. GET

Клиент запрашивает профиль пользователя

Ответ

200 OK в случае успеха.

При обращении возвращается структура UserProfile

PUT, DELETE, POST

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура UserProfile

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| name | string | Нет | Имя пользователя |
| username | string | Нет | Никнейм |
| phone | string | Нет | Телефон |
| email | string | Нет | Эл.адрес |
| sex | string | Нет | Пол |
| birthdate | string | Нет | День рождения |
| avatar | string | Нет | Аватар |
| loyalties | Массив структур ProfileLoyalty | Да | Данные программ лояльности |
|  |  |  |  |

Структура ProfileLoyalty

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID программы лояльности |
| alias | string | Да | Алиас программы лояльности |
| name | string | Да | Имя программы лояльности |
| token | string | Да | Номер карты |

URL аватара формируется следующим образом:  
http(s)://<HOST>/p/<w>x<h>/<ab>/<cd>/<uuid>.<ext>

Где:

* <HOST> -- конфигурируемое значение для хоста с изображениями. На текущий момент такой же, как хост для API;
* <w> и <h> -- ширина и высота изображения соответственно. При несовпадении пропорций по сравнению с оригинальным изображением, оно центрируется, пропорционально масштабируется и обрезается до нужных размеров.
* <ab> и <cd> -- первые 4 символа <uuid>

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Профиль пользователя – изменения номеров карт программ лояльность

### URI

/user/profile/loyalty

Пользователь изменяет номер карты указанной программы лояльности

### Методы

1. POST

Пользователь изменяет номер карты указанной программы лояльности

Тело запроса

Клиент передает в теле структуру ProfileLoyaltyRequest

Ответ

200 OK в случае успеха.

GET, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура ProfileLoayltyRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| loyalty\_id | int | Да | ID программы лояльности |
| token | string | Да | Номер карты пр.лояльности |

# Профиль пользователя – Привязка заказов к Удаление кинотеатра из списка избранных

Метод требует предварительной «связки» моб.сессии с уч.записью на сайте «Киноход» с помощью метода /user/bind. В противном случае, возвращается ошибка с HTTP-статусом 412 и errorCode = 80

### URI

/user/orders

Удаление пометки избранного у кинотеатров для текущего пользователя приложения

### Метод

POST

Удаление пометки избранного у кинотеатров для текущего пользователя приложения

Клиент передает в теле структуру UserOrdersRequest

Ответ

200 OK/ 201 Created в случае успеха.

GET, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура UserOrdersRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| orders | Array | Да | Массив ID заказов |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Отмена заказа клиентом

### URI

/cancel

Проверка данных заказа. Возвращает token для отмены заказа

/cancel\_order

Отмена заказа

### Метод

/cancel

GET

Клиент отправляет данные для проверки данных заказа.

Клиент передает в теле структуру OrderCheckRequest

Ответ

200 OK и JSON-объект структуры OrderCheckResponse

POST, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

/cancel\_order

GET

Клиент отменяет заказ.

Клиент передает в теле структуру OrderCancelRequest

Ответ

200 OK и JSON-объект структуры OrderCancelResponse

POST, PUT, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура OrderCheckRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| order | Int | Да | Внешний id заказа. |
| phone | String | Нет | Телефон клиента, указанный при заказе билетов. Поле обязательно к заполнению, если телефон был указан. Телефон указывается только перечислением цифрового ряда, без вспомогательных символов и пробелов. |
| email | String | Нет | Email клиента, указанный при заказе билетов. Поле обязательно к заполнению, если email был указан. |

Структура OrderCheckResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| token | String | Нет | Token для отмены заказа. Возвращается в случае успешной проверки данных заказа. Формат XXXX:YYYYYYY, где  XXXX - это числовое значение произвольной длины,  YYYYYYY - вспомогательный ключ произвольной длины, может содержать в себе числовые и буквенные символы. |
| error | Bool | Нет | Вспомогательное поле, возвращается в случае, если проверка заказа завершилась ошибкой. Может иметь значение 1 или 0. |
| error\_str | String | Нет | Поле возвращается в случае, если проверка заказа завершилась ошибкой. Содержит текст ошибки. |

Структура OrderCancelRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| token | String | Да | Ключ для отмены заказа |

Структура OrderCancelResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| data | Int | Нет | Статус отмены заказа. Содержит 0 или 1, 1 - если отмена прошла успешно. 0 - если отмена не удалась. |
| error\_str | String | Нет | Поле возвращается в случае, если отмена заказа завершилась ошибкой. Содержит текст ошибки. |
| error | Bool | Да | Принимает значение 1 или 0, в зависимости от того, возникли ошибки в процессе выполнения отмены или нет. |

### Поведение клиента

Клиент может самостоятельно отменить свой заказ.

Стандартный алгоритм: выполняется запрос “cancel” для проверки данных, если выполнение прошло успешно - клиенту возвращается token. Используя token, клиент отменяет свой заказ запросом "cancel\_order".

### Поведение сервера

Описание не требуется.

# Связка сессии мобильного приложения с учетной записью сайта «Киноход»

### URI

/user/bind

POST

Клиент связывает сессию мобильного приложения с учетной записью сайта «Киноход»

Тело запроса

Клиент передает в теле структуру UserBindRequest

Ответ

200 OK и JSON-объект структуры UserBindResponse

500 Internal Error – При прочих ошибках

GET, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Структуры данных

Структура OrderLoyaltyBindRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| user\_token | string | Да | Зашифрованный user\_id |

Структура UserBindResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| user\_id | int | Да | Id аккаунта пользователя на сайте |

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

# Отвязка сессии мобильного приложения от учетной записью сайта «Киноход»

### URI

/user/unbind

GET

Клиент отвязывает сессию мобильного приложения от учетной записью сайта «Киноход»

Ответ

200 OK и JSON-объект структуры UserBindResponse

500 Internal Error – При прочих ошибках

POST, GET, DELETE

Не поддерживаются. Сервер возвращает ответ 405.

### Поведение клиента

Описание не требуется.

### Поведение сервера

Описание не требуется.

## Приложение 1

# Аутентификация приложения

## Управление сессиями

Сервер ведет учет действий клиента в статистических и прочих целях. Реализация этого механизма основана на идентификаторе сессии, который клиент получает в результате *регистрации*. Под регистрацией в данном контексте понимается регистрация клиентского приложения на сервере API, и не связана с регистрацией и авторизацией пользователей.

Клиент получает идентификатор сессии через механизм cookies и должен возвращать его на сервер в каждом последующем запросе.

Если идентификатор сессии отсутствует в cookiesлюбого запроса, то сервер возвращает ответ 403 Forbidden. В теле ответа присутствует описание ошибки в виде структуры ErrorDesc.

# Регистрация клиента

Все клиенты проходят первоначальную регистрацию. Регистрация используется для отслеживания поведения клиентов (через механизм сессий), сбора статистики по устройствам, интенсивности использования приложения пользователей и т.п. Также во время регистрации клиенты получают от сервера базовые данные о системе, настройках, и пр.

### URI

**/registration**

### Методы

POST

Клиент запрашивает регистрацию.

Тело запроса

JSON-объект структуры RegistrationRequest

Ответ

200 OK и JSON-объект структуры RegistrationResponse, в случае успеха.

GET

Клиент запрашивает регистрацию. Метод GET используется только для возможности регистрации из браузера на тестовом сервере. См. примечание ниже.

Ответ

200 OK и JSON-объект структуры RegistrationResponse, в случае успеха.

405 – на боевом сервере.

DELETE, PUT

Не поддерживаются. Сервер возвращает ответ 405.

### Модификаторы URI

Не применяются

### Структуры данных

Структура RegistrationRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| deviceId | string | Да | Уникальный идентификатор устройства |
| deviceOs | string | Да | Название операционной системы устройства |
| deviceOsVer | string | Да | Версия ОС устройства |
| deviceName | string | Да | Модель и марка устройства |
| imei | string | Нет | IMEI устройства |
| apikey | string | Да | Серверный ключ партнера, идентифицирующий приложение |
| appVer | int | Да | Версия приложения |
| apiVer | int | Да | Версия API, используемая данным приложением |
| language | string | Да | Язык для пользователя. Допустимые значения:  RU – русский |
| location | СтруктураLocationInfo | Нет | Текущее местоположение устройства |
| city | int | Нет | Город по умолчанию (из настроек приложения) |

Структура LocationInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| latitude | decimal | Да | Географическая широта |
| longitude | decimal | Да | Географическая долгота |

Структура RegistrationResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| registrationId | string | Да | Идентификатор сессии |
| useHttps | boolean | Да | Клиент должен использовать безопасное соединение |
| authRequired | boolean | Да | Аутентификация пользователей обязательна |
| casServerUri | string | Нет | URLсервера CASдля аутентификации |
| eula | string | Да | URLтекстового документа с пользовательским соглашением |
| supportPhone | string | Да | Телефон службы поддержки сервиса |
| webAddress | string | Да | Адрес сайта системы |
| nearestCity | Структура CitySearch | Нет | Ближайший город |
| upgrade | Структура Upgrade | Нет | Информация о наличии новой версии приложения. |

Структура CitySearch

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| city | Структура City | Да | Город |
| proximity | numeric | Да | Расстояние до ближайшего города в километрах |

Структура City

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| id | int | Да | ID города |
| name | string | Да | Название города |
| utcOffset | int | Да | Часовой пояс города (в виде смещения в часах относительно UTC, без учета летнего времени) |

Структура Upgrade

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обяз. | Комментарий |
| version | int | Да | Версия доступного нового приложения (число) |
| versionName | string | Да | Имя версии нового доступного приложения (строка «1.2.44.1234») |
| mandatory | boolean | Да | Обновление обязательно. |

### Поведение клиента

Клиент начинает взаимодействие с сервером путем регистрации на сервере API. В момент регистрации клиент сообщает серверу всю информацию об устройстве, версии операционной системы и версии самого приложения. По возможности, клиент так же указывает свое текущее местоположение и город (id), если он известен приложению из предыдущих настроек.

В любой момент времени, на любой запрос, клиент может получить ответ 403 с индикацией того, что сервер требует повторную регистрацию.

Все полученные cookiesклиент должен сохранить и возвращать серверу

### Поведение сервера

Получив запрос на регистрацию, сервер генерирует идентификатор сессии, вычисляет ближайший город(город ближайшего кинотеатра).

Если параметры запроса некорректны, сервер возвращает 400 Bad Request.

Сервер присылает cookies клиенту с неустановленным сроком жизни:

* reg=идентификатор сессии
* city = IDгорода местоположения. Этот параметр равен либо city из запроса на регистрацию, если указан; ближайший город, если известен из местоположения пользователя; либо Москва.

Сервер может обрабатывать запрос методом GET исключительно в тестовых и отладочных целях. В этом случае сервер возвращает произвольные данные

# Авторизация мобильного приложения с учетной записью «Киноход»

На данный момент поддерживается авторизация через три основные социальные сети:

|  |  |
| --- | --- |
| Провайдер авторизации | social alias |
| Facebook | facebook |
| ВКонтакте | vkontakte |
| Одноклассники | odnoklassniki |

close\_url – http://kinohod.ru/auth/closeurl

Для авторизации необходимо перейти по url: http://kinohod.ru/auth/login/{social\_alias}/?retpath={close\_url}

После авторизации, социальная сеть «перебросит» по адресу {close\_url}.

В случае успеха, в HTTP-заголовках страницы будет содержаться параметр X-User\_ID, значение которого необходимо использовать в методе /user/bind для привязки пользовательского идентификатора к сессии мобильного приложения

## Приложение 2

# Аутентификация партнера

Все методы API подразделяются на две группы, Клиентские методы, вызов которых может совершаться клиентом с использованием браузера, и Серверные методы, вызов которых должен осуществляться с сервера партнера. На текущий момент, если не указано иное, все методы API относятся к Серверным методам.

Всем партнерам, пользующимся Партнёрским API выдаётся URL для доступа к API и два уникальных секретных ключа: ключ для запросов на клиентские и ключ для запросов на серверные методы.

При каждом запросе к методам API(как Клиентским, так и Серверным),Партнером должен передаваться GET-параметр apikey, равный соответствующему ключу.

https://<HOST>/api/rest/partner/<version>/<method>?apikey=<apikey>

При этом параметр<apikey> однозначно идентифицирует Партнера.

Пример запроса:

https://<HOST>/api/rest/partner/v1/movies?apikey=8e62f688-91b2-11e0-a950-bcaec529f916