

REST API v1 для мобильных клиентов

Введение

[Локализация значений справочников](#)

[Миниатюры изображений](#)

[Загрузка фотографий](#)

Версии API

[1.17 Кастинг без автомодерации](#)

[1.16 Карантин](#)

[1.15 Приватные базы](#)

[1.14 Добавлено получения типичных вопросов ответов \(FAQ\)](#)

[1.13 Добавлен выбор типа кастинга при создании](#)

[1.10 Добавлены новые поля в настройках уведомлений](#)

[1.9. Добавлен тип кастинга: Заявка на WhatsApp](#)

[1.8. Добавлены наименования незаполненных полей в оповещении о невозможности активации тестового периода](#)

[1.7. Добавлена специализация “модель” \(актуальная версия\)](#)

[1.6. Множественный выбор специализаций кастинга и прочее](#)

[1.5. Обновленный формат передачи версии приложения](#)

[1.4 - Бюст, игровой возраст, жалобы и прочее](#)

[1.3 - Кастинги с заявками на почту](#)

[1.2 - Размерность роста](#)

[1.1 - Версионирование в API](#)

[1.0 Первоначальная версия.](#)

1. Общие

[1.1. Получить страны](#)

[1.2. Получить страну по идентификатору](#)

[1.3. Получить города страны](#)

[1.4. Получить город по идентификатору](#)

[1.5. Отправить предложение по улучшению](#)

[1.6. Получить страну по названию](#)

[1.7. Получить город по названию](#)

[1.8. Получить ссылку на страницу покупки подписки](#)

[1.8. Получить возможные значения параметра “Тип внешности”](#)

2. Аккаунт

[2.1. Отправить SMS код](#)

[2.2 Авторизация с использованием SMS кода](#)

- [2.3 Авторизация по соц. сетям](#)
- [2.4. Получить мой профиль](#)
- [2.5. Сохранить мой профиль](#)
- [2.6. Получить счётчики](#)
- [2.7. Проверить правильность токена доступа](#)
- [2.8. Обновить токен регистрации FCM](#)
- [2.9. Получить список профилей пользователя в базах партнеров](#)
- [2.10. Получить профиль по партнеру](#)

3. Фотографии

- [3.1. Сохранить фото](#)
- [3.2. Удалить фото](#)
- [3.3. Получить фото по id](#)
- [3.4. Получить все фотографии пользователя](#)
- [3.5. Получить фотографии профиля](#)
- [3.6. Добавить фотографию в профиль](#)
- [3.7. Удалить фотографию из профиля](#)
- [3.8. Получить адрес сервера для загрузки фото](#)
- [3.9. Сохранить фото на фотосервер](#)

4. Видео

- [4.1. Сохранить видео](#)
- [4.2. Удалить видео](#)
- [4.3. Получить видео](#)
- [4.4. Получить все видео пользователя](#)
- [4.5. Получить видео профиля](#)
- [4.6. Добавить видео в профиль](#)
- [4.7. Удалить видео из профиля](#)

5. Пользователи

- [5.1. Получить количество пользователей по профессиям](#)
- [5.2. Поиск пользователей](#)
- [5.3. Получить информацию пользователя](#)
- [5.4. Поставить лайк пользователю](#)
- [5.5. Убрать лайк у пользователя](#)
- [5.6. Дать награду пользователю](#)
- [5.7. Забрать награду пользователя](#)
- [5.8. Получить количество в избранном по профессиям](#)
- [5.9. Получить избранное](#)
- [5.10. Добавить в избранное](#)
- [5.11. Удалить из избранного](#)
- [5.12. Запросить открытие контактных данных](#)
- [5.13. Открыть контакты](#)
- [5.14. Не открывать контакты](#)

[5.15. Создать жалобу на пользователя](#)

[5.16. Добавить отзыв](#)

[5.17. Редактировать отзыв](#)

[5.18. Получить отзывы](#)

[5.19. Удалить отзыв](#)

[6. Вакансии](#)

[6.1. Получить список вакансий](#)

[6.2. Получить информацию по вакансии](#)

[6.3. Создать вакансию](#)

[6.4. Редактировать вакансию](#)

[6.5. Удалить вакансию](#)

[6.6. Получить мои вакансии](#)

[6.7. Получить информацию о вакансии как создатель](#)

[6.8. Подать заявку на вакансию](#)

[6.9. Получить список предложений](#)

[6.10. Создать жалобу на вакансию](#)

[7. Кастинги](#)

[7.1. Получить кастинги](#)

[7.2. Получить информацию о кастинге](#)

[7.3. Создать кастинг](#)

[7.4. Редактировать кастинг](#)

[7.5. Удалить кастинг](#)

[7.6. Получить мои кастинги](#)

[7.7. Получить инфу о кастинге как создатель](#)

[7.8. Подать заявку на кастинг](#)

[7.9. Получить список предложений](#)

[7.10. Получить список одобренных](#)

[7.11. Принятие решения на отклик](#)

[7.12. Создать жалобу на кастинг](#)

[7.13. Получить html код email рассылки](#)

[7.14. Отменить автомодерацию кастинга](#)

[8. Оповещения](#)

[8.1. Получить оповещения](#)

[8.2. Пометить как прочитанное](#)

[8.3. Получить настройки оповещений](#)

[8.4. Сохранить настройки оповещений](#)

[8.5. Отправить тестовое оповещение](#)

[8.6. Получить оповещение по chain_id](#)

[9. Покупки](#)

[9.1. Купить подписку](#)

[9.2. Купить открытие контактов](#)

[9.3. Получить доступные для покупки подписки](#)

[10. Html анкета](#)

[10.1. Сохранить фото для html анкеты](#)

[10.2. Получить фото для html](#)

[10.3. Получить html своего профиля](#)

[11. FAQ Типичные вопросы и ответы](#)

[11.2 Получить список вопросов в заданной кате](#)

[Push уведомления](#)

[Общие коды ошибок](#)

[Перечисления](#)

[Профессии](#)

[Цвет волос](#)

[Цвет глаз](#)

[Размер одежды](#)

[Вождение](#)

[Вокал](#)

[Языки](#)

[Бюст](#)

[Типы проектов кастингов](#)

[Типы оповещений](#)

[Типы подписок](#)

[Типы кастингов](#)

[Типы вакансий](#)

[Типы ошибок активации тестового периода](#)

[Незаполненные дополнительные параметры](#)

[Роли пользователей](#)

[Статусы пользователей](#)

[Пол](#)

Введение

Запросы осуществляются по следующему адресу:

https://api.cinarta.com/v1/METHOD_NAME?PARAMETERS&token=TOKEN

Где:

- METHOD_NAME (обязательно) - название метода API;
- PARAMETERS (опционально) - входные параметры соответствующего метода API, последовательность пар name=value, разделенных амперсандом.
- TOKEN (опционально) - ключ доступа, требуется или нет указано в описании запроса;

Для каждого запроса нужно указывать следующий заголовок:

- X-API-Version - версия API используемая приложением в формате: x.y;
- X-App-Version - версия приложения в формате x.y.z, где x - major версия, y - minor версия, а z - версия сборки. Версию приложения нужно извлекать программно:
 - ios: [How can my iphone app detect its own version number?](#);
 - android: [How to get the build/version number of your Android application?](#);

В случае успеха выполнения возвращаемое значение помещается в response.

В случае ошибки вернется следующий ответ:

```
error: {  
    error_code: 1,  
    error_msg: "Error message."  
}
```

Все даты передаются и получаются в виде unix timestamp.

Локализация значений справочников

Для получения справочных данных (название стран, городов) на необходимом языке нужно задавать заголовок "Accept-Language" в запросе.

Сейчас поддерживаются значения:

- en-US - английский, значение по умолчанию;
- ru-RU - русский;

Миниатюры изображений

Для получения миниатюры изображения нужно модифицировать ссылку на изображение. Миниатюра сохраняет пропорции исходного изображения.

Правило формирования ссылки

Имеем исходное изображение в формате:

<http://api.cinarta.com/path-to-image>

Ссылка на миниатюру будет иметь вид:

<http://api.cinarta.com/image/%w%x%h%/path-to-image>, где %w% - ширина требуемой миниатюры, а %h% высота требуемой миниатюры.

Пример

Исходная ссылка: <http://api.cinarta.com/uploads/fixture/photo/7.jpg>

Ссылка на миниатюру: <http://api.cinarta.com/image/200x400/uploads/fixture/photo/7.jpg>

Загрузка фотографий

С версии 1.8 внедрено распределенное хранение фотографий.

Методы получения фотографий остались прежними, ссылки фотографий не всегда будут вести на основной сервер (api.cinarta.com).

Метод загрузки фотографий несколько изменился.

1. Нужно получить сервер для загрузки при помощи метода [3.8. Получить адрес сервера для загрузки фото](#).
2. Загрузить фотографию. [3.9. Сохранить фото на фотосервер](#). Фотографии загружаются по адресу: PHOTO_SERVER?token=TOKEN, где
 - PHOTO_SERVER - адрес сервера для загрузки фотографии.
 - TOKEN - ключ доступа;

Метод возвращает ссылку на фотографию и id.

3. Методы, где в параметрах передается id фотографии, не изменились.

Версии API

1.17 Кастинг без автомодерации

- В методы “7.3. Создать кастинг” и “7.4. Редактировать кастинг” добавлены параметры:
 - `moderation_on` - включить модерацию кастинга. Если передан 1, то кастинг будет проверен модератором, если 0 - то будет работать автомодерация. По умолчанию 0. Параметр только для проверенных кастинг-директоров.
 - `reason_for_moderation` - описание, для чего кастинг-директор, хочет, чтобы кастинг проверил модератор, не обязательный. Параметр только для проверенных кастинг-директоров.

1.16 Карантин

- В метод “7.2. Получить информацию о кастинге” добавлено поле `can_send_apply`. Если `true`, то подача заявок разрешена, если `false` - то запрещена.

1.15 Приватные базы

- Добавлен метод “2.9. Получить список профилей пользователя в базах партнеров”.
- Добавлен метод “2.10. Получить профиль по партнеру”.
- В метод “3.3. Получить фото по id” добавлена ошибка “331 - фотография не найдена”
- В метод “3.5. Получить фотографии профиля” добавлена ошибка “351 - передан несуществующий идентификатор пользователя”
- В метод “4.3. Получить видео” добавлена ошибка “431 - видео не найдено”.
- В метод “4.5. Получить видео профиля” текст ошибки “451 - нет доступа к видео пользователя” исправлен на “передан несуществующий идентификатор пользователя”.
- Добавлено получение поля “`type`” в методе “7.1. Получить кастинги”.
- Добавлен статус пользователя “11- частичная регистрация”. Устанавливается пользователю, если он создан при создании пользователя партнера. Для него не отображаются вакансии, отображаются только кастинги партнеров, профиль не выдается в списке пользователей. Меняется на “10 - зарегистрирован”, когда пользователь заполнит все необходимые данные (загрузит аватар).
- В метод “5.3. Получить информацию пользователя” добавлена ошибка “531 - пользователь не найден”.
- Добавлена настройка оповещений “`only_partners_casting`” - “Получать кастинги только от приватных баз”. Если параметр включен, и для пользователя есть связанные с ним пользователи партнеров, то оповещения приходят только о

кастингах партнеров. Если нет связанных пользователей партнеров, то параметр не учитывается при рассылке оповещений.

- В метод "8.3. Получить настройки оповещений" добавлено поле "only_partners_casting". Поле приходит в ответе, только если есть связанные с пользователем пользователи партнеров.
- В метод "8.4. Сохранить настройки оповещений" добавлен параметр "only_partners_casting". Параметр сохраняется, только если есть связанные с пользователем пользователи партнеров (Т.е. если метод 8.3. возвращает это поле).
- Добавлен тип проектов кастинга "Фотосессия".
- Добавлена настройка оповещений "photosession" - "фотосессия". Если параметр включен, для пользователя будут приходить оповещения о кастингах с типом проекта "фотосессия".
 - В метод "8.3. Получить настройки оповещений" добавлено поле "photosession".
 - В метод "8.4. Сохранить настройки оповещений" добавлен параметр "photosession".

1.14 Добавлено получения типичных вопросов ответов (FAQ)

- Добавлены методы `faq.getCategories` и `faq.getByCategory` для получения списка категорий вопросов и списка вопросов относящихся к определенной категории.
- Убрана проверка наличия подписки при создании кастингов.

1.13 Добавлен выбор типа кастинга при создании

- В методы `casting.create` и `casting.update` добавлен параметр `type` - "тип кастинга" и дополнительные параметры, задаваемые в зависимости от выбранного типа:
 - 1 - обычный
 - 2 - заявки только на email:
 - `requests_email` - email для отправки заявок;
 - `requests_email_subject` - тема письма для заявки;
 - 3 - заявки только по whatsapp:
 - `whatsapp_phone` - номер whatsapp;
 - `photo_description` - описание для фотографий.
- Добавлен метод удаления отзыва (5.19) `user.deleteReview`.
- Добавлено получение роли пользователя в методы авторизации.
- В метод "7.2. Получить информацию о кастинге" добавлена возможность получить кастинг с истекшим сроком подачи заявок.
- Для метода "7.8. Подать заявку на кастинг" добавлена ошибка "786 - окончился срок подачи заявок на кастинг".
- В метод "7.7. Получить инфу о кастинге как создатель" добавлен вывод параметра `"is_on_auto_moderation"`, обозначающий, находится ли кастинг на автомодерации.

- В метод “8.2. Пометить как прочитанное” добавлена возможность пометить все сообщения.

1.10 Добавлены новые поля в настройках уведомлений

- В метод `notification.GetSettings` добавлены поля:
 - `email_about_purchase` - Email оповещение о покупках;
 - `sms_about_purchase` - SMS оповещение о покупках;
 - `push_about_purchase` - Push оповещение о покупках;
- Добавлен метод `info.getAppearanceTypeValues`, возвращающий возможные значения характеристики “Тип внешности”;
- В профиле пользователя и в кастинге появилось дополнительное поле `ex_appearance_type`, хранящее настройку характеристики “Тип внешности” для целевых специализаций (см. Профессии);
- Добавлена новая ошибка при попытке отредактировать существующий кастинг 744 - Кастинг находится на автомодерации.
- Добавлен новый метод `casting.cancelAutoModeration` (7.14) для отмены автомодерации кастинга.

1.9. Добавлен тип кастинга: Заявка на WhatsApp

- В перечисление Типы кастингов добавлен тип: “отправка заявок только на whatsapp”;
- В метод `casting.get` добавлены поля:
 - `whatsapp_phone` - телефон для отправки заявки по WhatsApp;
 - `photo_description` - описание фотографий;
- Добавлены оповещения при отключении подписки

1.8. Добавлены наименования незаполненных полей в оповещении о невозможности активации тестового периода

- В оповещение “16 - тестовая подписка не активирована” добавлен параметр *missingParams* - массив с наименованиями незаполненных дополнительных параметров.
- Добавлено перечисление: *Незаполненные дополнительные параметры*.
- Добавлен метод 3.8. *Получить адреса сервера для загрузки фото*.
- Для загрузки фотографий больше не используется метод 3.1. *Сохранить фото*. Используется метод 3.9. *Сохранить фото на фотосервер*.
- Добавлен метод 9.3. *Получить доступные для покупки подписки*.
- В перечисление *Типы подписок* добавлены идентификаторы подписок:
 - на 3 месяца;
 - на 6 месяцев.

1.7. Добавлена специализация “модель” (актуальная версия)

- Добавлена специализация “модель”.
- Для профиля пользователя добавлен параметр “готов(а) к НЮ кастингам” (только для специализации “модель”)
- Для кастингов добавлены параметры:
 - is_nude - кастинг в стиле НЮ;
 - casting_director_status - статус кастинг директора;
 - warning_text - предупреждающий текст;
- Добавлены типы проектов кастингов:
 - Проморолик;
 - Соц ролик;
 - Театральная постановка;
- Добавлены настройки оповещений для новых типов проектов кастингов:
 - promo_clip;
 - social_video;
 - theatrical_performance;
- Добавлены типы вакансий:
 - Обычная;
 - Создана вне системы. Нельзя подать заявку на вакансию такого типа. Из мобильного приложения тип не задается;
- Добавлен параметр настроек оповещений ready_for_business_trips - готов к командировкам. Если параметр включен, оповещения приходят о кастингах с любых городов. Если выключен, то только для города пользователя.
- Добавлен идентификатор оповещения в push (chainId):
 - в push добавлен параметр chainId;
 - добавлен метод получения оповещения по chain_id: 8.6. *Получить оповещение по chain_id*;
 - в метод 8.1. *Получить оповещения* добавлен chain_id;
- Добавлен новый тип оповещений:
 - 16 - тестовая подписка не активирована
- Добавлено перечисление: *Типы ошибок активации тестового периода*,

1.6. Множественный выбор специализаций кастинга и прочее

- Для кастингов возможно несколько значений специализаций. Вместо параметра profession_id нужно передавать массив параметров professions.
- Добавлен метод info.getAdditionalInfo. Возвращает ссылку на страницу покупки подписки.
- Добавлены новые типы оповещений:
 - 61 - подписка будет автоматически продлена через 3 дня
 - 62 - подписка автоматически продлена

1.5. Обновленный формат передачи версии приложения

- Обновлён формат версии приложения. Версия приложения должна быть в формате x.y.z, где x - major версия, y - minor версия, а z - версия сборки.
- Версию приложения нужно извлекать программно:
 - ios: [How can my iphone app detect its own version number?](#);
 - android: [How to get the build/version number of your Android application?](#);
- В метод 7.2. *Получить информацию о кастинге (casting.get)* добавлен параметр `apply_submitted`, который отвечает за то, была ли уже подана текущем пользователем заявка на кастинг.
- Для оповещений “40 - личное сообщение” и “41 - массовое сообщение” в параметры добавлено поле “html” - сообщение в виде html-кода.
- Для кастингов добавлены параметры:
 - `cost` - цена (в рублях);
 - `not_commercial` - некоммерческий кастинг;
- Для настроек оповещений добавлены параметры:
 - `cost_from` - цена от - для специализаций "Актер, Актер любитель, Каскадер";
 - `cost_from_for_crowd_artist` - цена от - для специализации "Актер массовых сцен";
 - `without_cost` - получать оповещения о кастингах, в которых не указана цена;
 - `not_commercial` - получать оповещения о некоммерческих кастингах;

1.4 - Бюст, игровой возраст, жалобы и прочее

- добавлены новые параметры:
 - параметры:
 - бюст (значения в перечислении *Бюст*);
 - игровой возраст;
 - для каких профессий доступен и название параметров смотри в перечислении *Профессии*;
 - методы работу с которыми нужно доработать:
 - 2.4. Получить мой профиль - получать новые параметры;
 - 2.5. Сохранить мой профиль - сохранять новые параметры;
 - 5.2. Поиск пользователей - поиск по бюсту, и флаг учитывать или нет игровой возраст;
 - 5.3. Получить информацию пользователя - получать новые параметры;
 - 7.2. Получить информацию о кастинге;
 - 7.3. Создать кастинг;
 - 7.4. Редактировать кастинг;
 - 7.7. Получить инфу о кастинге как создатель;
- добавлены “Отзывы кастинг директоров”, методы по ним:
 - 5.16. Добавить отзыв;

- 5.17. Редактировать отзыв;
- 5.18. Получить отзывы;
- убраны награды повсеместно;
- добавлены жалобы на пользователей, кастинги, вакансии, новые методы:
 - 5.15. Создать жалобу на пользователя
 - 6.10. Создать жалобу на вакансию
 - 7.12. Создать жалобу на кастинг
- изменена логика работы со своими кастингами и вакансиями:
 - убраны все поля с префиксом "pub";
 - убрано поле is_published, теперь логика завязана только на поле review_status;
- отключена поддержка старых версий (до 1.3), обязательна передача версии 1.4;

1.3 - Кастинги с заявками на почту

Внесенные изменения:

- в метод casting.get добавлены поля:
 - type - тип кастинга, смотри возможные значения в перечислении "Типы кастингов";
 - requests_email - email для отправки заявок;
 - requests_email_subject - тема email письма для отправки заявки;
- добавлено перечисление "Типы кастингов";
- в метод casting.apply добавлен код ошибки: 785 - на кастинги с типом "отправка заявок только на email" нельзя подавать заявки в приложении;
- введены методы по управлению html профилем:
 - 10.1. Сохранить фото для html анкеты (html.savePhotos)
 - 10.2. Получить фото для html (html.getPhotos)
 - 10.3. Получить html своего профиля (html.getProfile)

1.2 - Размерность роста

Добавлена размерность для роста: сантиметры/футы.

Затрагивает следующие методы:

- account.getProfileInfo - в ответ приходит:
 - ex_height - рост в сантиметрах;
 - ex_height_ft - рост в футах;
 - ex_height_type - размерность роста, которую выбрал пользователь при сохранении профиля;
- account.saveProfileInfo - можно передавать одновременно два поля ex_height (значение в сантиметрах) и ex_height_ft (значение в футах), в зависимости от переданного ex_height_type будет браться соответствующее поле в качестве исходного, а второе пересчитываться;
- user.search, user.getFavourite - поиск можно делать как в футах так и в сантиметрах, для значений используются поля ex_height_from и ex_height_to, тип указывается в поле ex_height_type;
- user.get - в ответ приходят поля:

- ex_height - рост в сантиметрах;
- ex_height_ft - рост в футах;
- ex_height_type - размерность роста, которую выбрал пользователь при сохранении профиля;
- casting.get, casting.getAsAuthor - в ответ приходят поля:
 - ex_height_from - значение от в сантиметрах;
 - ex_height_to - значение до в сантиметрах;
 - ex_height_ft_from - значение от в футах;
 - ex_height_ft_to - значение до в футах;
- casting.create, casting.edit - для указания размерности роста переданной в полях: ex_height_from и ex_height_to нужно передавать поле ex_height_type;
- casting.getApproved, casting.getOffers - добавлены в ответ:
 - height_ft - рост в футах;
 - height_type - размерность роста;

1.1 - Версионирование в API

- Внедрены заголовки для версионирования (описаны в введении):
 - X-API-Version - версия API используемая приложением в формате: x.y;
 - X-App-Version - версия приложения, целое число, нужно получать программно из приложения;
- Внедрена выдача ошибки с кодом 5 (API устарел, необходимо обновить приложение), описано в общих кодах ошибок. Вбрасывается в случае устаревания используемой версии.
- Внедрена отправка оповещения при активации пробной подписки (Идентификатор оповещения: 15);

1.0 Первоначальная версия.

1. Общие

1.1. Получить страны

Запрос

Метод: info.getCountries

Тип: GET

Требуется token: да

Параметры:

- offset - отступ, необходимый для выбора определенного подмножества стран.
- count - количество стран, которое необходимо вернуть, по умолчанию 100, максимально 1000, минимально 10;
- q - поисковый запрос для выборки стран, если задан, то возвращаются страны соответствующие маске %q%, не обязательный

Ответ

```
{
  response: {
    count: 12,
    items: [
      {
        "id": "1",
        "name": "Москва"
      },
      ...
    ],
  }
}
```

1.2. Получить страну по идентификатору

Запрос

Метод: info.getCountryById

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор страны;

Ответ

```
{
  response: {
    "id": "1",
    "name": "Россия"
  }
}
```

1.3. Получить города страны

Запрос

Метод: info.getCities

Тип: GET

Требуется token: да

Параметры:

- country_id - идентификатор страны;
- offset - отступ, необходимый для выбора определенного подмножества городов.
- count - количество городов, которое необходимо вернуть, по умолчанию 100, максимально 1000, минимально 10;
- q - поисковый запрос для выборки городов, если задан, то возвращаются города соответствующие маске %q%, не обязательный

Ответ

```
{
  response: {
    count: 34,
    items: [
      {
        "id": "1",
        "name": "Москва"
      },
      ...
    ],
  }
}
```

1.4. Получить город по идентификатору

Запрос

Метод: info.getCityById

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор города;

Ответ

```
{
  response: {
    "id": "1",
    "name": "Москва"
  }
}
```

1.5. Отправить предложение по улучшению

Запрос

Метод: info.sendOffer

Тип: POST

Требуется token: да

Параметры:

- content - текст предложения;

Ответ

```
{
    response: 1 // в случае успешного сохранения предложения
}
```

1.6. Получить страну по названию

Запрос

Метод: info.getCountryByName

Тип: GET

Требуется token: да

Параметры:

- name - название страны;

Ответ

```
{
    response: {
        "id": "1" // идентификатор страны
    },
}
```

Коды ошибок:

- 161 - Передано несуществующее название страны;

1.7. Получить город по названию

Запрос

Метод: info.getCityByName

Тип: GET

Требуется token: да

Параметры:

- country_id - идентификатор страны;

- name - название города;

Ответ

```
{
  response: {
    "id": "1" // идентификатор города
  },
}
```

Коды ошибок:

- 171 - Передан несуществующий идентификатор страны или несуществующее название города;

1.8. Получить ссылку на страницу покупки подписки

Запрос

Метод: info.getAdditionalInfo

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  response: "https://cinarta.com/profile/subscription"
}
```

Для авторизации на сайте для этой страницы нужно к ссылке добавить токен в качестве get параметра: token=%TOKEN%, где %TOKEN% это токен доступа к API мобильного приложения полученный при входе в систему.

1.8. Получить возможные значения параметра “Тип внешности”

Запрос

Метод: info.getAppearanceTypeValues

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  "response": [
    {
      "id": 1,
      "name": "Американец"
    }
  ]
}
```

```
    },  
    {  
      "id": 2,  
      "name": "Африканец"  
    },  
    {  
      "id": 4,  
      "name": "Европеец"  
    }  
  ]  
}
```

2. Аккаунт

2.1. Отправить SMS код

Отправляет код подтверждения по SMS на указанный телефон.

Запрос

Метод: account.sendSmsCode

Тип: POST

Требуется token: нет

Параметры:

- phone - номер телефона, обязательный;
- app_token - токен приложения, необходимо запрашивать у разработчиков API;

Ответ

```
{
    response: 1 // в случае успеха
}
```

Коды ошибок:

- 211 - сервис временно недоступен;
- 212 - неверный формат телефона;
- 213 - недействительный токен приложения;
- 214 - слишком часто отправляет код;

2.2 Авторизация с использованием SMS кода

Авторизует посетителя по номеру телефону и коду.

Запрос

Метод: account.authBySmsCode

Тип: POST

Требуется token: нет

Параметры:

- phone - номер телефона, обязательный;
- sms_code - код пришедший в смс, обязательный;
- device_id - идентификатор устройства, обязательный;
- push_token - device registration token полученный у FCM, не обязателен;
- os_type - тип операционной системы, обязательный, возможны варианты:
 - 1 - Android;
 - 2 - iOS;

Ответ

Возвращает объект, содержащий следующие поля:

```
{
  response: {
    token: "", // токен авторизации, нужно передавать для совершения
    последующих запросов;
    status: 0, // статус пользователя:
                // - 0 - не зарегистрирован;
                // - 10 - зарегистрирован;
    user_id: 21 // идентификатор авторизованного пользователя;
    role: 1, // роль пользователя (см. перечисление ролей пользователя);
  }
}
```

Коды ошибок:

- 221 - неверный sms код;
- 222 - когда пробуют подобрать код, через 3 попытки возвращать что высланный код не действителен;
- 223 - посетитель с переданным телефоном заблокирован;
- 224 - неверное значение os_type;

2.3 Авторизация по соц. сетям

Запрос

Метод: account.authBySocialNetwork

Тип: POST

Требуется token: нет

Параметры:

- social - идентификатор соц. сети для авторизации, возможные значения:
 - 1 - facebook;
 - 2 - instagram;
 - 3 - twitter;
 - 4 - vk;
- token - полученный токен доступа соц. сети;
- device_id - идентификатор устройства, обязательный;
- push_token - device registration token полученный у FCM, не обязателен;
- token_secret - нужен для twitter, получается вместе с обычным токеном, обязателен для twitter;
- os_type - тип операционной системы, обязательный, возможны варианты:
 - 1 - Android;
 - 2 - iOS;

Ответ

```
{
  response: {
    token: "", // токен авторизации, нужно передавать для совершения
последующих запросов;
    status: 0, // статус пользователя:
                // - 0 - не зарегистрирован;
                // - 10 - зарегистрирован;
    user_id: 21 // идентификатор авторизованного пользователя;
    role: 1, // роль пользователя (см. перечисление ролей пользователя);
  }
}
```

Коды ошибок:

- общие ошибки;
- 231 - неверное значение social;
- 232 - неверное значение os_type;
- 233 - не удалось проверить валидность токена;
- 234 - аккаунт заблокирован;

2.4. Получить мой профиль

Запрос

Метод: account.getProfileInfo

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  response: {
    "id": "1", // идентификатор пользователя
    "first_name": "Николай", // имя на языке пользователя
    "last_name": "Петров", // фамилия на языке пользователя
    "first_name_inter": "Nikolai", // имя на латинице
    "last_name_inter": "Petrov", // фамилия на латинице;
    "status": "10", // статус пользователя:
                    // - 0 - не зарегистрирован
                    // - 10 - зарегистрирован)
    "email": "fgsdfsfsd@mail.ru",
    "sex": "1", // пол пользователя:
               // - 1 - женский;
               // - 2 - мужской;
    "is_public_contacts": "1", // если 0, то контакты отображаются только
кастинг директорам;
    "birthday": "1973-03-24", // день рождения в формате YYYY-MM-DD
  }
}
```

```

"lang": "en", // язык информации профиля;
"role": "1",    // роли пользователя, возможны значения:
                // - 1 - исполнитель;
                // - 2 - заказчик;
                // - 3 - администратор;
"user_comment": "Играю на трубе.", // доп. информация пользователя;
"phones": [ // массив телефонов пользователя
    {
        "phone_number": "+79234231144",
    },
],
"professions": [ // массив id профессий пользователя
    {
        "profession_id": "1",
    },
],
"photo": {
    "id": "1", // идентификатор фото
    "url": "http://5a1a7e2b6.png" // ссылка на фото
},
"country": {
    "id": "1", // идентификатор страны
    "name": "Россия" // название страны
},
"city": {
    "id": "1", // идентификатор города
    "name": "Москва" // название города
},
/* доп. параметры. подробнее смотри список профессий */
"ex_height": null,
"ex_hair_color": null,
"ex_eyes_color": null,
"ex_clothing_size": null,
"languages": [
    {
        "language_id": "1",
    },
    ...
],
"drivings": [
    {
        "driving_id": "3",
    },
    ...
],
}

```

```
}
```

Коды ошибок:

В ходе выполнения могут произойти общие ошибки.

2.5. Сохранить мой профиль

Запрос

Метод: account.saveProfileInfo

Тип: POST

Требуется token: да

Параметры:

- last_name - фамилия на языке пользователя, обязательное;
- first_name - имя на языке пользователя, обязательное;
- last_name_inter - фамилия на латинице, обязательное, только английский символы;
- first_name_inter - имя на латинице, обязательное, только английские символы;
- sex - пол, обязательный, возможны: 1 - женский, 2 - мужской
- email - почта, обязательный;
- user_comment - доп. информация о пользователе, необязательный;
- is_public_contacts - если 0, то контакты отображаются только кастинг директорам;
- country_id - id страны, обязательный;
- city_id - id города, обязательный;
- birthday - день рождения в формате YYYY-MM-DD;
- phones[i] - телефоны пользователя, обязателен хотя бы один, i от 0 и выше;
- professions[i] - id профессии пользователя, обязательна хотя бы одна, i от 0 и выше;
- photo_id - идентификатор ранее загруженной фотографии;
- Дополнительные (смотри список профессий):
 - ex_vocals - когда нужно передать один параметр;
 - ex_languages[i] - для передачи набора параметров (для удаления всех значений из профиля нужно передать массив с ключом 0 и пустым значением, т.е. ex_languages[0] с пустым значением);
 - ex_height - рост в сантиметрах
 - ex_height_ft -рост в футах
 - ex_height_type - единица измерения роста, возможны: 1 - см., 2 - фут. Обязательный, если передано значение роста в одной или двух единицах. Переданный тип будет использоваться как основной, рост в другой единице измерения будет пересчитан автоматически.

Ответ

```
{
  response: {
    "status": "10" // статус пользователя
  }
}
```

```
}
```

Коды ошибок:

В ходе выполнения могут произойти общие ошибки.

2.6. Получить счётчики

Возвращает значения счетчиков пользователя.

Запрос

Метод: account.getCounters

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  response: {
    likes: 2, // количество лайков
    views: 3, // количество просмотров
    notifications: 3, // количество непрочитанных оповещений
  }
}
```

2.7. Проверить правильность токена доступа

Запрос

Метод: account.validateToken

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  response: {
    status: 0, // статус пользователя:
              // - 0 - не зарегистрирован;
              // - 10 - зарегистрирован;
    role: 1, // роль пользователя (см. перечисление ролей пользователя);
    active_to: "1563072000" // время в unix time когда закончится подписка;
  }
}
```

2.8. Обновить токен регистрации FCM

Запрос

Метод: account.updatePushToken

Тип: GET

Требуется token: да

Параметры:

- device_id - идентификатор устройства, обязательный;
- push_token - device registration token полученный у FCM, обязателен;
- os_type - тип операционной системы, обязательный, возможны варианты:
 - 1 - Android;
 - 2 - iOS;

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 281 - не найдено устройство по переданным данным.

2.9. Получить список профилей пользователя в базах партнеров

Запрос

Метод: account.getProfilesFromPartners

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  response: {
    count: 3,
    items: [
      {
        "id": "7",
        "first_name": "Скарлет ",
        "last_name": "Йоханссон",
        "first_name_inter": "Scarlett",
        "last_name_inter": "Johansson",
        "photo_url": "http://api.kinopro365.com/uploads/fixture/photo/7.jpg",
        "country_name": "Russia",
        "city_name": "Moscow",
        "age": "32",
        "has_video": true
        "lang": "en",
        "partner_id" 1,
        "partner_name" "Partner Name",
      }
    ]
  }
}
```

```

        },
        ...
    ]
}
}

```

Параметр	Описание
count	количество элементов
items	
id	идентификатор пользователя в базе партнера
first_name	имя на языке пользователя
last_name	фамилия на языке пользователя
first_name_inter	имя на латинице
last_name_inter	фамилия на латинице
photo_url	ссылка на фото пользователя
country_name	страна
city_name	город
age	возраст
has_video	есть ли добавленное видео
lang	язык информации профиля
partner_id	идентификатор партнера, которому принадлежит пользователь
partner_name	название партнера

Коды ошибок:

В ходе выполнения могут произойти общие ошибки.

2.10. Получить профиль по партнеру

Запрос

Метод: account.getProfileInfoFromPartner

Тип: GET

Требуется token: да

Параметры:

- partner_id - идентификатор партнера, обязательный;

Ответ

```
{
  response: {
    "id": "1",
    "first_name": "Николай",
    "last_name": "Петров",
    "first_name_inter": "Nikolai",
    "last_name_inter": "Petrov",
    "role": "1",
    "user_comment": "Играю на трубе.",
    "status": "10",
    "email": "fgsdfsfsd@mail.ru",
    "sex": "1",
    "is_public_contacts": "1",
    "birthday": "1973-03-24",
    "lang": "en",
    "ex_height": null,
    "ex_hair_color": null,
    "ex_eyes_color": null,
    "ex_clothing_size": null,
    "phones": [
      {
        "phone_number": "+79234231144",
      },
      ...
    ],
    "professions": [
      {
        "profession_id": "1",
      },
      ...
    ],
    "photo": {
      "id": "1",
      "url": "http://5a1a7e2b6.png"
    },
    "country": {
      "id": "1",
      "name": "Россия"
    },
    "city": {
      "id": "1",
```

```

        "name": "Москва"
    },
    "languages": [
        {
            "language_id": "1",
        },
        ...
    ],
    "drivings": [
        {
            "driving_id": "3",
        },
        ...
    ],
}

```

Параметр	Описание
id	идентификатор пользователя в базе партнера
first_name	имя на языке пользователя
last_name	фамилия на языке пользователя
first_name_inter	имя на латинице
last_name_inter	фамилия на латинице
role	роль пользователя, см. перечисление ролей пользователей
user_comment	доп. информация пользователя
status	статус пользователя, см. перечисление статусов пользователей
email	email пользователя
sex	пол пользователя, см. перечисление "пол"
is_public_contacts	если 0, то контакты отображаются только кастинг директорам
birthday	день рождения в формате YYYY-MM-DD
lang	язык информации профиля
	доп. параметры, подробнее смотри список профессий
phones	массив телефонов пользователя

phone_number	номер телефона
professions	массив id профессий пользователя
profession_id	идентификатор профессии
photo	
id	идентификатор фото
url	ссылка на фото
country	
id	идентификатор страны
name	название страны
city	
id	идентификатор города
name	название города
languages	массив id языков пользователя
language_id	идентификатор языка, см. перечисление языков
drivings	массив id навыков вождения пользователя
driving_id	идентификатор навыка вождения, см. перечисление навыков вождения

Коды ошибок:

- 2101 - Для данного партнера нет профиля пользователя;

3. Фотографии

3.1. Сохранить фото

Только для версий api меньше 1.8.

Запрос

Метод: photo.save

Тип: POST

Требуется token: да

Параметры:

- file1 - файл фотографии;

Ответ

```
{
  response: {
    "id": "3",
    "url": "http://58868ca75af7e.jpg"
  }
}
```

3.2. Удалить фото

Запрос

Метод: photo.delete

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор фотографии для удаления;

Ответ

```
{
  response: 1
}
```

Коды ошибок:

- 321 - недостаточно прав для удаления фото;
- 322 - невозможно удалить аватарку профиля;

3.3. Получить фото по id

Запрос

Метод: photo.getByid

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор фотографии;

Ответ

```
{
  response: {
    "id": "3",
    "url": "http://58868ca75af7e.jpg"
  }
}
```

Коды ошибок:

- 331 - фотография не найдена;

3.4. Получить все фотографии пользователя

Запрос

Метод: photo.getAll

Тип: GET

Требуется token: да

Параметры:

- offset - отступ, необходимый для выбора определенного подмножества фотографий.
- count - количество фотографий, которое необходимо вернуть, по умолчанию 100, максимально 1000;

Ответ

```
{
  "response": {
    "count": "3",
    "items": [
      {
        "id": "1",
        "url": "http:///58868ca75af7e.jpg"
      },
      ...
    ],
  },
}
```

3.5. Получить фотографии профиля

Запрос

Метод: photo.getProfile

Тип: GET

Требуется token: да

Параметры:

- offset - отступ, необходимый для выбора определенного подмножества фотографий.
- count - количество фотографий, которое необходимо вернуть, по умолчанию 100, максимально 1000;
- owner_id - id владельца фотографий, если не передать, вернёт фото текущего профиля;

Ответ

```
{
  "response": {
    "count": "3",
    "items": [
      {
        "id": "1",
        "url": "http:///58868ca75af7e.jpg"
      },
    ],
  },
}
```

```
    ...  
  ],  
}  
}
```

Коды ошибок:

- 351 - передан несуществующий идентификатор пользователя;

3.6. Добавить фотографию в профиль

Запрос

Метод: photo.addToProfile

Тип: GET

Требуется token: да

Параметры:

- photo_id - идентификатор фото для добавления в профиль;

Ответ

```
{  
  "response": 1  
}
```

Коды ошибок:

- 361 - фото с переданным id не принадлежит пользователю;
- 362 - достигнут лимит количества фотографий в профиле;
- 363 - фото уже добавлено в профиль;
- стандартные коды ошибок;

3.7. Удалить фотографию из профиля

Запрос

Метод: photo.removeFromProfile

Тип: GET

Требуется token: да

Параметры:

- photo_id - идентификатор фото для удаления из профиля;

Ответ

```
{  
  "response": 1  
}
```

Коды ошибок:

- 371 - фото не добавлено в профиль пользователя;
- стандартные коды ошибок;

3.8. Получить адрес сервера для загрузки фото

Запрос

Метод: photo.getUploadServer

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  "response": "https://fs1.cinarta.com"
}
```

Параметры	Описание
response	адрес сервера

3.9. Сохранить фото на фотосервер

Запрос

Метод: нет

Тип: POST

Требуется token: да

Параметры:

- file1 - файл фотографии;

Ответ

```
{
  response: {
    "id": "3",
    "url": "https://58868ca75af7e.jpg"
  }
}
```

Параметры	Описание
id	идентификатор фотографии
url	ссылка на фотографию

Запрос отправляется не на основной сервер (не на cinarta.com), а на адрес, полученный из метода 3.8.

4. Видео

4.1. Сохранить видео

Запрос

Метод: video.save

Тип: POST

Требуется token: да

Параметры:

- link - ссылка на видео, обязательный;

Ответ

```
{
  response: {
    "id": "4" // идентификатор добавленного видео
  }
}
```

Коды ошибок:

- 421 - Не удалось нормализовать ссылку;

4.2. Удалить видео

Запрос

Метод: video.delete

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор видео для удаления;

Ответ

```
{
  response: 1
}
```

Коды ошибок:

- 421 - Видео не существует;
- 422 - Невозможно удалить не свои видео;

4.3. Получить видео

Запрос

Метод: video.getByid

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор видео;

Ответ

```
{
  "response": {
    "id": "1",
    "link": "https://www.youtube.com/.."
  }
}
```

Коды ошибок:

- 431 - видео не найдено;

4.4. Получить все видео пользователя

Запрос

Метод: video.getAll

Тип: GET

Требуется token: да

Параметры:

- offset - отступ, необходимый для выбора определенного подмножества видео.
- count - количество видео, которое необходимо вернуть, по умолчанию 100, максимально 1000;

Ответ

```
{
  "response": {
    "count": "3",
    "items": [
      {
        "id": "1",
        "link": "https://www.youtube.com/.."
      },
      ...
    ],
  }
}
```

Коды ошибок:

- Возможны общие коды ошибок;

4.5. Получить видео профиля

Запрос

Метод: video.getProfile

Тип: GET

Требуется token: да

Параметры:

- offset - отступ, необходимый для выбора определенного подмножества видео.

- count - количество видео, которое необходимо вернуть, по умолчанию 100, максимально 1000;
- owner_id - id владельца видео, если не передать, вернёт видео текущего профиля;

Ответ

```
{
  "response": {
    "count": "3",
    "items": [
      {
        "id": "1",
        "link": "https://www.youtube.com/.."
      },
      ...
    ],
  }
}
```

Коды ошибок:

- 451 - передан несуществующий идентификатор пользователя;
- Возможны общие коды ошибок;

4.6. Добавить видео в профиль

Запрос

Метод: video.addToProfile

Тип: GET

Требуется token: да

Параметры:

- video_id - идентификатор видео для добавления в профиль;

Ответ

```
{
  "response": 1
}
```

Коды ошибок:

- 461 - видео с переданным id не принадлежит пользователю;
- стандартные коды ошибок;

4.7. Удалить видео из профиля

Запрос

Метод: video.removeFromProfile

Тип: GET

Требуется token: да

Параметры:

- video_id - идентификатор видео для удаления из профиля;

Ответ

```
{  
    "response": 1  
}
```

Коды ошибок:

- 471 - видео не добавлено в профиль пользователя;
- стандартные коды ошибок;

5. Пользователи

5.1. Получить количество пользователей по профессиям

Запрос

Метод: user.countByProfessions

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  response: [
    {
      "profession_id": "1",
      "count": "11"
    },
    {
      "profession_id": "2",
      "count": "1"
    },
    ..
  ]
}
```

5.2. Поиск пользователей

Возвращает список пользователей в соответствии с заданным критерием поиска.

Запрос

Метод: user.search

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первого найденного пользователя для выборки определенного подмножества;
- count - количество возвращаемых пользователей, по умолчанию 20, максимальное значение 1000;
- profession_id - идентификатор профессии, обязательное;
- first_name - поисковая фраза по имени;
- last_name - поисковая фраза по фамилии;
- first_name_inter - поисковая фраза по международному имени;
- last_name_inter - поисковая фраза по международной фамилии;
- sex - пол для фильтрации, 1 - женщина, 2 - мужчина;

- age_from - возраст от для фильтрации;
- age_to - возраст до для фильтрации;
- country_id - идентификатор страны для фильтрации;
- city_id - идентификатор города для фильтрации;
- Дополнительные параметры фильтрации (зависит от профессии):
 - ex_height_from, ex_height_to - для поиска по росту;
 - ex_height_type - единица измерения роста, 1 - см, 2 - фут;
 - ex_languages[] - для фильтрации по языкам, возможна передача нескольких значений;
 - ex_drivings[] - для фильтрации по навыкам вождения, возможна передача нескольких значений;
 - ex_bust[i] - бюст, возможно несколько значений;
 - allow_playing_age - "учитывать игровой возраст" возможны значения:
 - 1 - учитывать.
 - остальные доп. параметры передаются: название_поля=значение

Ответ

```
{
  response: {
    count: 323,
    items: [
      {
        "id": "7", // идентификатор пользователя
        "first_name": "Скарлет ",
        "last_name": "Йоханссон",
        "first_name_inter": "Scarlett",
        "last_name_inter": "Johansson",
        "photo_url": "http://api.kinopro365.com/uploads/fixture/photo/7.jpg",
        "country_name": "Russia", // страна
        "city_name": "Moscow", // город
        "age": "32", // возраст
        "is_like": "0", // ставил или нет лайк пользователь
        "is_favourite": "0", // находится или нет в избранном
        "count_likes": "0", // количество лайков пользователя
        "lang": "en", // язык информации профиля
      },
      ...
    ]
  }
}
```

5.3. Получить информацию пользователя

Запрос

Метод: user.get

Тип: GET

Требуется token: да

Параметры:

- user_id - идентификатор пользователя;
- profession_id - идентификатор профессии, для определения дополнительных возвращаемых полей, не обязательный;

Ответ

```
{
  "response": {
    "id": "2",
    "first_name": "Джонни",
    "last_name": "Депп",
    "first_name_inter": "John",
    "last_name_inter": "Depp",
    "user_comment": null,
    "photo_url": "http://localhost:8086/uploads/fixture/photo/2.jpg",
    "sex": "2",
    "is_public_contacts": "0",
    "country_name": "Russia",
    "city_name": "Moscow",
    "age": "53",
    "is_like": "0", // поставлен или нет лайк пользователем
    "is_favourite": "0", // находится или нет в избранном пользователя
    "email": "user2@example.com", // email, если закрыты контакты, то поле не
    придёт;
    ... доп. параметры...
    "count_likes": "0", // количество лайков
    "count_photos": "4", // количество фотографий
    "count_videos": "1", // количество видео
    "lang": "en", // язык информации профиля
    "phones": [ // контактные телефоны, если закрыты, то массив не придёт;
      {
        "phone_number": "89137891425",
        "user_id": "2"
      },
      {
        "phone_number": "89237418596",
        "user_id": "2"
      }
    ],
    "languages": [
      {
        "language_id": "1",
        "user_id": "2"
      },
      {
        "language_id": "3",
```

```

        "user_id": "2"
      },
      ...
    ],
    "drivings": [
      {
        "driving_id": "1",
        "user_id": "2"
      },
      {
        "driving_id": "3",
        "user_id": "2"
      },
      ...
    ],
    "is_open_contacts": 1 // открыты или нет контакты
    "has_request_to_open_contacts": "1", // есть ли запрос на открытие
    контактов этого пользователя
  }
}

```

Коды ошибок:

- 531 - пользователь не найден.

5.4. Поставить лайк пользователю

Запрос

Метод: user.like

Тип: GET

Требуется token: да

Параметры:

- user_id - идентификатор пользователя;

Ответ

```

{
  response: 1
}

```

Коды ошибок:

- 541 - нельзя поставить лайк самому себе;
- 542 - передан несуществующий идентификатор пользователя;
- 543 - пользователю уже поставлен лайк;
- 544 - нет активной подписки;

5.5. Убрать лайк у пользователя

Запрос

Метод: user.unlike

Тип: GET

Требуется token: да

Параметры:

- user_id - идентификатор пользователя;

ОТВЕТ

```
{
    response: 1
}
```

Коды ошибок:

- 551 - Передан несуществующий идентификатор пользователя
- 552 - Пользователю не был поставлен лайк

~~5.6. Дать награду пользователю~~

Удален

~~5.7. Забрать награду пользователя~~

Удален

5.8. Получить количество в избранном по профессиям

Запрос

Метод: user.countFavouriteByProfessions

Тип: GET

Требуется token: да

Параметры: нет

ОТВЕТ

```
{
    response: [
        {
            "profession_id": "1",
            "count": "4"
        },
        {
            "profession_id": "2",
            "count": "1"
        },
        ..
    ]
}
```

5.9. Получить избранное

Запрос

Метод: user.getFavourite

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первого найденного пользователя для выборки определенного подмножества;
- count - количество возвращаемых пользователей, по умолчанию 20, максимальное значение 1000;
- profession_id - идентификатор профессии;
- first_name - поисковая фраза по имени;
- last_name - поисковая фраза по фамилии;
- first_name_inter - поисковая фраза по международному имени;
- last_name_inter - поисковая фраза по международной фамилии;
- sex - пол для фильтрации, 1 - женщина, 2 - мужчина;
- age_from - возраст от для фильтрации;
- age_to - возраст до для фильтрации;
- country_id - идентификатор страны для фильтрации;
- city_id - идентификатор города для фильтрации;
- Дополнительные параметры фильтрации (зависит от профессии):
 - ex_height_from, ex_height_to - для поиска по росту;
 - ex_height_type - единица измерения роста, 1 - см, 2 - фут;
 - ex_languages[] - для фильтрации по языкам, возможна передача нескольких значений;
 - ex_bust[i] - бюст, возможно несколько значений;
 - allow_playing_age - "учитывать игровой возраст" возможны значения:
 - 1 - учитывать.
 - остальные доп. параметры передаются: название_поля=значение

Ответ

```
{
  response: [
    {
      "id": "7", // идентификатор пользователя
      "first_name": "Скарлет ",
      "last_name": "Йоханссон",
      "first_name_inter": "Scarlett",
      "last_name_inter": "Johansson",
      "photo_url": "http://api.kinopro365.com/uploads/fixture/photo/7.jpg",
      "country_name": "Russia", // страна
      "city_name": "Moscow", // город
      "age": "32", // возраст
    }
  ]
}
```

```

        "is_like": "0", // ставил или нет лайк пользователь
        "count_likes": "0", // количество лайков пользователя
        "lang": "en", // язык информации профиля
    },
    ...
]
}

```

Коды ошибок:

- 591 - Передан несуществующий идентификатор профессии

5.10. Добавить в избранное

Запрос

Метод: user.addToFavourite

Тип: GET

Требуется token: да

Параметры:

- user_id - идентификатор пользователя;

Ответ

```

{
    response: 1
}

```

Коды ошибок:

- 5101 - Передан несуществующий идентификатор пользователя
- 5102 - Нельзя добавить в избранное самого себя
- 5103 - Пользователь уже находится в избранном

5.11. Удалить из избранного

Запрос

Метод: user.removeFromFavourite

Тип: GET

Требуется token: да

Параметры:

- user_id - идентификатор пользователя;

Ответ

```

{
    response: 1
}

```

Коды ошибок:

- 5111 - Передан несуществующий идентификатор пользователя
- 5112 - Пользователь не находится в избранном

5.12. Запросить открытие контактных данных

Запрос

Метод: user.requestToOpenContacts

Тип: POST

Требуется token: да

Параметры:

- user_id - идентификатор пользователя;
- profession_id - идентификатор профессии пользователя;
- request_message - сообщение (необязательный параметр);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 5121 - Запрос на открытие контактов доступен только пользователям
- 5122 - Превышен лимит запросов. Не более 5 запросов в сутки
- 5123 - Передан несуществующий идентификатор пользователя
- 5124 - Пользователю уже отправлен запрос

5.13. Открыть контакты

Запрос

Метод: user.acceptRequestContact

Тип: GET

Требуется token: да

Параметры:

- by_user_id - идентификатор пользователя, отправившего запрос;

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 5131 - Передан несуществующий идентификатор пользователя
- 5132 - От данного пользователя нет заявки на обмен контактными данными
- 5133 - Заявка уже была рассмотрена ранее

5.14. Не открывать контакты

Запрос

Метод: user.rejectRequestContact

Тип: GET

Требуется token: да

Параметры:

- by_user_id - идентификатор пользователя, отправившего запрос;

ОТВЕТ

```
{
    response: 1
}
```

Коды ошибок:

- 5141 - Передан несуществующий идентификатор пользователя
- 5142 - От данного пользователя нет заявки на обмен контактными данными
- 5143 - Заявка уже была рассмотрена ранее

5.15. Создать жалобу на пользователя

Запрос

Метод: user.createComplaint

Тип: POST

Требуется token: да

Параметры:

- to_user_id - идентификатор пользователя на которого подается жалоба (обязателен);
- text - текст жалобы (обязателен);

ОТВЕТ

```
{
    response: 1
}
```

Коды ошибок:

- 5151 - пользователь на которого подается жалоба не найден;

5.16. Добавить отзыв

Запрос

Метод: user.createReview

Тип: POST

Требуется token: да

Параметры:

- to_user_id - идентификатор пользователя на которого будет оставлен отзыв (обязателен);
- text - текст отзыва (обязателен);

ОТВЕТ

```
{
    response: 1
}
```

```
}
```

Коды ошибок:

- 5161 - пользователь на которого создается отзыв не найден;
- 5162 - метод доступен только проверенным кастинг-директорам;

5.17. Редактировать отзыв

Запрос

Метод: user.updateReview

Тип: POST

Требуется token: да

Параметры:

- id - идентификатор отзыва (обязателен);
- text - текст отзыва (обязателен);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 5171 - отзыв не найден;
- 5172 - метод доступен только проверенным кастинг-директорам;
- 5173 - пользователь не является автором отзыва;

5.18. Получить отзывы

Запрос

Метод: user.getReviews

Тип: GET

Требуется token: да

Параметры:

- user_id - идентификатор пользователя, отзывы которого нужно получить - обязательный для заполнения;
- count - количество отзывов, которое необходимо вернуть, по умолчанию 100, максимально 1000;
- offset - отступ, необходимый для выбора определенного подмножества отзывов;

Ответ

```
{
  response: {
    items: [
      {
        "text": "review text",           // текст отзыва
        "created_at": 1500123123,       // дата отзыва
        "id": 2,                        // идентификатор отзыва
        "author_request": true           // пользователь является ли автором отзыва
      }
    ]
  }
}
```

```
    },  
    ...  
  ],  
}  
}
```

5.19. Удалить отзыв

Запрос

Метод: user.deleteReview

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор отзыва (обязателен);

Ответ

```
{  
  response: 1  
}
```

Коды ошибок:

- 5191 - отзыв не найден;

6. Вакансии

6.1. Получить список вакансий

Запрос

Метод: vacancy.getList

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первой найденной вакансии для выборки определенного подмножества;
- count - количество возвращаемых вакансий, по умолчанию 20, максимальное значение 1000, минимальное - 10;
- country_id - идентификатор страны для фильтрации (не обязателен);
- city_id - идентификатор города для фильтрации (не обязателен);

Ответ

```
{
  response: {
    count: 542,
    items: [
      {
        "id": "1", // идентификатор вакансии
        "logo_url":
          "http://api.kinopro365.com/uploads/fixture/photo/2.jpg", // логотип
          (если есть или null)
        "name": "vacancy 1", // название
        "description": "text", // описание вакансии
        "country_name": "Russia", // страна
        "city_name": "Moscow", // город
        "end_at": "1490123124", // до какого времени активна
        "profession_id": "1", // идентификатор профессии
      },
      ...
    ],
  }
}
```

6.2. Получить информацию по вакансии

Запрос

Метод: vacancy.get

Тип: GET

Требуется token: да

Параметры:

- vacancy_id - идентификатор вакансии;

Ответ

```
{
  response: {
    "id": "4", // идентификатор вакансии
    "logo_url": "http://api.kinopro365.com/uploads/fixture/photo/3.jpg", // ссылка
    на логотип
    "name": "vacancy 4", // название
    "description": "text", // описание вакансии
    "country_name": "Belarus", // страна
    "city_name": "Minsk", // город
    "end_at": "1490123125", // до какого времени активна
    "profession_id": "1", // идентификатор профессии
    "count_offer": "2", // количество заявок
    "type": "1", // тип вакансии (из перечисления типов вакансий)
  }
}
```

Коды ошибок:

- 621 - Вакансия не найдена

6.3. Создать вакансию

Запрос**Метод:** vacancy.create**Тип:** POST**Требуется token:** да**Параметры:**

- name - название вакансии, обязательный;
- logo_id - идентификатор ранее загруженного изображения (не обязательно);
- end_at - unix time, дата окончания приема заявок (обязательный);
- profession_id - идентификатор профессии (обязательный);
- country_id - идентификатор страны (обязательный);
- city_id - идентификатор города (обязательный);
- description - описание (обязательный).

Ответ

```
{
  response: 1
}
```

Коды ошибок:

- 631 - нет активной подписки;

6.4. Редактировать вакансию

Запрос

Метод: vacancy.edit

Тип: POST

Требуется token: да

Параметры:

- vacancy_id - идентификатор вакансии для редактирования (обязательный);
name - название вакансии;
- logo_id - идентификатор ранее загруженного изображения (не обязательно);
- end_at - 1490123123, дата окончания приема заявок;
- profession_id - идентификатор профессии
- country_id - идентификатор страны
- city_id - идентификатор города
- description - описание

Ответ

```
{  
    response: 1  
}
```

Коды ошибок:

- 641 - вакансия не найдена;
- 642 - вакансия с переданным id не принадлежит пользователю;
- 643 - нет активной подписки;

6.5. Удалить вакансию

Запрос

Метод: vacancy.delete

Тип: GET

Требуется token: да

Параметры:

- vacancy_id - идентификатор вакансии для удаления;

Ответ

```
{  
    response: 1  
}
```

Коды ошибок:

- 651 - Вакансия не найдена
- 652 - Вакансия с переданным id не принадлежит пользователю

6.6. Получить мои вакансии

Запрос

Метод: vacancy.getListAsAuthor

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первой найденной вакансии для выборки определенного подмножества;
- count - количество возвращаемых вакансий, по умолчанию 20, максимальное значение 1000, минимальное - 10
- is_active - выбирать активные или неактивные вакансии;
 - 1 - только активные
 - 0 - только неактивные
 - параметр не указан - все вакансии

Ответ

```
{
  response: {
    "count": "10" // количество вакансий;
    "items": [
      {
        "id": "5", // идентификатор вакансии;
        "logo_url": "http://api.kinopro365.com/uploads/fixture/photo/6.jpg", //
        ссылка на логотип;
        "name": "vacancy", // название вакансии;
        "description": "text", // описание;
        "country_name": "Россия",
        "city_name": "Москва",
        "end_at": "1490123000",
        "profession_id": "1", // идентификатор профессии;
        "review_status": "1", // статус модерации;
        // 1 - ожидает модерацию;
        // 2 - одобрена;
        // 3 - отклонена;
        "is_active": "1",
        "count_offer": "3", // количество заявок;
      },
      ...
    ],
  }
}
```

6.7. Получить информацию о вакансии как создатель

Запрос

Метод: vacancy.getAsAuthor

Тип: GET

Требуется token: да

Параметры:

- vacancy_id - идентификатор вакансии;

Ответ

```
{
```

```

response: {
  "id": "1", // идентификатор вакансии;
  "logo_url": "http://api.kinopro365.com/uploads/fixture/photo/8.jpg", // ссылка
на логотип;
  "name": "vacancy", // название вакансии;
  "description": "text", // описание;
  "country_id": "1", // идентификатор страны;
  "country_name": "Россия", // страна;
  "city_id": "1", // идентификатор города;
  "city_name": "Москва", // город;
  "end_at": "15000000", // дата окончания приема заявок;
  "profession_id": "1", // идентификатор профессии;
  "review_status": "1", // статус модерации;
    // 1 - ожидает модерацию;
    // 2 - одобрена;
    // 3 - отклонена;
  "review_message": "", // причина отклонения;
  "count_offer": "3" // количество заявок;
}
}

```

Коды ошибок:

- 661 - Вакансия не найдена
- 662 - Вакансия с переданным id не принадлежит пользователю

6.8. Подать заявку на вакансию

Запрос

Метод: vacancy.apply

Тип: GET

Требуется token: да

Параметры:

- vacancy_id - идентификатор вакансии;

Ответ

```

{
  response: 1
}

```

Коды ошибок:

- 681 - вакансия не найдена;
- 682 - нельзя подать заявку на свою вакансию;
- 683 - заявка уже была подана;
- 684 - нет активной подписки;
- 685 - нельзя подать заявку на вакансию с типом "создана вне системы";

6.9. Получить список предложений

Запрос

Метод: vacancy.getOffers

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первого найденного предложения для выборки определенного подмножества;
- count - количество возвращаемых предложений, по умолчанию 20, максимальное значение 1000, минимальное - 10
- vacancy_id - идентификатор вакансии;

Ответ

```
{
  response: {
    count: 43, // количество заявок
    items: [
      {
        "id": "2",
        "first_name": "Джонни", // имя на языке пользователя;
        "last_name": "Депп", // фамилия на языке пользователя;
        "first_name_inter": "John", // имя на латинице;
        "last_name_inter": "Depp", // фамилия на латинице;
        "photo_url":
          "http://api.kinopro365.com/uploads/fixture/photo/2.jpg", // ссылка на
          фото пользователя;
        "sex": "1", // пол пользователя:
          // - 1 - женский;
          // - 2 - мужской;
        "country_name": "Россия", // страна;
        "city_name": "Москва", // город;
        "age": "53", // возраст;
        "email": "user2@example.com", // email
        "is_like": "1", // ставил или нет лайк пользователь
        "is_favourite": "0", // находится или нет в избранном
        "count_likes": "3", // количество лайков пользователя
        "lang": "en", // язык информации профиля
        "phones": [ // номера телефонов;
          {
            "phone_number": "+79519501001",
            "user_id": "2"
          },
          ...
        ]
      }
    ]
  }
}
```

```
        },
        ...
    ]
}
```

Коды ошибок:

- 691 - Вакансия не найдена
- 692 - Вакансия с переданным id не принадлежит пользователю

6.10. Создать жалобу на вакансию

Запрос

Метод: vacancy.createComplaint

Тип: POST

Требуется token: да

Параметры:

- to_vacancy_id - идентификатор вакансии на которую подается жалоба (обязателен);
- text - текст жалобы (обязателен);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 6101 - вакансия не найдена;

7. Кастинги

7.1. Получить кастинги

Запрос

Метод: casting.search

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первого найденного кастинга для выборки определенного подмножества;
- count - количество возвращаемых кастингов, по умолчанию 20, максимальное значение 1000, минимальное - 10;
- country_id - идентификатор страны
- city_id - идентификатор города
- profession_id - идентификатор профессии
- project_type_id - идентификатор типа проекта

Ответ

```
{
  response: {
    count: 343,
    items: [
      {
        "id": "1",
        "logo_url":
"http://api.kinopro365.com/uploads/fixture/photo/2.jpg",
        "name": "casting 1",
        "description": "text text",
        "country_name": "Russia",
        "city_name": "Moscow",
        "end_at": "1501000000",
        "project_type_id": "1",
        "casting_director_status": "Проверенный кастинг директор";
        "warning_text": "Возможно эро сцены";
        "type": 1,
        "professions": [
          {
            "profession_id": 1,
            "casting_id": 1,
          },
          {
            "profession_id": 2,
```

```
        "casting_id": 1,
      },
      ...
    ],
  },
  ...
]
}
```

Параметр	Описание
count	количество элементов
items	массив кастингов
id	идентификатор кастинга
logo_url	ссылка на логотип
name	название кастинга
description	описание
country_name	страна
city_name	город
end_at	дата окончания приема заявок
project_type_id	тип проекта
casting_director_status	статус кастинг директора
warning_text	предупреждающий текст
type	тип кастинга, смотри возможные значения в перечислении Типы кастингов.
professions	массив специализаций
profession_id	идентификатор специализации
casting_id	идентификатор кастинга

7.2. Получить информацию о кастинге

Запрос

Метод: casting.get

Тип: GET

Требуется token: да

Параметры:

- casting_id - идентификатор кастинга (обязательный);

Ответ

```
{
  response: {
    "id": "1",
    "logo_url": "http://api.kinopro365.com/uploads/fixture/photo/2.jpg",
    "name": "casting 1",
    "description": "text text",
    "country_name": "Russia",
    "city_name": "Moscow",
    "end_at": "1501000000",
    "project_type_id": "1",
    "profession_id": "1",
    "apply_submitted": 1,
    "age_from": "20",
    "age_to": "40",
    "sex": "2",
    "type": 1,
    "requests_email": "request@test.com",
    "requests_email_subject": "На кастинг.",
    "whatsapp_phone": "+71234567788",
    "photo_description": "Test photo description",
    "use_playing_age": false,
    "ex_bust": null,
    "ex_height_from": "170",
    "ex_height_to": "180",
    "ex_height_ft_from": "5.2",
    "ex_height_ft_to": "6.0",
    "ex_height_type": "1",
    "ex_hair_color": "1",
    "ex_eyes_color": "1",
    "ex_clothing_size": "8",
    "ex_vocals": null,
    "ex_fencing": "1",
    "ex_martial_stunt": null,
    "ex_acrobatics": null,
  }
}
```

```

        "ex_car_stunt": null,
        "ex_moto_stunt": null,
        "ex_fiery_stunt": "1",
        "ex_fights_director": null,
        "ex_appearance_type": 4,
        "count_offer": "3",
        "contact_info": "casting_director@mail.ru, 8-800-900-01-02",
        "cost": "1000.00",
        "not_commercial": false,
        "is_nude": false,
        "casting_director_status": "Проверенный кастинг директор",
        "warning_text": "Возможно эро сцены",
        "can_send_apply": true,
        "languages": [
            {
                "language_id": "3",
                "casting_id": "1",
            },
            ...
        ],
        "drivings": [
            {
                "driving_id": "3",
                "casting_id": "1"
            },
            ...
        ],
        "professions": [
            {
                "profession_id": 1,
                "casting_id": "1"
            },
            ...
        ],
    },
}

```

Параметр	Описание
id	количество элементов
logo_url	лого кастинга
name	название кастинга

description	описание кастинга
country_name	название страны
city_name	название города
end_at	дата окончания приема заявок
project_type_id	
profession_id	профессия, для API версий < 1.6
apply_submitted	была или нет подана заявка текущим пользователем на кастинг, возможные значения: • 1 - да; • 0 - нет.
age_from	возраст от
age_to	возраст до
sex	пол
type	тип кастинга, смотри возможные значения в перечислении Типы кастингов.
requests_email	email для отправки заявок
requests_email_subject	тема email для отправки заявок
whatsapp_phone	whatsapp телефон для отправки заявки
photo_description	описание для фотографий
use_playing_age	
ex_bust	
ex_height_from	
ex_height_to	
ex_height_ft_from	
ex_height_ft_to	
ex_height_type	
ex_hair_color	
ex_eyes_color	

ex_clothing_size	
ex_vocals	
ex_fencing	
ex_martial_stunt	
ex_acrobatics	
ex_car_stunt	
ex_moto_stunt	
ex_fiery_stunt	
ex_fights_director	
ex_appearance_type	идентификатор типа внешности
count_offer	общее количество заявок
contact_info	поле выдается, если заявка пользователя на кастинг одобрена
cost	цена - 2 знака после запятой
not_commercial	если true, то некоммерческий
is_nude	если true, то ню кастинг
casting_director_status	
warning_text	
can_send_apply	если true, тогда разрешено подавать заявки на кастинг
languages	языки
language_id	идентификатор специализации
casting_id	идентификатор кастинга
drivings	навыки вождения
driving_id	идентификатор навыка вождения
casting_id	идентификатор кастинга
professions	профессии
profession_id	идентификатор профессии
casting_id	идентификатор кастинга

Коды ошибок:

- 721 - Кастинг не найден

7.3. Создать кастинг

Запрос

Метод: casting.create

Тип: POST

Требуется token: да

Параметры:

- logo_id - логотип кастинга (не обязательно)
- name - название кастинга (обязательно)
- description - описание (обязательно)
- country_id - идентификатор страны
- city_id - идентификатор города
- end_at - 1490123123, дата окончания приема заявок (обязательно)
- project_type_id - идентификатор типа проекта (обязательно)
- profession_id - идентификатор профессии (обязательно), для версий < 1.6
- contact_info - информация и контакты для утвержденных анкет (обязательно)
- age_from
- age_to
- use_playing_age - учитывать игровой возраст: 1 - Да, 0 - Нет
- sex
- ex_bust - бюст
- ex_height_from - рост "от" в сантиметрах
- ex_height_to - рост "до" в сантиметрах
- ex_height_ft_from - рост "от" в футах
- ex_height_ft_to - рост "до" в футах
- ex_height_type - единица измерения роста: 1 - см., 2 - фут. (Обязательный, если передано хотя бы одно значение роста. Будет использоваться значение роста для переданной единицы измерения, рост в другой единице измерения будет пересчитан)
- ex_hair_color
- ex_eyes_color
- ex_clothing_size
- ex_vocals
- ex_fencing
- ex_martial_stunt
- ex_acrobatics
- ex_car_stunt
- ex_moto_stunt
- ex_fiery_stunt
- ex_fights_director
- ex_appearance_type
- ex_languages[i]

- ex_drivings[i]
- cost - цена - число, 2 знака после запятой
- not_commercial - некоммерческий: 1 - Да, 0 - Нет. Если передан 1, то цена (cost) не сохраняется (обязательно)
- professions[i] - специализации (обязательно хотя бы один параметр)
- is_nude - кастинг в стиле НЮ
- type - тип кастинга, см. перечисление Типы кастингов. (обязательно)
- requests_email - email для отправки заявок (обязательно к заполнению и сохранится только, если выбран type = 2)
- requests_email_subject - тема письма для заявки (обязательно к заполнению и сохранится только, если выбран type = 2)
- whatsapp_phone - номер whatsapp для отправки заявок (обязательно к заполнению и сохранится только, если выбран type = 3)
- photo_description - описание для фотографий (сохранится только, если выбран type = 3)
- moderation_on - включить модерацию кастинга. Если передан 1, то кастинг будет проверен модератором, если 0 - то будет работать автомодерация. По умолчанию 0. Параметр только для проверенных кастинг-директоров.
- reason_for_moderation - описание, для чего кастинг-директор, хочет, чтобы кастинг проверил модератор, не обязательный. Параметр только для проверенных кастинг-директоров.

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- В ходе выполнения могут произойти общие ошибки.

7.4. Редактировать кастинг

Запрос

Метод: casting.edit

Тип: POST

Требуется token: да

Параметры:

- casting_id (обязательно)
- logo_id - логотип кастинга (не обязательно)
- name - название кастинга
- description - описание
- country_id - идентификатор страны
- city_id - идентификатор города
- end_at - 1490123123, дата окончания приема заявок
- project_type_id - идентификатор типа проекта
- profession_id - идентификатор профессии, для версий < 1.6
- contact_info - информация и контакты для утвержденных анкет

- age_from
- age_to
- use_playing_age - учитывать игровой возраст: 1 - Да, 0 - Нет
- ex_bust - бюст
- sex - пол
- ex_height_from - рост "от" в сантиметрах
- ex_height_to - рост "до" в сантиметрах
- ex_height_ft_from - рост "от" в футах
- ex_height_ft_to - рост "до" в футах
- ex_height_type - единица измерения роста: 1 - см., 2 - фут. (Обязательный, если передано хотя бы одно значение роста. Будет использоваться значение роста для переданной единицы измерения, рост в другой единице измерения будет пересчитан)
- ex_hair_color
- ex_eyes_color
- ex_clothing_size
- ex_vocals
- ex_fencing
- ex_martial_stunt
- ex_acrobatics
- ex_car_stunt
- ex_moto_stunt
- ex_fiery_stunt
- ex_fights_director
- ex_appearance_type
- ex_languages[i] - ((для удаления всех значений из профиля нужно передать массив с ключом 0 и пустым значением, т.е. ex_languages[0] с пустым значением))
- ex_drivings[i] - ((для удаления всех значений из профиля нужно передать массив с ключом 0 и пустым значением, т.е. ex_drivings[0] с пустым значением))
- cost - цена - число, 2 знака после запятой
- not_commercial - некоммерческий: 1 - Да, 0 - Нет. Если передан 1, то цена (cost) не сохраняется (обязательно)
- professions[i] - специализации (обязательно хотя бы один параметр)
- is_nude - кастинг в стиле НЮ
- type - тип кастинга, см. перечисление Типы кастингов. (обязательно)
- requests_email - email для отправки заявок (обязательно к заполнению и сохранится только, если выбран type = 2)
- requests_email_subject - тема письма для заявки (обязательно к заполнению и сохранится только, если выбран type = 2)
- whatsapp_phone - номер whatsapp для отправки заявок (обязательно к заполнению и сохранится только, если выбран type = 3)
- photo_description - описание для фотографий (сохранится только, если выбран type = 3)

- `moderation_on` - включить модерацию кастинга. Если передан 1, то кастинг будет проверен модератором, если 0 - то будет работать автомодерация. По умолчанию 0. Параметр только для проверенных кастинг-директоров.
- `reason_for_moderation` - описание, для чего кастинг-директор, хочет, чтобы кастинг проверил модератор, не обязательный. Параметр только для проверенных кастинг-директоров.

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 741 - кастинг не найден;
- 742 - кастинг с переданным `id` не принадлежит пользователю;
- 743 - нет активной подписки;
- 744 - кастинг находится на автомодерации;

7.5. Удалить кастинг

Запрос

Метод: `casting.delete`

Тип: GET

Требуется token: да

Параметры:

- `casting_id` - идентификатор кастинга (обязательно);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 751 - Кастинг не найден
- 752 - Кастинг с переданным `id` не принадлежит пользователю

7.6. Получить мои кастинги

Запрос

Метод: `casting.getListAsAuthor`

Тип: GET

Требуется token: да

Параметры:

- `offset` - смещение относительно первого найденного кастинга для выборки определенного подмножества;
- `count` - количество возвращаемых кастингов, по умолчанию 20, максимальное значение 1000, минимальное - 10;
- `is_active` - выбрать активные или неактивные кастинги;

- 1 - только активные
- 0 - только неактивные
- параметр не указан - все кастинги

Ответ

```
{
  response: {
    count: 343,
    items: [
      {
        "id": "1",
        "logo_url":
"http://api.kinopro365.com/uploads/fixture/photo/2.jpg",
        "name": "casting 1",
        "description": "text text",
        "country_name": "Russia",
        "city_name": "Moscow",
        "end_at": "1501000000",
        "project_type_id": "1",
        "casting_director_status": "Проверенный кастинг директор";
        "warning_text": "Возможно эро сцены";
        "review_status": "1",
        "review_message": "",
        "is_active": "1",
        "count_pending_offer": "2",
        "professions": [
          {
            "profession_id": 1,
            "casting_id": 1,
          },
          {
            "profession_id": 2,
            "casting_id": 1,
          },
          ...
        ],
      },
      ...
    ]
  }
}
```

Параметры	Описание
-----------	----------

count	количество элементов
items	
id	идентификатор кастинга
logo_url	ссылка на логотип
name	название кастинга
description	описание
country_name	страна
city_name	город
end_at	дата окончания приема заявок
project_type_id	тип проекта
casting_director_status	статус кастинг директора
warning_text	предупреждающий текст
review_status	статус модерации: 1 - ожидает модерацию; 2 - одобрен; 3 - отклонен;
review_message	причина отклонения
is_active	время подачи заявок не закончилось: 1 - активный; 0 - время подачи заявок закончилось;
count_pending_offer	количество заявок на рассмотрение
professions	
profession_id	идентификатор специализации
casting_id	идентификатор кастинга

7.7. Получить инфу о кастинге как создатель

Запрос

Метод: casting.getAsAuthor

Тип: GET

Требуется token: да

Параметры:

- casting_id - идентификатор кастинга (обязательно);

Ответ

```
{
  response: {
    "id": "1",
    "logo_url": "http://api.kinopro365.com/uploads/fixture/photo/2.jpg",
    "name": "casting 1",
    "description": "text text",
    "country_id": "2",
    "country_name": "Беларусь",
    "city_id": "4",
    "city_name": "Минск",
    "end_at": "1501000000",
    "project_type_id": "1",
    "profession_id": "4", // для версий < 1.6
    "contact_info": "casting_director@mail.ru, 8-800-900-01-02",
    "review_status": "1",
    "review_message": null,
    "age_from": "20",
    "age_to": "40",
    "sex": "2",
    "use_playing_age": true, // Учитывать игровой возраст
    "ex_bust": 3, // Бюст
    "ex_height_from": "170",
    "ex_height_to": "180",
    "ex_height_ft_from": "5.2",
    "ex_height_ft_to": "6.0",
    "ex_height_type": "1",
    "ex_hair_color": "1",
    "ex_eyes_color": "1",
    "ex_clothing_size": "8",
    "ex_vocals": null,
    "ex_fencing": "1",
    "ex_martial_stunt": null,
    "ex_acrobatics": null,
    "ex_car_stunt": null,
    "ex_moto_stunt": null,
    "ex_fiery_stunt": "1",
    "ex_fights_director": "0",
    "ex_appearance_type": "4",
    "count_offer": "3", // общее количество заявок
    "count_approved_offer": "2", // количество одобренных заявок
    "count_pending_offer": "2", // количество заявок в ожидании
    "cost": "1000.00", // цена - 2 знака после запятой,
    "not_commercial": false, // некоммерческий,
    "is_nude": false,
```

```

    "casting_director_status": "Проверенный кастинг директор";
    "warning_text": "Возможно эро сцены";
    "languages": [
        {
            "language_id": "3",
            "casting_id": "1"
        },
        ...
    ],
    "drivings": [
        {
            "driving_id": "3",
            "casting_id": "1"
        },
        ...
    ],
    "professions": [
        {
            "profession_id": 1,
            "casting_id": "1"
        },
        {
            "profession_id": 2,
            "casting_id": "1"
        },
        ...
    ],
    "type": 1,
    "requests_email": "request@test.com",
    "requests_email_subject": "На кастинг.",
    "whatsapp_phone": "+71234567788",
    "photo_description": "Test photo description",
    "is_on_auto_moderation": false, // находится ли кастинг на автомодерации
}
}

```

Коды ошибок:

- 771 - Кастинг не найден
- 772 - Кастинг с переданным id не принадлежит пользователю

7.8. Подать заявку на кастинг

Запрос

Метод: casting.apply

Тип: GET

Требуется token: да

Параметры:

- casting_id - идентификатор кастинга (обязательно);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 781 - кастинг не найден;
- 782 - нельзя подать заявку на свой кастинг;
- 783 - заявка уже была подана;
- 784 - нет активной подписки;
- 785 - на кастинги с типом "отправка заявок только на email" нельзя подавать заявки в приложении;
- 786 - окончился срок подачи заявок на кастинг.

7.9. Получить список предложений

Запрос**Метод:** casting.getOffers**Тип:** GET**Требуется token:** да**Параметры:**

- offset - смещение относительно первого найденной заявки для выборки определенного подмножества;
- count - количество возвращаемых заявок, по умолчанию 20, максимальное значение 1000, минимальное - 10;
- casting_id - идентификатор кастинга (обязательно);

Ответ

```
{
    response: {
        count: 233,
        items: [
            {
                "casting_offer_id": "8", // идентификатор заявки
                "id": "2", // идентификатор пользователя
                "first_name": "Джонни", // имя на языке пользователя;
                "last_name": "Депп", // фамилия на языке пользователя;
                "first_name_inter": "John", // имя на латинице;
                "last_name_inter": "Depp", // фамилия на латинице;
                "photo_url":
                    "http://api.kinopro365.com/uploads/fixture/photo/2.jpg", // ссылка на
                    фото пользователя;
                "sex": "1", // пол пользователя:
                    // - 1 - женский;
                    // - 2 - мужской;
            }
        ]
    }
}
```

```

        "country_name": "Россия", // страна;
        "city_name": "Москва", // город;
        "age": "53", // возраст;
        "height": "178", // рост;
        "height_ft": "5.6", // рост в футах;
        "height_type": 1 // размерность роста, 1 - см., 2 - фут.
        "email": "user2@example.com", // email
        "is_like": "1", // ставил или нет лайк пользователь
        "is_favourite": "0", // находится или нет в избранном
        "count_likes": "3", // количество лайков пользователя
        "lang": "en" // язык информации профиля
        "phones": [ // номера телефонов;
            {
                "phone_number": "+79519501001",
                "user_id": "2"
            },
            ...
        ]
    },
    ...
]
}
}
}

```

Коды ошибок:

- 791 - Кастинг не найден
- 792 - Кастинг с переданным id не принадлежит пользователю

7.10. Получить список одобренных

Запрос

Метод: casting.getApproved

Тип: GET

Требуется token: да

Параметры:

- offset - смещение относительно первого найденного кастинга для выборки определенного подмножества;
- count - количество возвращаемых кастингов, по умолчанию 20, максимальное значение 1000, минимальное - 10;
- casting_id - идентификатор кастинга (обязательно);

Ответ

```

{
    response: {
        count: 233,
        items: [

```

```

{
  "id": "2",
  "first_name": "Джонни", // имя на языке пользователя;
  "last_name": "Депп", // фамилия на языке пользователя;
  "first_name_inter": "John", // имя на латинице;
  "last_name_inter": "Depp", // фамилия на латинице;
  "photo_url":
"http://api.kinopro365.com/uploads/fixture/photo/2.jpg", // ссылка на
  фото пользователя;
  "sex": "1", // пол пользователя:
    // - 1 - женский;
    // - 2 - мужской;
  "country_name": "Россия", // страна;
  "city_name": "Москва", // город;
  "age": "53", // возраст;
  "height": "178", // рост;
  "height_ft": "5.6", // рост в футах;
  "height_type": 1 // размерность роста, 1 - см., 2 - фут.
  "is_like": "1", // ставил или нет лайк пользователь
  "is_favourite": "0", // находится или нет в избранном
  "count_likes": "3", // количество лайков пользователя
  "lang": "en" // язык информации профиля
  "phones": [ // номера телефонов;
    {
      "phone_number": "+79519501001",
      "user_id": "2"
    },
    ...
  ]
},
...
]
}
}

```

Коды ошибок:

- 7101 - Кастинг не найден
- 7102 - Кастинг с переданным id не принадлежит пользователю

7.11. Принятие решения на отклик

Запрос

Метод: casting.decide

Тип: GET

Требуется token: да

Параметры:

- casting_offer_id - идентификатор заявки (обязательно);
- decision - решение, возможные значения (обязательно):
 - 1 - принять;
 - 2 - отклонить;

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 7111 - Заявка не найдена
- 7112 - Заявка уже была рассмотрена
- 7113 - Заявка на несуществующий кастинг
- 7114 - Заявка на кастинг не принадлежащий пользователю

7.12. Создать жалобу на кастинг

Запрос

Метод: casting.createComplaint

Тип: POST

Требуется token: да

Параметры:

- to_casting_id - идентификатор пользователя на которого подается жалоба (обязателен);
- text - текст жалобы (обязателен);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 7121 - пользователь на которого подается жалоба не найдена;

7.13. Получить html код email рассылки

Запрос

Метод: casting.getHtmlEmail

Тип: GET

Требуется token: да

Параметры:

- casting_id - идентификатор кастинга (обязателен);

Ответ

```
{
    response: (html-код)
}
```

Коды ошибок:

- 7131 - Кастинг не найден;
- 7132 - Кастинг с переданным id не принадлежит пользователю.

7.14. Отменить автомодерацию кастинга

Запрос

Метод: casting.cancelAutoModeration

Тип: GET

Требуется token: да

Параметры:

- casting_id - идентификатор кастинга (обязателен);

Ответ

```
{
    response: 1
}
```

Коды ошибок:

- 7141 - Кастинг не найден;
- 7142 - Кастинг с переданным id не принадлежит пользователю;
- 7143 - Кастинг не является автомодерируемым;

8. Оповещения

8.1. Получить оповещения

Возвращает оповещения отсортированные по убыванию даты создания (первое - самое свежее оповещение).

Запрос

Метод: notification.getList

Тип: GET

Требуется token: да

Параметры:

- all - если 1, то вернет все оповещения, если 0, то только непрочитанные. По умолчанию 0;
- offset - смещение относительно первого оповещения для выборки определенного подмножества;
- count - количество возвращаемых оповещений, по умолчанию 20, максимальное значение 1000, минимальное - 10;

Ответ

```
{
  "response": {
    "count": "5", //
    "items": [
      {
        "id": "3",
        "type": 33,
        "is_read": 1,
        "created_at": "1480123123",
        "country_name": "Russia",
        "city_name": "Moscow",
        "params": {}
      },
      ...
    ]
  }
}
```

Параметры	Описание
count	количество элементов
items	

id	идентификатор оповещения
type	тип оповещения, описание значений приведены в разделе перечислений
created_at	дата создания оповещения
is_read	1 - если оповещение прочитано, 0 - если нет
params	доп. параметры в виде json объекта, описание доп. параметров смотри в перечислении типов оповещений
chain_id	идентификатор цепочки оповещений
country_name	локализованное название страны
city_name	локализованное название города

8.2. Пометить как прочитанное

Помечает оповещения как прочитанные.

Запрос

Метод: notification.markAsRead

Тип: GET

Требуется token: да

Параметры:

- notification_ids - идентификаторы оповещений через запятую
- notification_start_id - при передаче этого параметра будут помечены как прочитанные все оповещения, начиная с данного.

Передавать нужно либо notification_ids либо notification_start_id либо без параметров, (если нужно пометить все уведомления как прочитанные). Передача обоих параметров запрещена.

Ответ

```
{
  response: 1
}
```

Коды ошибок:

- 821 - Оповещение не найдено / ниодно оповещение не найдено (если передан список)
- 822 - Оповещение не принадлежит пользователю / оповещение id:'i' не принадлежит пользователю (если передан список)

8.3. Получить настройки оповещений

Запрос

Метод: notification.getSettings

Тип: GET

Требуется token: да

Параметры: нет

Ответ

```
{
  "response": {
    "feature_film": 1,
    "telecast": 1,
    "music_video": 1,
    "short_film": 1,
    "advertising": 1,
    "sound_and_dubbing": 1,
    "tv_series": 0,
    "reality_show": 1,
    "documentary": 1,
    "crowd_and_audience": 1,
    "promo_clip": 1,
    "social_video": 0,
    "theatrical_performance": 1,
    "photosession": 1,
    "cost_from": "1000.00",
    "cost_from_for_crowd_artist": "500.00",
    "without_cost": 1,
    "not_commercial": 1,
    "ready_for_business_trips": 1,
    "email_about_purchase": 1,
    "sms_about_purchase": 1,
    "push_about_purchase": 1,
    "only_partners_casting": 0,
  }
}
```

8.4. Сохранить настройки оповещений

Запрос

Метод: notification.saveSettings

Тип: GET

Требуется token: да

Параметры:

- feature_film
- telecast
- music_video
- short_film
- advertising

- sound_and_dubbing
- tv_series
- reality_show
- documentary
- crowd_and_audience
- promo_clip
- social_video
- theatrical_performance
- photosession
- cost_from - цена от - для специализаций "Актер, Актер любитель, Каскадер, Модель", число, 2 знака после запятой
- cost_from_for_crowd_artist - цена от - для специализации "Актер массовых сцен", число, 2 знака после запятой
- without_cost
- not_commercial
- ready_for_business_trips
- email_about_purchase
- sms_about_purchase
- push_about_purchase
- only_partners_casting

для всех параметров, кроме cost_from и cost_from_for_crowd_artist, 1 - получать оповещения/параметра включен, 0 - не получать оповещения/параметр выключен

ОТВЕТ

```
{
    response: 1
}
```

8.5. Отправить тестовое оповещение

Отправит текущему пользователю push уведомление.

Запрос

Метод: notification.test

Тип: GET

Требуется token: да

Параметры:

- id - идентификатор оповещения;

ОТВЕТ

```
{
    response: 1
}
```

8.6. Получить оповещение по chain_id

Запрос

Метод: notification.getByChainId

Тип: GET

Требуется token: да

Параметры:

- chain_id - идентификатор цепочки оповещений;

Ответ

```
{
  response: {
    "id": 1,
    "type": 12,
    "created_at": 1524722540,
    "is_read": false,
    "params":
      "{\"user_id\":null,\"user_first_name\":\"Администрация\",\"user_last_name\":\"Cin
      arta\",\"subscription_id\":2,\"count_days\":184}\",
    "chain_id": "testChainId1",
    "country_name": "Russia",
    "city_name": "Moscow"
  }
}
```

Параметры	Описание
id	идентификатор оповещения
type	тип оповещения, описание значений приведены в разделе перечислений
created_at	дата создания оповещения
is_read	1 - если оповещение прочитано, 0 - если нет
params	доп. параметры, описание параметров смотри в перечислении типов оповещений
chain_id	идентификатор цепочки оповещений
country_name	локализованное название страны
city_name	локализованное название города

9. Покупки

Общие коды ошибок для покупок:

- 901 - неверные данные покупки;
- 902 - покупка уже была использована;
- 903 - покупка была отменена;
- 904 - невалидный платёж;

9.1. Купить подписку

Запрос

Метод: purchase.buySubscription

Тип: POST

Требуется token: да

Параметры:

- subscription_id - идентификатор подписки:
 - 3 - подписка на год;
- user_id - идентификатор пользователя кому покупается подписка, для покупки себе этот параметр передавать не нужно;
- os_type - тип операционной системы, обязательный, возможны варианты:
 - 1 - Android;
 - 2 - iOS;
- purchase_info - данные покупки:
 - для Android в формате json;
 - для iOS - receipt полученный от iTunes.

Ответ

```
{
    response: 1488814825 // дата окончания оплаченных подписок;
}
```

Коды ошибок:

- 911 - невалидный платёж;
- 912 - несуществующий идентификатор подписки;
- 913 - несуществующий идентификатор типа операционной системы;
- 914 - передан несуществующий идентификатор пользователя;

9.2. Купить открытие контактов

Запрос

Метод: purchase.buyOpenContacts

Тип: POST

Требуется token: да

Параметры:

- user_id - идентификатор пользователя, контакты которого покупают;

- `os_type` - тип операционной системы, обязательный, возможны варианты:
 - 1 - Android;
 - 2 - iOS;
- `purchase_info` - данные покупки:
 - для Android в формате json;
 - для iOS - receipt полученный от iTunes.

Ответ

```
{
  "response": {
    "id": "6", // идентификатор пользователя, чьи контакты открыты
    "email": "user3@example.com",
    "phones": [
      {
        "user_id": "6",
        "phone_number": "+79515901007"
      },
      ...
    ],
  }
}
```

Коды ошибок:

- 921 - невалидный платёж;
- 922 - передан несуществующий идентификатор пользователя;
- 923 - несуществующий идентификатор типа операционной системы;
- 924 - контакты пользователя уже открыты;
- 925 - нельзя открывать себе свои контакты;
- 926 - контакты доступны только кастинг директорам;

9.3. Получить доступные для покупки подписки

Запрос

Метод: `purchase.getAvailableSubscriptions`

Тип: GET

Требуется token: да

Параметры:

- `os_type` - тип операционной системы, обязательный, возможны варианты:
 - 1 - Android;
 - 2 - iOS;

Ответ

```
{
  response:{
    "subscription_ids": [
      1, 2, ...
    ]
  }
}
```

```
    ]
  }
}
```

Параметры	Описание
subscription_ids	массив идентификаторов доступных подписок

Коды ошибок:

- 931 - несуществующий идентификатор типа операционной системы;

10. Html анкета

10.1. Сохранить фото для html анкеты

Запрос

Метод: html.savePhotos

Тип: POST

Требуется token: да

Параметры:

- photos[i] - массив идентификаторов фотографий, допустимо передавать не более 5;
- avatar_id - идентификатор фотографии аватара, данный идентификатор должен быть обязательно в массиве photos;

Ответ

```
{
  "response": 1
}
```

Коды ошибок:

- общие ошибки;
- 1011 - передано более 5 идентификаторов фотографий;
- 1012 - одно или несколько фото, идентификаторы которых переданы, загружены другим пользователем;
- 1013 - идентификатор аватара не передан в массиве photos;

10.2. Получить фото для html

Метод: html.getPhotos

Тип: GET

Требуется token: да

Параметры: нет;

Ответ

```
{
  "response": [
    {
      "id": 12, // идентификатор фотографии
      "is_avatar": false, // фото аватара или нет
      "url": "http://api.cinarta.com/.../2-1.jpg" // ссылка на фото
    },
    ...
  ]
}
```

Коды ошибок:

- общие ошибки;

10.3. Получить html своего профиля

Запрос**Метод:** html.getProfile**Тип:** GET**Требуется token:** да**Параметры:**

- withPrivateInfo - 0 - без личной информации, 1 - с личной информацией, по умолчанию 0;

Ответ

```
{  
    response: (html-код)  
}
```

Коды ошибок:

- общие коды ошибок

11. FAQ Типичные вопросы и ответы

11.1 Получить список категорийЗапрос

Метод: faq.getCategories

Тип: GET

Требуется token: нет

Параметры: нет

Ответ:

```
{
  "response": {
    "count": 4, // Количество записей.
    "items": // Массив записей
    [
      {
        "id": 2, // Id записи
        "name": "Основные", // Наименование
        "sort": 10 // Сортировка.
      },
      * * *
    ]
  }
}
```

11.2 Получить список вопросов в заданной кате

Запрос

Метод: faq.getByCategory

Тип: GET

Требуется token: нет

Параметры: categoryId - Id категории, для которой идет запрос

Ответ:

```
"response":
{
  "count":3,
  "items":
  [
    {
      "id":2, // Id записи.
      "category_id":1, // id категории
      "sort":10, // Позиция сортировки
      "question":"Вопрос", // Текст вопроса
    }
  ]
}
```

"answer": "Текст", // Текст ответа

"link": "http://cinarta.test/site/faq?id=2" // Ссылка на ответ на сайте.

}

* * *

]

}

Push уведомления

Для отправки push уведомлений используется сервис [Firebase Cloud Messaging](#).

Отправляемые параметры:

- data - параметры:
 - type - тип оповещения, возможные значения:
 - 10 - пользователь зарегистрировался;
 - 11 - пользователь оплатил аккаунт сам;
 - 12 - пользователю оплатил аккаунт другой пользователь;
 - 13 - истечение срока подписки через 2 дня;
 - 14 - истечении срока подписки;
 - 15 - активирована пробная подписка;
 - 16 - тестовая подписка не активирована;
 - 17 - пользователь отключил подписку;
 - 20 - вакансия прошла модерацию и опубликована;
 - 21 - вакансия не прошла модерацию;
 - 22 - подходящая вакансия для пользователя;
 - 30 - кастинг прошел модерацию и опубликован;
 - 31 - кастинг не прошел модерацию;
 - 32 - подходящий кастинг для пользователя;
 - 33 - пользователь утверждён на кастинг;
 - 35 - создан кастинг с авто модерацией;
 - 40 - отправка сообщения пользователю;
 - 41 - массовая рассылка сообщений;
 - 50 - запрос на открытие контактов;
 - 51 - пользователь открыл контакты;
 - 52 - пользователь отказался открывать контакты;
 - chainId - идентификатор оповещения, по этому идентификатору можно получить оповещение при помощи метода: *8.6. Получить оповещение по chain_id*;
- для iOS:
 - notification
 - body_loc_key - push_body_message_%type%;
 - body_loc_args - аргументы для вставки в локализованную строку;
 - title_loc_key - push_title_message_%type%;
 - title_loc_args - аргументы для вставки в локализованную строку;
- для Android;
 - notification - не присылается;

Общие коды ошибок

- 1 - неизвестная ошибка;
- 2 - передан неверный токен доступа;
- 3 - не верный запрос, проверьте список передаваемых параметров;
- 4 - метод устарел;
- 5 - API устарел, необходимо обновить приложение;
- 6 - метод не найден;
- 7 - превышен лимит запросов;
- 8 - неверный тип запроса;

Перечисления

Профессии

- 1 - Актёр
 - Рост (ex_height) - рост в сантиметрах;
 - Рост в футах (ex_height_ft);
 - Основная размерность роста (ex_height_type);
 - Цвет волос (ex_hair_color) - возможные значения смотри в перечислении *Цвет волос*;
 - Цвет глаз (ex_eyes_color), возможные значения смотри в перечислении *Цвет глаз*;
 - Размер одежды (ex_clothing_size), возможные значения смотри в перечислении: *Размер одежды*;
 - Вождение (ex_drivings[i]), возможно одно или несколько значений из перечисления *Вождение*;
 - Вокал (ex_vocals), возможно одно из значений перечисления *Вокал*;
 - Языки (ex_languages[i]), возможно одно или несколько из значений из перечисления *Языки*;
 - Игровой возраст “от” (playing_age_from) - значение нижнего предела игрового возраста
 - Игровой возраст “до” (playing_age_to) - значение верхнего предела игрового возраста
 - Бюст (ex_bust) - размер бюста, возможно одно из значения перечисления *Бюст*;
 - Тип внешности (ex_appearance_type) - идентификатор типа внешности, одно из возможных значений, полученных с помощью метода info.getAppearanceTypeValues;
- 2 - Актер без образования
 - Рост (ex_height) - рост в сантиметрах;
 - Рост в футах (ex_height_ft);
 - Основная размерность роста (ex_height_type);
 - Цвет волос (ex_hair_color) - возможные значения смотри в перечислении *Цвет волос*;
 - Цвет глаз (ex_eyes_color), возможные значения смотри в перечислении *Цвет глаз*;
 - Размер одежды (ex_clothing_size), возможные значения смотри в перечислении: *Размер одежды*;
 - Вождение (ex_drivings[i]), возможно одно или несколько значений из перечисления *Вождение*;
 - Вокал (ex_vocals), возможно одно из значений перечисления *Вокал*;
 - Языки (ex_languages[i]), возможно одно или несколько из значений из перечисления *Языки*;

- Игровой возраст “от” (playing_age_from) - значение нижнего предела игрового возраста
- Игровой возраст “до” (playing_age_to) - значение верхнего предела игрового возраста
- Бюст (ex_bust) - размер бюста, возможно одно из значения перечисления *Бюст*;
- Тип внешности (ex_appearance_type) - идентификатор типа внешности, одно из возможных значений, полученных с помощью метода info.getAppearanceTypeValues;
- 3 - Актер массовых сцен
 - Рост (ex_height) - рост в сантиметрах;
 - Рост в футах (ex_height_ft);
 - Основная размерность роста (ex_height_type);
 - Цвет волос (ex_hair_color) - возможные значения смотри в перечислении *Цвет волос*;
 - Цвет глаз (ex_eyes_color), возможные значения смотри в перечислении *Цвет глаз*;
 - Размер одежды (ex_clothing_size), возможные значения смотри в перечислении: *Размер одежды*;
 - Вождение (ex_drivings[i]), возможно одно или несколько значений из перечисления *Вождение*;
 - Вокал (ex_vocals), возможно одно из значений перечисления *Вокал*;
 - Языки (ex_languages[i]), возможно одно или несколько из значений из перечисления *Языки*;
 - Игровой возраст “от” (playing_age_from) - значение нижнего предела игрового возраста
 - Игровой возраст “до” (playing_age_to) - значение верхнего предела игрового возраста
 - Бюст (ex_bust) - размер бюста, возможно одно из значения перечисления *Бюст*;
 - Тип внешности (ex_appearance_type) - идентификатор типа внешности, одно из возможных значений, полученных с помощью метода info.getAppearanceTypeValues;
- 4 - Каскадер
 - Рост (ex_height) - рост в сантиметрах;
 - Рост в футах (ex_height_ft);
 - Основная размерность роста (ex_height_type);
 - Цвет волос (ex_hair_color) - возможные значения смотри в перечислении *Цвет волос*;
 - Цвет глаз (ex_eyes_color), возможные значения смотри в перечислении *Цвет глаз*;
 - Размер одежды (ex_clothing_size), возможные значения смотри в перечислении: *Размер одежды*;
 - Вождение (ex_drivings[i]), возможно одно или несколько значений из перечисления *Вождение*;
 - Вокал (ex_vocals), возможно одно из значений перечисления *Вокал*;

- Языки (ex_languages[i]), возможно одно или несколько из значений из перечисления *Языки*;
- Игровой возраст “от” (playing_age_from) - значение нижнего предела игрового возраста
- Игровой возраст “до” (playing_age_to) - значение верхнего предела игрового возраста
- Бюст (ex_bust) - размер бюста, возможно одно из значения перечисления *Бюст*;
- Тип внешности (ex_appearance_type) - идентификатор типа внешности, одно из возможных значений, полученных с помощью метода info.getAppearanceTypeValues;
- Фехтование (ex_fencing) - 1 - да, 0 - нет.
- Боевые трюки (ex_martial_stunt) - 1 - да, 0 - нет.
- Акробатика (паркур) (ex_acrobatics) - 1 - да, 0 - нет.
- Авто трюки (ex_car_stunt) - 1 - да, 0 - нет.
- Мото трюки (ex_moto_stunt) - 1 - да, 0 - нет.
- Трюки с огнем (ex_fiery_stunt) - 1 - да, 0 - нет.
- Постановщик боев (ex_fights_director) - 1 - да, 0 - нет.
- 5 - Режиссер
- 6 - Второй режиссер
- 7 - Второй режиссер планирование
- 8 - Режиссер CG
- 9 - Оператор постановщик
- 10 - Оператор
- 11 - Оператор Стедикам
- 12 - 1-й Ассистент – Механик
- 13 - 2-й Ассистент оператора (Хлопушка)
- 14 - Долли
- 15 - Фокус пулер
- 16 - Плейбек
- 17 - Сценарист
- 18 - Редактор
- 19 - Скрипт супервайзер
- 20 - Креативный продюсер
- 21 - Администратор
- 22 - Директор
- 23 - Исполнительный продюсер
- 24 - Линейный продюсер
- 25 - Кастинг директор
- 26 - Режиссер монтажа
- 27 - Монтажер
- 28 - Композитор
- 29 - Художник по костюмам
- 30 - Костюмер
- 31 - Художник по гриму
- 32 - Гример

- 33 - Ассистент по актерам
- 34 - Ассистент по массовке
- 35 - Водитель
- 36 - Буфет
- 37 - Художник постановщик
- 38 - Декоратор
- 39 - Гафер
- 40 - Осветитель
- 41 - Рабочий постановочного цеха
- 42 - Звукорежиссер
- 43 - Звукооператор
- 44 - Микрофонщик (Boom)
- 45 - Режиссёр звуковых эффектов
- 46 - Звукооформитель
- 47 - Реквизитор
- 48 - Супервайзер по спецэффектам
- 49 - Постановщик трюков
- 50 - Хореограф
- 51 - Менеджер по локациям
- 52 - Фотограф (портфолио)
- 53 - Showgeel (Портфолио)
- 54 - Модель
 - Рост (ex_height) - рост в сантиметрах;
 - Рост в футах (ex_height_ft);
 - Основная размерность роста (ex_height_type);
 - Цвет волос (ex_hair_color) - возможные значения смотри в перечислении *Цвет волос*;
 - Цвет глаз (ex_eyes_color), возможные значения смотри в перечислении *Цвет глаз*;
 - Размер одежды (ex_clothing_size), возможные значения смотри в перечислении: *Размер одежды*;
 - Вождение (ex_drivings[i]), возможно одно или несколько значений из перечисления *Вождение*;
 - Вокал (ex_vocals), возможно одно из значений перечисления *Вокал*;
 - Языки (ex_languages[i]), возможно одно или несколько из значений из перечисления *Языки*;
 - Игровой возраст “от” (playing_age_from) - значение нижнего предела игрового возраста
 - Игровой возраст “до” (playing_age_to) - значение верхнего предела игрового возраста
 - Бюст (ex_bust) - размер бюста, возможно одно из значения перечисления *Бюст*;
 - Тип внешности (ex_appearance_type) - идентификатор типа внешности, одно из возможных значений, полученных с помощью метода `info.getAppearanceTypeValues`;
 - Готов(а) на НЮ кастинги (ready_for_nude_casting) - 1 - да, 0 - нет.

Цвет волос

- 1 - Темные;
- 2 - Русые;
- 3 - Каштановые;
- 4 - Рыжие;
- 5 - Блонд;
- 6 - Седые;
- 7 - Темно-русые;
- 8 - Темно-каштановые;

Цвет глаз

- 1 - Зеленые;
- 2 - Серые;
- 3 - Голубые;
- 4 - Карие;
- 5 - Синие;
- 6 - Темно-зеленые;
- 7 - Темно-серые;
- 8 - Серо-голубые;

Размер одежды

- 1 - 30 - 32;
- 2 - 32 - 34;
- 3 - 34 - 36;
- 4 - 36 - 38;
- 5 - 38 - 40;
- 6 - 40 - 42;
- 7 - 42 - 44;
- 8 - 44 - 46;
- 9 - 46 - 48;
- 10 - 48 - 50;
- 11 - 50 - 52;
- 12 - 52 - 54;
- 13 - 54 - 56;
- 14 - 56 - 58;
- 15 - 58 - 60;
- 16 - 60 - 62;
- 17 - 62 - 64;

Вожделение

- 1 - Не вожу;

- 2 - Категория А;
- 3 - Категория В;
- 4 - Категория С;
- 5 - Категория D;
- 6 - Вожу без прав;

Вокал

- 1 - Не пою;
- 2 - Пою хорошо;
- 3 - Проф. уровень;
- 4 - Проф. с большим опытом;

Языки

- 1 - китайский
- 2 - испанский
- 3 - английский
- 4 - хинди
- 5 - арабский
- 6 - португальский
- 7 - бенгальский
- 8 - русский
- 9 - японский
- 10 - немецкий
- 11 - корейский
- 12 - французский
- 13 - турецкий
- 14 - тамильский
- 15 - урду
- 16 - итальянский
- 17 - малайский
- 18 - персидский
- 19 - польский
- 20 - нидерландский
- 21 - греческий
- 22 - украинский
- 23 - армянский
- 24 - грузинский
- 25 - казахский
- 26 - белорусский

Бюст

- 1 - А (1);
- 2 - В (2);

- 3 - C (3);
- 4 - D (4);
- 5 - E (5);
- 6 - E+ (5+);

Типы проектов кастингов

- 1 - Художественный фильм;
- 2 - Телепередача;
- 3 - Музыкальный видеоклип;
- 4 - Короткометражный фильм;
- 5 - Реклама;
- 6 - Озвучивание и дубляж;
- 7 - Телесериал;
- 8 - Реалити шоу;
- 9 - Документальный фильм;
- 10 - Массовка и зрители;
- 11 - Проморолик;
- 12 - Соц ролик;
- 13 - Театральная постановка;
- 14 - Фотосессия.

Типы оповещений

На втором уровне списка указаны значения json объекта в поле params.

- 10 - пользователь зарегистрировался;
- 11 - пользователь оплатил аккаунт сам;
 - subscription_id - идентификатор подписки;
 - count_days - количество дней, на которое был активирован аккаунт;
- 12 - пользователю оплатил аккаунт другой пользователь;
 - user_id - идентификатор пользователя оплативший подписку;
 - user_first_name - имя того кто оплатил;
 - user_last_name - фамилия того кто оплатил;
 - subscription_id - идентификатор подписки;
 - count_days - количество дней, на которое был активирован аккаунт;
- 13 - истечение срока подписки через 2 дня;
- 14 - истечение срока подписки;
- 15 - активация пробной подписки;
 - subscription_id - идентификатор подписки;
 - count_days - количество дней, на которое был активирован аккаунт;
- 16 - тестовая подписка не активирована
 - errors - массив с идентификаторами условий, которые нужно выполнить, для активации тестовой подписки (см. перечисление типов ошибок активации тестового периода);

- missing_params - массив с названиями незаполненных параметров (см. перечисление незаполненных дополнительных параметров)
- 17 - пользователь отключил подписку;
- 20 - вакансия прошла модерацию и опубликована;
 - id - идентификатор вакансии;
 - name - название вакансии;
- 21 - вакансия не прошла модерацию;
 - id - идентификатор вакансии;
 - message - причина отклонения вакансии;
- 22 - подходящая вакансия для пользователя;
 - id - идентификатор вакансии;
 - profession - название профессии;
 - description - описание вакансии;
 - city - город вакансии;
 - country - страна вакансии;
 - end_at - время завершения вакансии;
- 30 - кастинг прошел модерацию и опубликован;
 - id - идентификатор кастинга;
 - name - название кастинга;
- 31 - кастинг не прошел модерацию;
 - id - идентификатор кастинга;
 - message - причина отклонения кастинга;
- 32 - подходящий кастинг для пользователя;
 - id - идентификатор кастинга;
 - project_type - тип проекта кастинга;
 - name - название кастинга;
 - description - описание кастинга;
 - city - город кастинга;
 - country - страна кастинга;
 - end_at - время завершения кастинга;
- 33 - пользователь утвержден на кастинг;
 - id - идентификатор кастинга;
 - project_type - тип проекта кастинга;
 - name - название кастинга;
 - contact_info - контактная информация по кастингу;
 - description - описание кастинга;
 - city - город кастинга;
 - country - страна кастинга;
 - end_at - время завершения кастинга;
- 35 - создан кастинг с авто модерацией;
 - id - идентификатор кастинга;
 - name - название кастинга;
 - timestamp - timestamp авто модерации кастинга;
- 40 - личное сообщение;
 - message - сообщение;
 - html - сообщение в виде html-кода;

- 41 - массовое сообщение;
 - message - сообщение;
 - html - сообщение в виде html-кода;
- 50 - запрос на открытие контактов;
 - solution - решение пользователя:
 - null - запрос не отвечен;
 - 1 - разрешил;
 - 2 - отклонил;
 - by_user_id - идентификатор пользователя, запрашивающего контакты;
 - user_first_name - имя пользователя, запрашивающего контакты;
 - user_last_name - фамилия пользователя, запрашивающего контакты;
 - text - текст запроса;
- 51 - пользователь открыл контакты;
 - contacts_user_id - идентификатор пользователя, чьи контакты были запрошены;
 - user_first_name - имя пользователя, чьи контакты были запрошены;
 - user_last_name - фамилия пользователя, чьи контакты были запрошены;
 - profession_id - идентификатор профессии;
- 52 - пользователь отказался открывать контакты;
 - contacts_user_id - идентификатор пользователя, чьи контакты были запрошены;
 - user_first_name - имя пользователя, чьи контакты были запрошены;
 - user_last_name - фамилия пользователя, чьи контакты были запрошены;
- 61 - подписка будет автоматически продлена через 3 дня
- 62 - подписка автоматически продлена

Типы подписок

- 1 - подписка на 3 месяца;
- 2 - подписка на 6 месяцев;
- 3 - подписка на 1 год;
- 6 - пробная подписка на 2 недели (невозможно купить);
- 7 - пробная подписка на 3 недели (невозможно купить);
- 8 - пробная подписка на 4 недели (невозможно купить);

Типы кастингов

- 1 - обычный, с возможностью подать заявки;
- 2 - "отправка заявок только на email", заявки отправлять можно только на email;
- 3 - "отправка заявок только на whatsapp", заявки отправлять можно только на whatsapp;

Типы вакансий

- 1 - обычная, с возможностью подать заявки;

- 2 - “создана вне системы” - заявка не подается, информация для связи указывается в описании вакансии;

Типы ошибок активации тестового периода

- 1 - аватар не загружен;
- 2 - добавлено в профиль менее четырех фотографий;
- 3 - заполнены не все дополнительные параметры специализаций пользователя;

Незаполненные дополнительные параметры

- ex_height - рост;
- ex_hair_color - цвет волос;
- ex_eyes_color - цвет глаз;
- ex_clothing_size - размер одежды;
- playing_age_from - игровой возраст от;
- playing_age_to - игровой возраст до;
- ex_bust - бюст;
- ex_appearance_type - идентификатор типа внешности;
- ex_languages - языки;

Роли пользователей

- 1 - обычный пользователь;
- 2 - проверенный кастинг-директор;
- 3 - админ;
- 4 - партнер - обычный пользователь;
- 5 - партнер - проверенный кастинг-директор.

Статусы пользователей

- 0 - не зарегистрирован;
- 10 - зарегистрирован;
- 11 - частичная регистрация.

Пол

- 1 - женский;
- 2 - мужской.