

УТВЕРЖДЕН
RU.17701729.05.04-01 81 01-1-ЛУ

ДЕСКТОПНОЕ ПРИЛОЖЕНИЕ ДЛЯ РАБОТЫ С АРІ НИУ ВШЭ

Пояснительная записка

RU.17701729.05.04-01 81 01-1

Листов 19

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
RU.17701729.05.04-01 81 01-1				

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ГЛОССАРИЙ.....	4
1. ВВЕДЕНИЕ.....	5
1.1. Наименование программы.....	5
1.2. Основания для разработки.....	5
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	6
2.1. Назначение программы.....	6
2.1.1. Функциональное назначение	6
2.1.2. Эксплуатационное назначение	6
2.2. Краткая характеристика области применения.....	6
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	7
3.1. Постановка задачи на разработку программы.....	7
3.2. Описание алгоритма программы.....	7
3.2.1. Процедуры при запуске.....	7
3.2.2. Стандартное меню	7
3.2.3. Страница авторизации.....	7
3.2.4. Страница собственного расписания.....	7
3.2.5. Страница поиска	7
3.2.6. Страница информации о сущностях	7
3.3. Описание Архитектуры.....	8
3.3.1. Общая архитектура	8
3.3.2. Архитектура серверной части	10
3.3.3. Архитектура клиентской части	11
3.4. Описание и обоснование выбора метода организации входных и выходных данных 11	
3.4.1. Авторизация	11
3.5. Просмотр собственного расписания.....	12
3.5.1. Поиск студентов, сотрудников, групп и аудиторий.....	12
3.5.2. Страницы информации о сущностях	12
3.5.3. Страница поиска свободных аудиторий.....	12
3.6. Описание и обоснование выбора состава технических и программных средств.....	12

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.	ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	14
4.1.	Предполагаемая потребность	14
4.2.	Экономические преимущества разработки по сравнению с аналогами	14
5.	ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ.....	15
	ПРИЛОЖЕНИЕ 1	16
	ПРИЛОЖЕНИЕ 2	17
	ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	19

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ГЛОССАРИЙ

1. НИУ ВШЭ – Национальный исследовательский университет «Высшая школа экономики»
2. Пользователь – студент или сотрудник НИУ ВШЭ, использующий приложение
3. Фреймворк – это набор инструментов, правил и готовых частей программы, который помогает разработчикам создавать приложения быстрее и проще, поскольку они могут использовать уже готовые решения вместо того, чтобы писать всё с нуля. Представьте, что вы собираете конструктор: фреймворк – это набор деталей и инструкция, которые говорят, как из этих деталей собрать определённую модель.
4. API (Application Programming Interface) — это набор правил, протоколов и инструментов для создания программного обеспечения, который позволяет разным приложениям и системам взаимодействовать между собой.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование разрабатываемой программы: «Десктопное приложение для работы с API НИУ ВШЭ»

Наименование разрабатываемой программы на английском языке: «Desktop Application to Work with the NRU HSE API»

1.2. Основания для разработки

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта — «Десктопное приложение для работы с API НИУ ВШЭ».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение программы

2.1.1. Функциональное назначение

Программа предоставляет пользователям инструменты для просмотра расписания занятий студентов и преподавателей и загруженности аудиторий НИУ ВШЭ.

2.1.2. Эксплуатационное назначение

Программа направлена на упорядочивание и агрегацию информации о расписании персонала, студентов, групп, загруженности аудиторий НИУ ВШЭ.

Конечными пользователями программы являются студенты и сотрудники НИУ ВШЭ, использующие персональный компьютер в качестве основного рабочего устройства.

2.2. Краткая характеристика области применения

«Десктопное приложение для работы с API НИУ ВШЭ» — это приложение для просмотра расписания занятий в НИУ ВШЭ, а также поиска служебной информации о студентах, сотрудниках, свободных аудиториях НИУ ВШЭ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Разрабатываемый программный продукт должен являться приложением для персональных компьютеров, направленным на поиск, агрегацию, структурирование и просмотр служебной информации о студентах, сотрудниках, группах, расписании занятий и загруженности аудиторий НИУ ВШЭ.

3.2. Описание алгоритма программы

3.2.1. Процедуры при запуске

При входе в приложение, производится проверка авторизационных данных на актуальность:

1. Если ключ доступа не устарел, пользователь незамедлительно получает доступ к приложению, а также запрашивается основная информация о пользователе для отображения имени, почты и аватара в верхней части меню.
2. Если ключ доступа устарел, а ключ обновления актуален, он используется для актуализации ключа доступа, после чего повторяется шаг 1.
3. Если ключ доступа и ключ обновления устарели, то пользователя переносит на страницу авторизации.

3.2.2. Стандартное меню

Меню представляет собой выезжающий список, который расположен с левой стороны приложения. Меню включает в себя три кнопки, нажатие на которые переносит на соответствующую страницу. Кнопки направляют на следующие страницы: собственное расписание, поиск, поиск свободных аудиторий.

3.2.3. Страница авторизации

При нажатии на кнопку «войти» пользователя переносит на внешний ресурс авторизации НИУ ВШЭ. После авторизации пользователя переносит обратно в приложение, а данные шифруются и сохраняются. Пользователь получает доступ к приложению и переносится на страницу собственного расписания.

3.2.4. Страница собственного расписания

При попадании на страницу, приложение автоматически посылает запрос к API НИУ ВШЭ, получая расписание пользователя на ближайшие две недели. Данные группируются по датам и отображаются пользователю.

3.2.5. Страница поиска

Пользователь вводит в строку поиска данные и выбирает категорию, когда пользователь перестает печатать, приложение посылает запрос к API НИУ ВШЭ. Из полученных данных, результатов поиска, извлекается основная информация и в списке отображается пользователю.

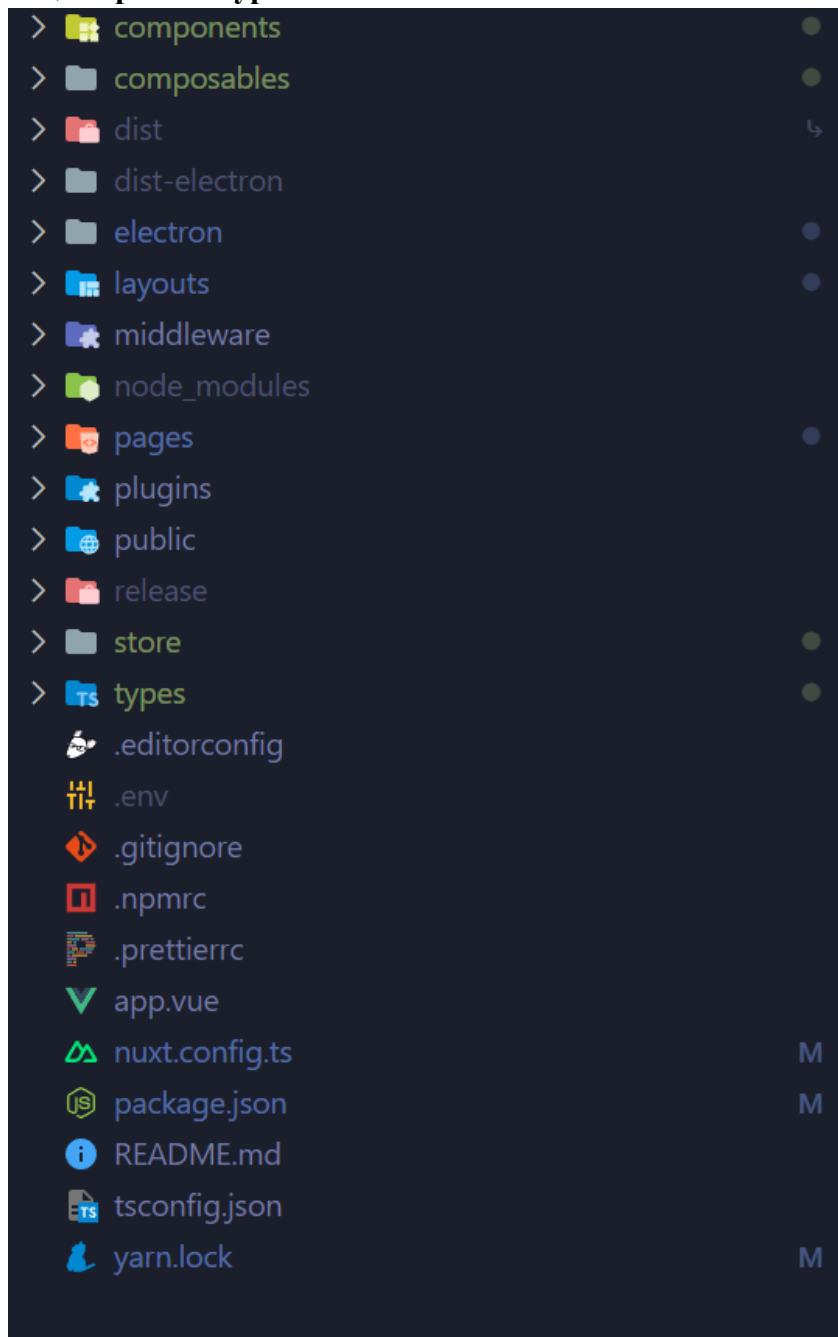
3.2.6. Страница информации о сущностях

При попадании на страницу приложение посылает запрос к API НИУ ВШЭ, получая информацию о студенте или преподавателе. Данные формируются и структурируются приложением для читаемого вывода.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание Архитектуры

3.3.1. Общая архитектура



components – папка, содержащая интерфейсные компоненты

composables – папка, содержащая вычисляемые свойства

dist, dist-electron – папки, содержащие служебные, промежуточные результаты компиляции

electron – папка содержащая логический слой приложения

layouts – папка, содержащая шаблоны основной разметки интерфейса

middleware – папка, содержащая функции для проверки авторизации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

node_modules – служебная папка с необходимыми библиотеками

public – служебная папка

plugins – папка, содержащая плагины для nuxt.js

pages – папка, содержащая страницы приложения

store – папка, содержащая инициализацию хранилищ состояния

types – папка, содержащая глобальные типы данных

.editorconfig – служебный файл для конфигурации среды разработки

.env – служебный файл для переменных окружения

.gitignore – служебный файл системы Git

.npmrc – файл конфигурации пакетного менеджера, npm

.prettierrc – служебный файл для конфигурации среды разработки

nuxt.config.js – файл, конфигурации управляющего фреймворка nuxt.js

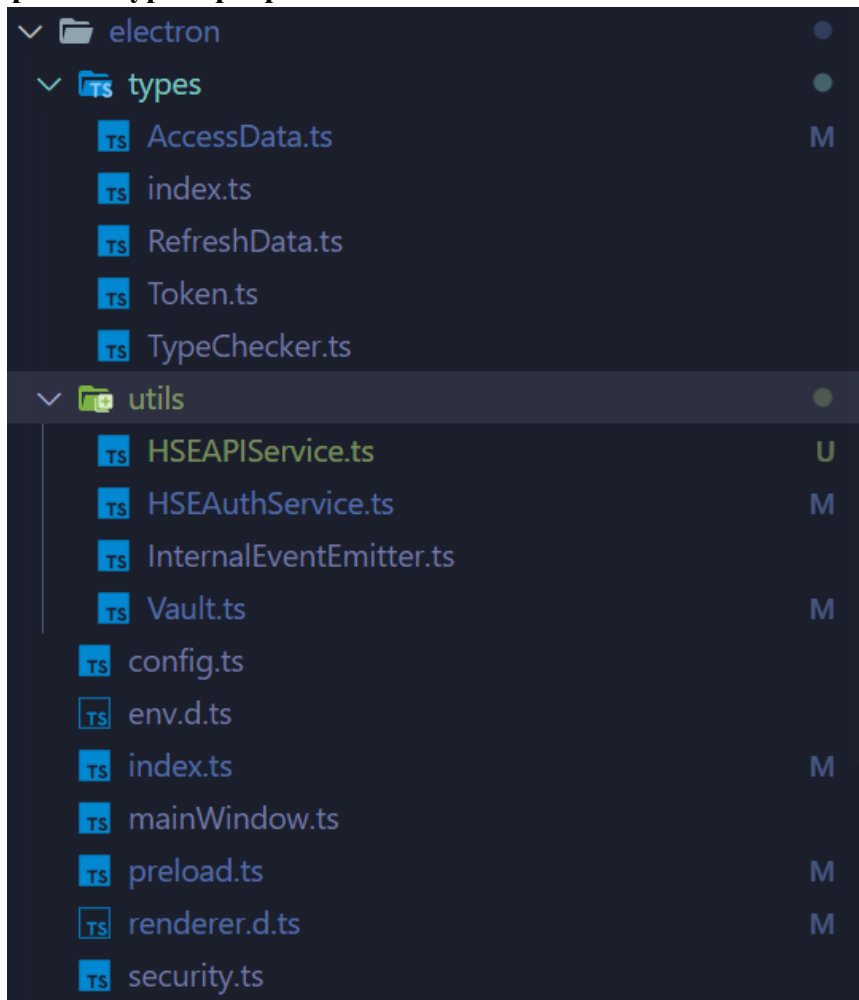
package.json – файл, содержащий информацию о проекте node.js

readme.md – файл, содержащий базовые инструкции по установке зависимостей

tsconfig.json – файл, содержащий конфигурации языка typescript

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3.2. Архитектура серверной части



types – папка с вспомогательными типами данных

utils – папка с утилитарными классами для работы с системой и API

index.ts – основной процесс программы, инициализирующий объекты классов и другие процессы

config.ts – обертка над использованием переменных среды

env.d.ts – типизация встроенных переменных среды для более удобной интеграции с TypeScript

mainWindow.ts – основные настройки окна и инициализация приложения

preload.ts – объявление событий коммуникации между клиентской и серверной частью приложения

renderer.d.ts – типизация событий коммуникации для более удобной интеграции с TypeScript

security.ts – дополнительные процедуры защиты данных, защиты от инъекций

types/AccessData.ts – типизация ответа от сервиса авторизации НИУ ВШЭ

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

types/index.ts – экспортирующий файл каталога для модуля

types/RefreshData.ts – типизация ответа от сервиса авторизации при использовании ключа обновления

types/Token.ts – типизация ключа доступа/обновления

types/TypeChecker.ts – функция проверки принадлежности объекта к сложному типу данных

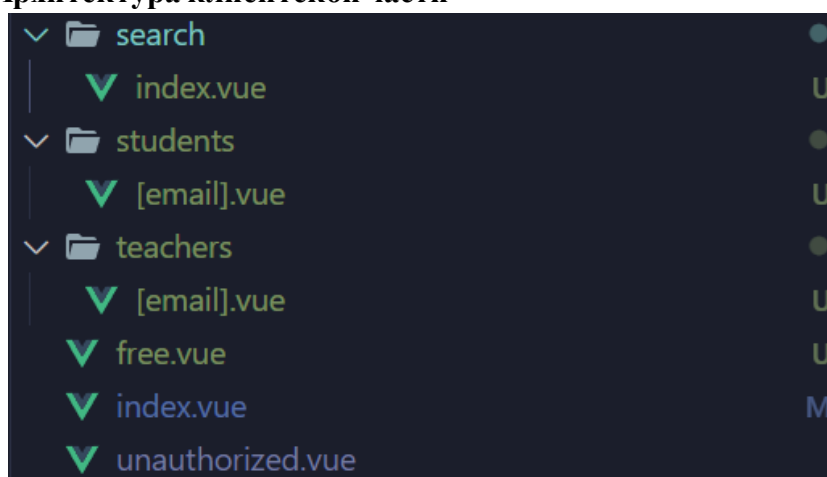
utils/HSEAPIService.ts – класс работы с API НИУ ВШЭ

utils/HSEAuthService.ts – класс для работы с сервисом авторизации НИУ ВШЭ

utils/InternalEventEmmitter.ts – класс внутренних событий

utils/Vault.ts – класс для работы с сохранением данных для доступа после начала новой сессии

3.3.3. Архитектура клиентской части



index.vue – основная страница с расписанием студента

free.vue – страница поиска свободных аудиторий

unauthorized.vue – страница авторизации

search/index.vue – страница поиска студентов, групп, аудиторий, преподавателей

students/[email].vue – страница информации о студенте

teachers/[email].vue – страница информации о сотруднике

3.4. Описание и обоснование выбора метода организации входных и выходных данных

3.4.1. Авторизация

Экран авторизации представляет собой ссылку на внешнюю страницу, отображаемую в браузере на компьютере пользователя. Таким образом, приложение обеспечивает изоляцию данных пользователя от приложения. Входными данными на внешней странице являются учетные данные единого личного кабинета. Выходными данными является обратный переход в приложение с помощью технологии deep link, а также получение доступа к приложению и его функционалу.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.5. Просмотр собственного расписания

Для просмотра собственного расписания, пользователю достаточно перейти на соответствующую страницу, в случае, когда пользователь авторизован, дополнительных входных данных не требуется. Выходными данными является структурированное представление расписания занятий текущего пользователя на ближайшие 14 календарных дней в виде перечисления списков, сгруппированных по дате проводимых занятий. Для каждого занятия должна быть отражена следующая информация: время с соответствующей ему парой и место проведения, тип (семинар, лекция и т.п.), а также лицо, проводящее занятие.

3.5.1. Поиск студентов, сотрудников, групп и аудиторий

Входными данными на странице поиска являются тип искомого объекта и поисковой запрос. Выходными данными является структурированный список сжатых данных (заголовков и описание), отвечающий запросу в соответствии с API НИУ ВШЭ. Каждая сущность студента или сотрудника из списка имеет возможность нажатия, для перехода к информации о ней.

3.5.2. Страницы информации о сущностях

3.5.2.1. Страница информации о сущности студента

Входными данными является внутренняя информация о сущности, полученная после перехода со страницы поиска. Выходными данными является отображение доступной информации о студенте, такой как: почта, курс, группа, факультет, фотография, направление, расписание на ближайшие две недели.

3.5.2.2. Страница информации о сущности сотрудника

Входными данными является внутренняя информация о сущности, полученная после перехода со страницы поиска. Выходными данными является отображение доступной информации о студенте, такой как: почта, место работы, рабочий телефон, время работы, время присутствия, день рождения, расписание на ближайшие две недели.

3.5.3. Страница поиска свободных аудиторий

Входными данными на странице поиска свободных аудиторий являются здание, дата и желаемый временной интервал. Выходными данными является структурированный ответ от API НИУ ВШЭ, соответствующий входным данным.

3.6. Описание и обоснование выбора состава технических и программных средств

Для надёжной и бесперебойной работы программы требуется персональный компьютер имеющий стабильный доступ к сети Интернет, 4 гигабайта оперативной памяти или выше, Intel Pentium 4 или выше.

Для разработки были использованы следующие технологии:

1. **Node.js:** Среда выполнения JavaScript, позволяющая запускать JS на сервере или вне браузера. Это делает возможным создание серверной части приложений, взаимодействие с базами данных и файловой системой, а также выполнение операций ввода-вывода. В контексте десктопных приложений Node.js может использоваться для доступа к системным ресурсам и интеграции с другими программами и службами.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. **TypeScript:** Язык программирования, расширяющий JavaScript типизацией и компилирующийся в JavaScript. TypeScript добавляет строгую типизацию и объектно-ориентированные возможности, что упрощает разработку крупных приложений и повышает их надежность и поддерживаемость.
3. **Vue.js:** Прогрессивный фреймворк для построения пользовательских интерфейсов. Vue.js идеально подходит для создания реактивных компонентов и одностраничных приложений (SPA), обеспечивая быструю и отзывчивую работу приложения.
4. **Nuxt.js:** Фреймворк на основе Vue.js, предназначенный для создания универсальных Vue-приложений. Он автоматизирует многие процессы, такие как маршрутизация и серверный рендеринг, делая разработку более простой и быстрой. Хотя Nuxt.js чаще используется для веб-сайтов, его можно адаптировать для десктопных приложений в комбинации с Electron.
5. **Vuetify:** Материальный компонентный фреймворк для Vue.js. Vuetify предлагает готовые компоненты в дизайне Material, что позволяет быстро создавать красивые и функциональные пользовательские интерфейсы без необходимости дизайна с нуля.
6. **Pinia:** Состояние управления для Vue.js приложений, является преемником Vuex. Pinia предоставляет простой и эффективный способ управления состоянием приложения, облегчая обмен данными между компонентами и их синхронизацию.
7. **Electron:** Фреймворк для разработки кросс-платформенных десктопных приложений с использованием веб-технологий (HTML, CSS, JavaScript). Electron позволяет упаковывать веб-приложение в десктопное приложение, работающее на различных операционных системах, используя Chromium для рендеринга интерфейса и Node.js для доступа к системным ресурсам.

Выбор таких технологий обусловлен несколькими факторами:

1. Разработчики были знакомы с частью технологий
2. Данные технологии популярны, активно развиваются и поддерживаются, что гарантирует большое количество обучающих материалов
3. Технологии хорошо оптимизированы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Программа предназначена для использования студентами и сотрудниками НИУ ВШЭ.

4.2. Экономические преимущества разработки по сравнению с аналогами

Сравнение проекта “Десктопное приложение для работы с API НИУ ВШЭ” с аналогами находится в Таблице 1.

Таблица 1

Название	Ссылка	Преимущества	Недостатки
HSE App X	https://apps.apple.com/ru/app/hse-app-x/id1527320487	Приложение представляет из себя широкий инструментарий для студентов и сотрудников НИУ ВШЭ.	Приложение распространяется только на мобильных платформах.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101–77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.103–77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.102–77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104–78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105–78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201–78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603–78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604–78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 15150–69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
11. ГОСТ 19.301–79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ/ТИПОВ

Таблица 2

КЛАСС/ТИП	НАЗНАЧЕНИЕ
type AccessData	Данные ключа доступа
type RefreshData	Данные ключа обновления
type Token	Данные ключа доступа/обновления
type PersonEntitie	Данные студента/сотрудника
type RawScheduleElement	Получаемые от API данные расписания
type LecturerProfile	Данные о преподавателе
type Location	Данные о месте проведения занятия
type ScheduleElement	Структурированные данные о расписании
type ScheduleDay	Агрегированные данные о расписании по дню
type SearchType	Перечисление типов поиска
class HSEAPIService	Класс, содержащий запросы к API
class HSEAuthService	Класс для авторизации в системе НИУ ВШЭ
class InternalEventEmmitter	Класс внутренних событий
class Vault	Класс для постоянного хранения данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ

Таблица 2

Класс HSEAPIService			
ПОЛЯ			
МЕТОДЫ			
НАЗВАНИЕ	ТИП ВОЗВРАЩАЕМОГО	АРГУМЕНТЫ	НАЗНАЧЕНИЕ
requestSearchResults	any	Query, SearchType	Отправляет поисковой запрос к серверу
getFullPersonInfo	any	email	Получает информацию о студенте/сотруднике
GetSchedule	any	email	Получает информацию о расписании занятий студента/сотрудника
Класс HSEAuthService			
ПОЛЯ			
МЕТОДЫ			
НАЗВАНИЕ	ТИП ВОЗВРАЩАЕМОГО	АРГУМЕНТЫ	НАЗНАЧЕНИЕ
getCodeFromUrl	String	url	Извлекает код из полученной ссылки
requestAccessData	Promise<AccessData RefreshData>	grant_type, additional_data	Запрашивает ключ доступа
authorizeByRedirectUrl	Void	redirectUrl	Процесс авторизации
refreshAccessData	Void	-	Обновить ключ доступа
isExpiredToken	boolean	-	Проверка истечения ключа доступа/обновления

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

getURLEncodedParams	String	grant_type, additional_data	Закодировать параметры
getStdAuthURL	String	-	Получить стандартную ссылку
openAuhtBrowserExternal	Void	-	Открыть ссылку авторизации
authorize	Promise<boolean>	-	Запуск процесса авторизации
getFullUserInfo	Promise<any>	-	Получить информацию о текущем пользователем
leave	boolean	-	Выйти из системы
isAuthorized	boolean	-	Проверка авторизации
Класс Vault			
ПОЛЯ			
НАЗВАНИЕ	ТИП	НАЗНАЧЕНИЕ	
store	Store	Статическое хранилище данных	
МЕТОДЫ			
НАЗВАНИЕ	ТИП ВОЗВРАЩАЕМОГО	АРГУМЕНТЫ	НАЗНАЧЕНИЕ
saveAccessData	String	data	Извлекает код из полученной ссылки
getToken	Token	tokenType	Извлекает ключ доступа/обновления из хранилища
resetTokens	void	-	Очищает хранилище от данных о ключах доступа/обновления

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Лист регистрации изменений

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.04-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата