

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет компьютерных наук
Департамент программной инженерии**

СОГЛАСОВАНО
Доцент департамента
программной инженерии
факультета компьютерных наук
кандидат экономических наук

_____/ Песоцкая Е.Ю.
«__» _____ 2015 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»

_____/ Шилов В.В.
«__» _____ 2015 г.

WEB-ПРИЛОЖЕНИЕ ДЛЯ ВЗАИМОДЕЙСТВИЯ СТУДЕНТОВ

Руководство программиста

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.509000-01 33 01-1-ЛУ

Исполнитель
Студент группы 105 ПИ НИУ ВШЭ
_____/ Лазаренко А.В.
«__» _____ 2015 г.

2015

| | |
|--------------|--|
| Подп. и дата | |
| Инв. № дубл. | |
| Взам. инв. № | |
| Подп. и дата | |
| Инв. № подл. | |

УТВЕРЖДЕНО

RU.17701729.509000-01 33 01-1 ЛУ

WEB-ПРИЛОЖЕНИЕ ДЛЯ ВЗАИМОДЕЙСТВИЯ СТУДЕНТОВ

Руководство программиста

RU.17701729.509000-01 33 01-1

Листов 19

| | | | | |
|--------------|--------------|--------------|--------------|--------------|
| Инв. № подл. | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
| | | | | |

2015

АННОТАЦИЯ

В данном документе описана API приложения и способы взаимодействия с ней. В API реализовано большинство ключевых функций приложения, позволяющих успешно взаимодействовать с другими пользователями.

В разделе «Назначение и условия применения API» описано назначение, функции, выполняемые API, а так же условия, необходимые для работы API.

В разделе «Характеристика API» описаны временные характеристики и режим работы API.

В разделе «Обращение к API» описаны пути взаимодействия с функциями API.

В разделе «Входные и выходные данные» описаны все объекты, использующиеся на входе и выходе функций API.

В разделе «Сообщения» описаны сообщения, которые программист может получить в ответе с платформы.

До ознакомления с настоящим документом настоятельно рекомендуется ознакомиться с описанием JSON формата (приложение 1) и терминологией (приложение 2)

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Име. № подл. | Подп. и дата | Взам. инв. № | Име. № дубл. | Подп. и дата |

СОДЕРЖАНИЕ

| | |
|---|----|
| 1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ API | 5 |
| 1.1. Назначение API | 5 |
| 1.2. Функции, выполняемые API | 5 |
| 1.3. Условия, необходимые для работы API | 5 |
| 1.3.1. Требования к устройству | 6 |
| 1.3.2. Требования к программному обеспечению | 6 |
| 2. ХАРАКТЕРИСТИКА API | 6 |
| 2.1. Временные характеристики | 6 |
| 2.2. Режим работы | 6 |
| 3. ОБРАЩЕНИЕ К API | 6 |
| 3.1. Регистрация пользователя в системе | 6 |
| 3.2. Авторизация пользователя в системе | 6 |
| 3.3. Добавление устройства пользователя в базу данных | 7 |
| 3.4. Получение информации о профиле пользователя | 7 |
| 3.5. Получение всех локаций, на которые подписан пользователь | 7 |
| 3.6. Получение информации о различных аукционах пользователя | 7 |
| 3.7. Загрузка аватарки пользователя | 7 |
| 3.8. Добавление информации о городе, фамилии, имени и университете пользователя | 7 |
| 3.9. Получение списка всех людей, на которых подписан пользователь | 8 |
| 3.10. Подписка на пользователя | 8 |
| 3.11. Отписка от пользователя | 8 |
| 3.12. Получение списка всех пользователей | 8 |
| 3.13. Получение списка всех доступных университетов | 8 |
| 3.14. Получение списка локаций университета | 8 |
| 3.15. Получение информации о локации университета | 8 |
| 3.16. Получение сообщений на форуме локации университета | 8 |
| 3.17. Отсылка сообщения на форум локации | 9 |
| 3.18. Отписка от локации | 9 |
| 3.19. Подписка на локацию | 9 |
| 3.20. Получение списка всех локаций | 9 |
| 3.21. Поиск аукциона | 9 |
| 3.22. Загрузка фотографии для аукциона или товара | 9 |
| 3.23. Получение списка активных аукционов | 9 |
| 3.24. Получение списка ставок на аукционе | 10 |
| 3.25. Создание аукциона | 10 |
| 3.26. Выбор лидера на аукционе | 10 |
| 3.27. Завершение аукциона | 10 |
| 3.28. Добавление ставки на аукцион | 10 |
| 4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ | 10 |

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Име. № подл. | Подп. и дата | Взам. инв. № | Име. № дубл. | Подп. и дата |

| | |
|---|----|
| 4.1. Объекты, используемые при обращении к API и получении ответов. | 10 |
| 4.1.1. Объект UserCredentials | 10 |
| 4.1.2. Объект TokenContainer | 11 |
| 4.1.3. Объект RegistrateResponse | 11 |
| 4.1.4. Объект DeviceAdder | 11 |
| 4.1.5. Объект StandartResponse | 11 |
| 4.1.6. Объект UserId | 11 |
| 4.1.7. Объект ProfileData | 11 |
| 4.1.8. Объект UsersAuctions | 12 |
| 4.1.9. Объект UserInfo | 12 |
| 4.1.10. Объект EditProfile | 12 |
| 4.1.11. Объект ShortHouseData | 12 |
| 4.1.12. Объект Holder | 13 |
| 4.1.13. Объект HoldersHouse | 13 |
| 4.1.14. Объект House | 13 |
| 4.1.15. Объект HouseInfo | 13 |
| 4.1.16. Объект AuctionInfo | 14 |
| 4.1.17. Объект CreateAuctionInfo | 14 |
| 4.1.18. Объект LeaderInfo | 14 |
| 4.1.19. Объект Id | 14 |
| 4.1.20. Объект SearchString | 14 |
| 4.1.21. Объект CreateAuctionRespond | 14 |
| 4.1.22. Объект ForumPost | 15 |
| 4.1.23. Объект Message | 15 |
| 4.1.24. Объект ProductInfo | 15 |
| 4.1.25. Объект Bid | 15 |
| 5. СООБЩЕНИЯ | 16 |
| 5.1. Ответ типа StandartResponse | 16 |
| 5.2. Ответ типа CreateAuctionRespond | 16 |
| ПРИЛОЖЕНИЕ 1. ОПИСАНИЕ ФОРМАТА JSON | 17 |
| ПРИЛОЖЕНИЕ 2. ТЕРМИНОЛОГИЯ | 18 |
| ПРИЛОЖЕНИЕ 3. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ | 19 |

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Изм. № подл. | Подп. и дата | Взам. инв. № | Изм. № дубл. | Подп. и дата |

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ API

1.1. Назначение API

API предназначена для обеспечения возможности взаимодействия с платформой сторонних разработчиков и написания собственных клиентских приложений.

1.2. Функции, выполняемые API

Ниже приведен список функций платформы, доступных через API:

- 1) Регистрация пользователя в системе;
- 2) Авторизация пользователя в системе;
- 3) Добавление устройства пользователя в базу данных (необходимо для получения push-уведомлений);
- 4) Получение информации о профиле пользователя;
- 5) Получение всех локаций, на которые подписан пользователь;
- 6) Получение информации о различных аукционах пользователя;
- 7) Загрузка аватарки пользователя;
- 8) Добавление информации о городе, фамилии, имени и обществе пользователя;
- 9) Получение списка всех людей, на которых подписан пользователь;
- 10) Подписка на пользователя;
- 11) Отписка от пользователя;
- 12) Получение списка всех пользователей;
- 13) Получение списка доступных университетов;
- 14) Получение списка локаций университета;
- 15) Получение информации о локации университета;
- 16) Получение сообщений на форуме локации университета;
- 17) Отсылка сообщения на форум локации;
- 18) Отписка от локации;
- 19) Подписка на локацию;
- 20) Получение списка всех локаций;
- 21) Поиск аукциона;
- 22) Загрузка фотографии для аукциона или товара;
- 23) Получение списка активных аукционов;
- 24) Получение списка ставок на аукционе;
- 25) Создание аукциона;
- 26) Выбор лидера на аукционе;
- 27) Завершение аукциона;
- 28) Добавление ставки на аукцион;

1.3. Условия, необходимые для работы API

Условия, описанные ниже, актуальны прежде всего для развертывания всего приложения, поскольку без него API работать не будет.

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Име. № подл. | Подп. и дата | Взам. инв. № | Име. № дубл. | Подп. и дата |

1.3.1. Требования к устройству

Необходима машина, со следующими техническими характеристиками:

- 1) Процессор не хуже, чем 64-битный, с частотой 1.4 GHz
- 2) Не менее, чем 512MB оперативной памяти
- 3) Не менее, чем 32 GB свободной памяти на жестком диске
- 4) Gigabit (10/100/1000baseT) Ethernet adapter
- 5) Необходим доступ к интернету

1.3.2. Требования к программному обеспечению

Необходим предустановленный Windows Server 2012 (или выше) + MS SQL.

2. ХАРАКТЕРИСТИКА API

2.1. Временные характеристики

Временные характеристики выполнения зависят от сервера, на котором развернуто приложения, от скорости соединения с интернетом и от количества пользователей, активных в данный момент на платформе.

2.2. Режим работы

После того, как приложение развернуто, API доступна на протяжении всего времени работы приложения. Он-лайн всегда.

3. ОБРАЩЕНИЕ К API

Далее следует описание обращения к функциям API, хронологический порядок совпадает с порядком в пункте 1.2. Все посылаемые и получаемые объекты имеют формат JSON.

3.1. Регистрация пользователя в системе

Посылается POST-запрос по адресу <http://cocktion.com/api/Account/Registrate>, в параметре запроса посылается объект UserCredentials (см. 4.1.1), в ответ API возвращает объект типа RegistrateResponse (см. 4.1.3).

3.2. Авторизация пользователя в системе

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Изм. № подл. | Подп. и дата | Взам. инв. № | Изм. № дубл. | Подп. и дата |

Посылается POST-запрос по адресу <http://cocktion.com/api/Account/Authenticate>, в параметре запроса посылается объект UserCredentials (см. 4.1.1). В ответ приходит объект типа TokenContainer (см. 4.1.2).

В поле Token объекта TokenContainer (см. 4.1.2) содержится ключ, который необходимо прикреплять во всех остальных запросах к API, в теле HTTP запроса, причем необходимо указывать в следующей форме: «Authorization: Bearer _token», где «_token» - ключ.

3.3. Добавление устройства пользователя в базу данных

Посылается POST запрос по адресу <http://cocktion.com/api/Account/AddDevice>, в параметре запроса посылается объект типа DeviceAdder (см. 4.1.4), в ответ возвращается объект типа StandartResponse (см. 4.1.5).

3.4. Получение информации о профиле пользователя

Посылается POST запрос по адресу <http://cocktion.com/api/Profile/GetInfo>, в параметре запроса посылается объект UserId (см. 4.1.6), в ответ возвращается объект ProfileData (см. 4.1.7)

3.5. Получение всех локаций, на которые подписан пользователь

Посылается POST-запрос по адресу <http://cocktion.com/api/Profile/GetMyHouses>, в параметрах посылать ничего не нужно. В ответ возвращается коллекция объектов типа ShortHouseData (см. 4.1.11)

3.6. Получение информации о различных аукционах пользователя

Посылается POST-запрос по адресу <http://cocktion.com/api/Profile/GetMyAuctions>, в параметрах ничего не посылается. Приходит информация об аукционах, которые можно сгруппировать следующим образом:

- 1) Активные аукционы пользователя
- 2) Завершенные аукционы пользователя
- 3) Аукционы в локациях, на которые подписан пользователь
- 4) Все аукционы, на которых пользователь сделал ставку

Возвращается объект типа UsersAuctions (см. 4.1.8)

3.7. Загрузка аватарки пользователя

Посылается POST запрос (в теле запроса - фотография) по адресу <http://cocktion.com/api/Profile/UploadProfilePhoto>. Вернет объект типа StandartResponse(см. 4.1.5)

3.8. Добавление информации о городе, фамилии, имени и университете пользователя

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

Посылается POST запрос по адресу <http://cocktion.com/api/Profile/EditProfile>. В параметре запроса посылается объект класса EditProfile (см. 4.1.10). Если пользователь ничего не ввел в некоторых полях - отправляем пустые поля в тех местах, где он не ввел. Возвращает объект класса StandartResponse (см. 4.1.5).

3.9. Получение списка всех людей, на которых подписан пользователь

Посылается POST-запрос по адресу <http://cocktion.com/api/Profile/GetInformators>, в параметрах запроса ничего указывать не надо. Возвращает коллекцию объектов класса UserInfo (см. 4.1.9)

3.10. Подписка на пользователя

Посылается POST-запрос по адресу <http://cocktion.com/api/Users/SubscribeOnUser>, в параметрах посылается объект класса UserId (см. 4.1.6), возвращается объект класса StandartResponse (см. 4.1.5).

3.11. Отписка от пользователя

Посылается POST-запрос по адресу <http://cocktion.com/api/Users/UnsubscribeFromUser>, в параметрах посылается объект класса UserId (см. 4.1.6), вернет объект класса StandartResponse (см. 4.1.5).

3.12. Получение списка всех пользователей

Посылается POST-запрос по адресу <http://cocktion.com/api/Users/GetAllUsers>, в параметрах указывать ничего не надо, вернет коллекцию объектов класса UserInfo (см. 4.1.9)

3.13. Получение списка всех доступных университетов

Посылается POST-запрос по адресу <http://cocktion.com/api/House/GetGuilds>, в параметрах ничего посылать не надо, вернет коллекцию объектов класса Holder (см. 4.1.12)

3.14. Получение списка локаций университета

Посылается POST-запрос по адресу <http://cocktion.com/api/House/GetGuildsHouses>, в параметрах посылается объект класса Id (см. 4.1.19), возвращает коллекцию объектов класса HoldersHouse (см. 4.1.13)

3.15. Получение информации о локации университета

Посылается POST-запрос по адресу <http://cocktion.com/api/House/GetHouse>, в параметрах объект класса Id (см. 4.1.19), возвращает коллекцию объектов класса House (см. 4.1.14).

3.16. Получение сообщений на форуме локации университета

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

Посылается POST запрос по адресу <http://cocktion.com/api/House/GetHouseForumPosts>, в параметрах посылаем объект класса Id (см. 4.1.19), возвращает коллекцию объектов класса ForumPost (см. 4.1.22).

3.17. Отсылка сообщения на форум локации

Посылается POST-запрос по адресу <http://cocktion.com/api/House/SendPost>, в параметрах посылаем объект класса Message (см. 4.1.23), вернет объект класса StandartResponse (см. 4.1.5).

3.18. Отписка от локации

Посылается POST-запрос по адресу <http://cocktion.com/api/House/UnsubscribeFromHouse>, в параметрах посылаем объект класса Id (см. 4.1.19), возвращает назад объект StandartResponse (см. 4.1.5).

3.19. Подписка на локацию

Посылается POST-запрос по адресу <http://cocktion.com/api/House/SubscribeOnHouse>, в параметрах посылаем объект класса Id (см. 4.1.19), возвращает объект класса StandartResponse (см. 4.1.5).

3.20. Получение списка всех локаций

Посылается POST-запрос по адресу <http://cocktion.com/api/Data/GetHouses>, в параметрах ничего указывать не надо, возвращает коллекцию объектов HouseInfo (см. 4.1.15).

3.21. Поиск аукциона

Посылается POST-запрос по адресу <http://cocktion.com/api/Data/FindAuctions>, в параметре посылается объект класса SearchString (см. 4.1.20), возвращает коллекцию объектов AuctionInfo (см. 4.1.16).

3.22. Загрузка фотографии для аукциона или товара

Посылается POST-запрос по адресу <http://cocktion.com/api/Data/UploadPhoto>, в теле запроса посылается поле type, в котором указываем auction, если фотография предназначена для аукциона и product, если для добавления ставки на аукцион. Вернет объект класса Id, с айдишником фотки из базы для прикрепления к товару в запросе, либо -1, если по какой-то причине загрузка не удалась.

3.23. Получение списка активных аукционов

Посылается POST-запрос по адресу <http://cocktion.com/api/GetActiveAuctions>, никакие параметры не нужны. Вернет коллекцию объектов AuctionInfo (см. 4.1.16).

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

3.24. Получение списка ставок на аукционе

Посылается POST-запрос по адресу <http://cocktion.com/api/GetAuctionBids>, в параметре посылается объект класса Id (см. 4.1.19), возвращает коллекцию объектов ProductInfo (см. 4.1.24).

3.25. Создание аукциона

Посылается POST-запрос по адресу <http://cocktion.com/api/CreateAuction>, в параметрах посылается объект класса CreateAuctionInfo (см. 4.1.17), возвращает объект класса CreateAuctionRespond (см. 4.1.21).

3.26. Выбор лидера на аукционе

Посылается POST-запрос по адресу <http://cocktion.com/api/ChooseLeader>, в параметрах посылается объект класса LeaderInfo (см. 4.1.18), возвращает объект класса StandartRespond (см. 4.1.5).

3.27. Завершение аукциона

Посылается POST-запрос по адресу <http://cocktion.com/api/EndAuction>, в параметрах посылается объект класса Id (см. 4.1.19), вернет объект класса StandartRespond (см. 4.1.5)

3.28 Добавление ставки на аукцион

Посылается POST-запрос по адресу <http://cocktion.com/api/AddBid>, в параметрах посылается объект класса Bid (см. 4.1.25), вернет в ответ объект класса StandartRespond (см. 4.1.5).

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

4.1. Объекты, используемые при обращении к API и получении ответов.

Формат всех входных данных имеет JSON формат, за исключением фотографий. Далее последует описание всех объектов, которые посылаются на вход API для взаимодействия с платформой.

4.1.1. Объект UserCredentials

Данный тип используется для регистрации и авторизации на платформе. Имеет следующий набор полей:

string email - представление почтового адреса пользователя

string password - представление пароля

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

4.1.2. Объект TokenContainer

Данный тип используется для передачи токена. Имеет единственное поле:
string Token - представление самого токена

4.1.3. Объект RegistrateResponse

Данный тип возвращается в ответ при регистрации нового пользователя. Имеет следующие поля:

string Status - статус регистрации, принимает значение «Success», если регистрация прошла успешно, «Failure», если регистрация не удалась.

string Message - содержит сообщение с типом ошибки, либо null, если все прошло удачно.

4.1.4. Объект DeviceAdder

Данный тип используется для добавления устройства в базу данных. Устройство добавляется с целью подписки на push-уведомления (в данной версии реализовано только для iOS). Имеет следующие поля:

string type - тип устройства, может принимать значения (ios/android/wp).

string token - id устройства, полученный из приложения для подписки на уведомления.

4.1.5. Объект StandartResponse

Данный тип используется в качестве стандартного ответа сервера. Содержит единственное поле -

string Status - принимает значения «Success», если результат действия успешен, «Failure» в обратном случае.

4.1.6. Объект UserId

Данный тип используется для передачи id пользователя. Имеет единственное поле - string id, в этом поле указывается уникальный идентификатор пользователя, о котором надо получить информацию, либо «self», если нужно получить информацию о данном пользователе.

4.1.7. Объект ProfileData

Данный тип используется для получения информации о пользователе. Имеет следующий набор полей:

string title - имя пользователя

string surname - фамилия пользователя

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

string login - логин пользователя (он же имейл)

int rating - рейтинг пользователя

int eggs - количество яиц пользователя

int auctionsAmount - количество аукционов, которые пользователь сделал

int productsAmount - количество товаров, которые пользователь ставил

string city - город пользователя

string society - университет пользователя

string photoPath - путь к аватарке пользователя

В полях title, surname, city, society записано none, если пользователь не предоставил данные.

4.1.8. Объект UsersAuctions

Данный тип используется для передачи информации об аукционах пользователя. Имеет следующие поля:

List<AuctionInfo> active - все активные аукционы пользователя

List<AuctionInfo> finished - все законченные аукционы пользователя

List<AuctionInfo> houses - аукционы во всех домах, на которые подписан пользователь

List<AuctionInfo> products - аукционы, на которых пользователь ставил ставки

AuctionInfo - смотреть пункт 4.1.16

4.1.9. Объект UserInfo

Данный тип используется для передачи информации о пользователе. Имеет следующие поля:

string id - идентификатор пользователя

string title - имя пользователя

string photoPath - путь к аватарке пользователя

bool isInformator - индикатор «Подписаны ли вы на этого пользователя»

4.1.10. Объект EditProfile

Данный тип используется для изменения информации о пользователе, имеет следующий набор полей:

string title - имя пользователя

string surname - фамилия пользователя

string society - университет пользователя

string city - город пользователя

4.1.11. Объект ShortHouseData

Данный тип используется в качестве контейнера для небольшого количества информации о локации. Имеет следующие поля:

int id - номер дома в базе данных

string holderName - университет, к которому относится локация

string houseName - название локации

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

string photoPath - путь к фотографии дома

4.1.12. Объект Holder

Данный объект используется в качестве контейнера для небольшого количества информации об университете. Имеет следующие поля:

string title - название университета

string photoPath - путь к фотографии университета

int id - номер университета в базе данных

4.1.13. Объект HoldersHouse

Данный тип используется для передачи информации о локации. Имеет следующие поля:

string title - название локации

int id - номер в базе данных

string adress - адрес

string photoPath - путь к фотке

bool isSubscribed - подписан ли пользователь на данную локацию.

4.1.14 Объект House

Данный тип используется в качестве контейнера для информации о локации. Имеет следующие поля:

string photoPath - путь к фотографии

int likes - количество лайков

int dislikes - количество дизлайков

int rating - рейтинг локации

int peopleAmount - количество подписчиков

int auctionsAmount - количество аукционов

string description - описание

bool isSubscribed - подписан ли пользователь на локацию

string title - название локации

4.1.15 Объект HouseInfo

Данный объект используется в качестве контейнера для информации о локации.

Имеет следующие поля:

string title - название локации

string adress - адрес локации

int rating - рейтинг локации

int likes - лайки локации

bool isSubscribed - подписан ли пользователь на данную локацию

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Име. № подл. | Подп. и дата | Взам. инв. № | Име. № дубл. | Подп. и дата |

4.1.16 Объект AuctionInfo

Данный объект используется в качестве контейнера для информации об аукционе. Имеет следующий набор полей:

string description - описание товара
 string title - название товара
 string photoPath - путь к фотке
 int endTime - время окончания аукциона
 int auctionId - айдишник аукциона в базе данных
 int leaderId - айдишник лидера
 string category - категория, к которой относится товар
 bool isActive - активен ли в данный момент
 UserInfo owner - владелец аукциона

4.1.17 Объект CreateAuctionInfo

Данный объект используется для создания аукциона. Имеет следующие поля:

string title - название товара
 string description - описание товара
 int photoId - айдишник фотки, которую вернул сервер после загрузки
 string timeBound - временная граница, может принимать следующие значения: «4hoursTime» - 4 часа, «12hoursTime» - 12 часов, «1dayTime» - 1 день, «1weekTime» - 1 неделя.

4.1.18 Объект LeaderInfo

Данный тип используется для отправки информации о лидере. Имеет следующие поля:

int auctionId - айдишник аукциона, на котором надо выбрать лидера
 int productId - айдишник товара, который надо сделать лидером

4.1.19 Объект Id

Данный тип используется в качестве контейнера для айдишника.

int id - айдишник

4.1.20. Объект SearchString

Моделирует строку, посылаемую в поисковый запрос. Имеет единственное поле: string name - название аукциона, который надо найти.

4.1.21. Объект CreateAuctionRespond

| | | | | |
|-------------------------------|--------------|--------------|---------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Инов. № подл. | Подп. и дата | Взам. инв. № | Инов. № дубл. | Подп. и дата |

Используется в качестве ответа сервера на запрос о создании аукциона. Имеет следующий набор полей:

string PhotoPath - путь к фотке
string Status - статус

4.1.22. Объект ForumPost

Используется в качестве сообщения на форуме локации. Имеет следующий набор полей:

string authorName - имя автора
string message - само сообщение
int likes - количество лайков

4.1.23. Объект Message

Используется в качестве сообщения. Имеет следующий набор полей:

string message - само сообщение
int id - айдишник в базе данных.

4.1.24. Объект ProductInfo

Используется в качестве контейнера для информации о товаре. Имеет следующий набор полей:

string title - название товара
string description - описание товара
string photoPath - путь к фотографии товара
int id - айдишник в базе данных
string category - категория товара

4.1.25. Объект Bid

Используется в качестве контейнера для ставки. Имеет следующие поля:

string title - название вещи
string description - описание вещи
string category - категория, к которой относится вещь
int auctionId - айдишник аукциона, к которому относится вещь
int photoId - айдишник фотки, (берется из запроса к загрузке фотографии)

| | | | | |
|-------------------------------|--------------|--------------|---------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Инов. № подл. | Подп. и дата | Взам. инв. № | Инов. № дубл. | Подп. и дата |

5. СООБЩЕНИЯ

5.1. Ответ типа StandartResponse

Если в поле Status этого объекта пришло значение «Success» - проводимая операция завершена успешно, а предпринимать ничего не нужно, если же «Failure», то операция завершилась с ошибкой, а значит, нужно либо повторить попытку, либо исправить исходные данные у себя.

5.2. Ответ типа CreateAuctionRespond

Если в поле Status значение «Success», то аукцион добавился успешно, если же «Failure», то произошла какая-то ошибка и стоит повторить попытку, либо проверить свои исходные данные на корректность.

| | | | | |
|-------------------------------|--------------|--------------|---------------|--------------|
| | | | | |
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Инов. № подл. | Подп. и дата | Взам. инв. № | Инов. № дубл. | Подп. и дата |

ОПИСАНИЕ ФОРМАТА JSON

Ниже приведено описание формата JSON [1].

JSON (с английского JavaScript Object Notation) - текстовый формат, использующийся для обмена данными, основанный на JavaScript. За счет своей лаконичности используется для сериализации сложных структур.

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

1) Набор пар ключ : значение.

2) Упорядоченный набор значений.

В качестве значений в JSON используются структуры:

1) Объект - неупорядоченное множество пар ключ:значение, заключенное в фигурные скобки {}. Ключ описывается строкой, между ним и значением стоит символ .

2) Массив (одномерный) - упорядоченное множество значений. Массив заключается в квадратные скобки «[]»

3) Значение может быть строкой в двойных кавычках, числом, объектом, массивом, одним из литералов: true, false, null.

4) Строка - упорядоченное множество из нуля или более символов юникода, заключенное в двойные кавычки. Символы могут быть указаны с эскейп последовательностями.

Пример:

```
{  
    «name»: «Иван»,  
    «surname»:»Иванов»  
    «haveBaby»:true  
}
```

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Изм. № подл. | Подл. и дата | Взам. инв. № | Изм. № дубл. | Подл. и дата |

ТЕРМИНОЛОГИЯ

Далее приведен список терминов, фигурирующих в данном документе.

API (интерфейс программирования приложений, интерфейс прикладного программирования) (англ. application programming interface) - набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах [3]

Регистрация - создание аккаунта пользователя на платформе.

Авторизация - предоставлению пользователю права на выполнение определенных действий на платформе.

PUSH-уведомление - один из способов распространения информации (контента) в Интернете, когда данные поступают от сервера к клиенту на основе ряда параметров, установленных клиентом, где запрос на передачу информации происходит по инициативе центрального сервера. [4]

Локация - факультет или общежитие, относящееся к конкретному университету.

Аукцион - формат взаимодействия студентов, при котором они в режиме реального времени имеют возможность обмениваться вещами и обсуждать обмен.

Аватарка - фотография пользователя.

Подписка - добавление пользователя или локации в список, на основании которого будет производиться отбор аукционов для отображения пользователю.

Отписка - изъятие пользователя или локации из списка, на основании которого будет производиться отбор аукционов для отображения пользователю.

Форум - компонент приложения для организации общения пользователей.

POST-запрос - тип запроса, при котором веб-сервер принимает данные, заключенные в тело сообщения, для хранения. [5]

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Име. № подл. | Подп. и дата | Взам. инв. № | Име. № дубл. | Подп. и дата |

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) JSON [Электронный ресурс] //URL.: <https://ru.wikipedia.org/wiki/JSON> (Дата обращения: 16.05.2015, режим доступа: свободный)
- 2) Installing Windows Server 2012 [Электронный ресурс] //URL.: <https://technet.microsoft.com/en-us/library/jj134246.aspx> (Дата обращения 15.05.2015, режим доступа: свободный)
- 3) API [Электронный ресурс] //URL.: <https://ru.wikipedia.org/wiki/API> (Дата обращения 16.05.2015, режим доступа: свободный)
- 4) Технология Push [Электронный ресурс] //URL.: https://ru.wikipedia.org/wiki/Технология_Push (Дата обращения 16.05.2015, режим доступа: свободный)
- 5) POST (HTTP) [Электронный ресурс] //URL.: [https://ru.wikipedia.org/wiki/POST_\(HTTP\)](https://ru.wikipedia.org/wiki/POST_(HTTP)) (Дата обращения 16.05.2015, режим доступа: свободный)

| | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|
| Изм. | Лист | № докум. | Подп. | Дата |
| RU.17701729.509000-01 33 01-1 | | | | |
| Ине. № подл. | Подп. и дата | Взам. инв. № | Ине. № дубл. | Подп. и дата |

[illegible]