

Отчет о проверке на заимствования №1



Автор: Нюхалов Денис Глебович

Проверяющий: Государев Илья Борисович (igossoudarev@herzen.spb.ru / ID: 10407)

Организация: Российский государственный педагогический университет им. А. И. Герцена

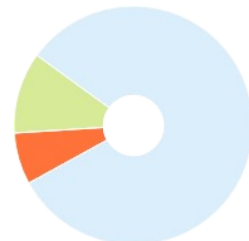
Отчет предоставлен сервисом «Антиплагиат» - <http://herzen.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 79
Начало загрузки: 24.05.2022 16:58:47
Длительность загрузки: 00:00:08
Корректировка от 24.05.2022 18:01:04
Имя исходного файла: денис.docx
Название документа: Разработка API онлайн-микросервиса по редактированию изображений на основе импортозамещенных веб-технологий
Размер текста: 1 кБ
Тип документа: Выпускная квалификационная работа
Символов в тексте: 33818
Слов в тексте: 4198
Число предложений: 252

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 24.05.2022 16:58:56
Длительность проверки: 00:01:05
Комментарии: не указано
Поиск с учетом редактирования: да
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, СПС ГАРАНТ, Модуль поиска "РГПУ им.А.И.Герцена", Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley, Переводные заимствования



ЗАИМСТВОВАНИЯ

6,9% ■

САМОЦИТИРОВАНИЯ

0% ■

ЦИТИРОВАНИЯ

11,16% ■

ОРИГИНАЛЬНОСТЬ

81,94% ■

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа. Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте	Комментарии
[01]	9,53%	9,53%	не указано	13 Янв 2022	Библиография	1	1	
[02]	0,92%	2,01%	b117637	15 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	4	3	
[03]	0%	1,91%	b 1 1 7 9 5 53атоннаяpdf	14 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	3	
[04]	0,32%	1,86%	b120529	24 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	1	3	
[05]	0%	1,68%	b117951.pdf Мелик-Карамянц Карина Сергеевна	14 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	
[06]	0%	1,68%	b121191 Лисевич Дарья Владимировна	23 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	
[07]	1,63%	1,63%	не указано	13 Янв 2022	Шаблонные фразы	14	14	
[08]	0%	1,62%	Своеобразие формирования образа «Я» у дошкольников с нарушением зрения	19 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	
[09]	0%	1,62%	b117912 Муравьева М.П.	13 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	
[10]	0%	1,47%	Своеобразие освоения рельефно-точечного шрифта Брайля слепыми младшими школьниками	19 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	
[11]	0%	1,46%	42.03.01Реклама и связи с общественностью_Реклама и связи с общественностью в индустрии моды_КошелеваТ (2).pdf	30 Мая 2019	Кольцо вузов	0	3	
[12]	0%	1,36%	b120520	23 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	
[13]	0%	1,34%	b120540	24 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2	

[14]	0%	1,1%	Особенности проектирования серверного и клиентского программного обеспечения web-сайта с использованием rest-архитектуры – тема научной статьи по компьютерным и информационным наукам читайте бесплатно текст научно-исследовательской работы в электронной ... https://cyberleninka.ru	12 Мая 2022	Интернет Плюс	0	2
[15]	0,37%	1,06%	Веб термины - краткий словарь // Вебшкола http://weblabla.ru	06 Янв 2016	Интернет Плюс	2	5
[16]	0,48%	0,88%	Примеры сервисов использующих soap. Пишем SOAP клиент-серверное приложение на PHP. Введение в XML-Schema https://halzen.ru	27 Дек 2020	Интернет Плюс	3	5
[17]	0%	0,85%	https://xn--d1abbusdciv.xn--p1ai/wp-content/uploads/2021/06/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%D0%BD%D0%B0%D1%8F-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-%D0%93%D0%BE%D1%80%D0%BB%D0%BE%D0%B2-%D0%9C.%D0%98.-20.06.21.pdf https://xn--d1abbusdciv.xn--p1ai	23 Фев 2022	Интернет Плюс	0	2
[18]	0%	0,74%	Туристско-рекреационный потенциал как основа развития въездного туризма Китая http://dep.nlb.by	16 Янв 2020	Диссертации НББ	0	2
[19]	0,72%	0,72%	Анализ SIP-сообщений на локальной сети кафедры АЭС Сибирского государственного университета телекоммуникаций и информатики. http://elibrary.ru	10 Фев 2020	Перефразирования по eLIBRARY.RU	1	1
[20]	0,38%	0,7%	Разъяснения Минцифры понятий 284 и 427 статей НК «разработка», «адаптация», «модификация», «установка», «тестирование», «сопровождение», «программа для ЭВМ», база данных, (программное средство и информационный продукт вычислительной техники)» Верное ... https://xn----dtbhaacat8bfloi8h.xn--p1ai	10 Апр 2022	Интернет Плюс	1	2
[21]	0%	0,69%	https://files.stroyinf.ru/Data2/1/4293791/4293791958.pdf https://files.stroyinf.ru	23 Мар 2020	Интернет Плюс	0	4
[22]	0%	0,69%	ГОСТ Р 54324-2011 (IEC/TS 61970-2:2004) Интерфейс прикладных программ системы управления производством и распределением электроэнергии. Часть 2. Термины и определения, ГОСТ Р от 30 мая 2011 года №54324-2011 http://docs.cntd.ru	24 Мая 2022	Интернет Плюс	0	4
[23]	0%	0,69%	ГОСТ Р 54324-2011: Интерфейс прикладных программ системы управления производством и распределением электроэнергии. Часть 2. Термины и определения https://standartgost.ru	23 Мая 2022	Интернет Плюс	0	4
[24]	0%	0,65%	https://esu.citis.ru/ikrbs/6GWYEZJRYNIRY1H1DDDKQEKX(1/2) https://esu.citis.ru	20 Мар 2018	Интернет Плюс	0	3
[25]	0%	0,64%	72179 http://e.lanbook.com	10 Мар 2016	Сводная коллекция ЭБС	0	1
[26]	0%	0,64%	https://physics.herzen.spb.ru/wp-content/docs/trebovaniya-vkr-2019.pdf https://physics.herzen.spb.ru	17 Янв 2021	Интернет Плюс	0	1
[27]	0%	0,64%	https://physics.herzen.spb.ru/wp-content/docs/trebovaniya-vkr-2020.pdf https://physics.herzen.spb.ru	23 Мая 2022	Интернет Плюс	0	1
[28]	0,62%	0,62%	Cloud Compute. Аренда облачного сервера на Windows и Linux. Р Калькулятор стоимости. https://mcs.mail.ru	24 Мая 2022	Интернет Плюс	2	2
[29]	0,61%	0,61%	СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ В РАЗРАБОТКЕ API ВЕБ-ПРИЛОЖЕНИЙ. http://elibrary.ru	09 Июл 2020	eLIBRARY.RU	3	3
[30]	0%	0,6%	Набор волонтеров на Международную математическую олимпиаду для школьников http://sanktpeterburg.bezformata.com	20 Мая 2020	СМИ России и СНГ	0	1

[31]	0,55%	0,55%	Сборник ПИС-2016.pdf https://sibsutis.ru	01 Янв 2017	ПЕРЕФРАЗИРОВАНИЯ ПО ИНТЕРНЕТУ	1	1
[32]	0%	0,52%	Приказ Министерства образования и науки Российской Федерации (Минобрнауки России) от 30 августа 2017 г. N 866 г. Москва "Об утверждении перечня олимпиад школьников и их уровней на 2017/18 учебный год"	27 Дек 2018	СМИ России и СНГ	0	2
[33]	0%	0,52%	Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript http://ibooks.ru	21 Янв 2020	Сводная коллекция ЭБС	0	1
[34]	0%	0,52%	Использование REST архитектуры для построения взаимодействия компонентов в веб-сервисе. http://elibrary.ru	11 Фев 2020	ПЕРЕФРАЗИРОВАНИЯ ПО eLIBRARY.RU	0	1
[35]	0,22%	0,49%	Об утверждении Правил электронного обмена данными в интегрированной информационной системе внешней и взаимной торговли, Решение Коллегии ЕЭК от 27 января 2015 года №5 http://docs.cntd.ru	22 Июн 2020	Интернет Плюс	1	3
[36]	0%	0,45%	Разработка системы для хранения видеоархивов http://refleader.ru	29 Мая 2021	Интернет Плюс	0	1
[37]	0%	0,45%	BUNDLED AUTHORIZATION REQUESTS - Oracle International Corporation (3/4) http://freepatentsonline.com	09 Ноя 2016	Патенты СССР, РФ, СНГ	0	1
[38]	0%	0,45%	Обработка ошибок в RESTful приложениях http://gigamir.net	23 Дек 2018	СМИ России и СНГ	0	1
[39]	0%	0,45%	Реферат - Использование методов и алгоритмов анализа программного кода и серверного архитектурного стиля REST для повышения качества С-программного продукта - Мазалов Роман Александрович https://masters.donntu.org	24 Сен 2021	Интернет Плюс	0	3
[40]	0,27%	0,45%	Разработка программы- тренажера 'Управление электросетями'. Дипломная (ВКР). Информационное обеспечение, программирование. 2008-12-09 https://bibliofond.ru	02 Ноя 2020	Интернет Плюс	2	3
[41]	0%	0,43%	Особенности проектирования серверного и клиентского программного обеспечения web-сайта с использованием rest-архитектуры https://cyberleninka.ru	02 Мая 2020	Интернет Плюс	0	2
[42]	0%	0,42%	4147-6901-1-SM.docx	14 Июл 2021	Кольцо вузов	0	1
[43]	0,2%	0,42%	Применение сервис-ориентированной архитектуры в корпоративных информационных системах. http://elibrary.ru	раньше 2011	eLIBRARY.RU	1	2
[44]	0%	0,42%	ПОДВОДНЫЕ КАМНИ SOA.	15 Дек 2018	СМИ России и СНГ	0	2
[45]	0%	0,42%	https://fgosvo.ru/uploadfiles/Projects_PO_OP/MAG/09.04.01_2.pdf https://fgosvo.ru	15 Фев 2022	Интернет Плюс	0	2
[46]	0%	0,42%	Исхаков, Алмаз Раилевич Методы математического моделирования обработки и анализа изображений в модифицированных дескриптивных алгебрах изображений : диссертация ... кандидата физико-математических наук : 05.13.18 Уфа 2017 http://dlib.rsl.ru	19 Фев 2018	Сводная коллекция РГБ	0	1
[47]	0%	0,42%	ВКР_ПЗ_Секачкин	27 Июн 2019	Кольцо вузов	0	1
[48]	0%	0,42%	Security enhancements for UDDI https://doi.org	раньше 2011	Издательство Wiley	0	3
[49]	0%	0,41%	Система методической подготовки будущего учителя химии к использованию информационно-коммуникационных технологий http://dep.nlb.by	06 Дек 2018	Диссертации НББ	0	1
[50]	0%	0,41%	Architecting the Web of Things for the fog computing era https://doi.org	раньше 2011	Издательство Wiley	0	1
[51]	0%	0,41%	REST4Mobile: A framework for enhanced usability of REST services on smartphones	10 Янв 2020	Издательство Wiley	0	1

			https://doi.org				
[52]	0%	0,41%	Creating a gateway that enables large-scale science cooperation https://doi.org	28 Фев 2015	Издательство Wiley	0	1
[53]	0%	0,41%	Definition of REST web services with JSON schema https://doi.org	30 Июн 2017	Издательство Wiley	0	1
[54]	0%	0,39%	RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content https://datatracker.ietf.org	24 Мая 2022	Интернет Плюс	0	1
[55]	0%	0,39%	Ишкина, Евгения Геннадиевна диссертация ... кандидата технических наук : 05.13.10 Б. м. Б. г. http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	0	1
[56]	0,35%	0,35%	Информатика 2015 https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	1	1
[57]	0%	0,35%	Особенности проектирования серверного и клиентского программного обеспечения web-сайта с использованием rest-архитектуры – тема научной статьи по компьютерным и информационным наукам читайте бесплатно текст научно-исследовательской работы в электронной ... https://cyberleninka.ru	01 Июн 2021	Интернет Плюс	0	1
[58]	0%	0,35%	Информатика 2015 http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[59]	0%	0,34%	Глушакoвa_ВКР v1.docx	07 Дек 2016	Кольцо вузов	0	1
[60]	0%	0,34%	Разработка сайта с использованием технологий Node.js, Express, MongoDB и Pug	05 Июл 2020	Кольцо вузов	0	1
[61]	0%	0,33%	Решение Арбитражного суда г.Санкт-Петербурга и Ленинградской области от 23 июня 2011 г. по делу N А56-7756/2011 (ключевые темы: программное обеспечение - государственный оборонный заказ - разработка - предписание - расходование денежных средств) http://arbitr.garant.ru	29 Дек 2016	СПС ГАРАНТ	0	1
[62]	0%	0,33%	Англо-русский словарь сокращений по компьютерным технологиям, информатике, электронике и связи http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	0	1
[63]	0%	0,33%	Информатика. Новый систематизированный толковый словарь-справочник https://e.lanbook.com	20 Янв 2020	Сводная коллекция ЭБС	0	1
[64]	0%	0,33%	2365 http://e.lanbook.com	09 Мар 2016	Сводная коллекция ЭБС	0	1
[65]	0%	0,33%	Информатика. Новый систематизированный толковый словарь-справочник http://biblioclub.ru	20 Апр 2016	Сводная коллекция ЭБС	0	1
[66]	0%	0,33%	Программирование на языке C в Microsoft Visual Studio 2010 Контент-платформа Pandia.ru https://pandia.ru	24 Мая 2022	Интернет Плюс	0	1
[67]	0%	0,33%	https://www.miigaik.ru/upload/iblock/ca4/ca46c33076ee355d2a0efa80d6105268.pdf https://miigaik.ru	24 Мая 2022	Интернет Плюс	0	1
[68]	0%	0,33%	Информатика http://studentlibrary.ru	20 Янв 2020	Медицина	0	1
[69]	0%	0,33%	Информатика http://studentlibrary.ru	20 Янв 2020	Медицина	0	1
[70]	0%	0,33%	Медицинская информатика http://studentlibrary.ru	26 Янв 2018	Медицина	0	1
[71]	0%	0,33%	Информатика для медицинских колледжей: учебное пособие http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[72]	0%	0,33%	Основы информационных и телекоммуникационных технологий. Программные средства информационных технологий http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[73]	0%	0,33%	Основы информационных и телекоммуникационных технологий. Введение в компьютерную графику http://studentlibrary.ru	19 Дек 2016	Медицина	0	1

[74]	0%	0,32%	Перечень ТР ТС 001/2011 http://gostinfo.ru	24 Мая 2022	Интернет Плюс	0	1
[75]	0,32%	0,32%	[Перевод] CORS, CSP, HTTPS, HSTS: о технологиях веб-безопасности http://pcnews.ru	21 Дек 2018	СМИ России и СНГ	1	1
[76]	0%	0,31%	https://dspace.tltsu.ru/bitstream/123456789/864/1/%D0%A1%D0%B0%D0%B2%D0%BA%D0%B8%D0%BD%20%D0%94.%D0%95_%D0%9C%D0%9E%D0%91_1201.pdf https://dspace.tltsu.ru	12 Апр 2022	Интернет Плюс	0	1
[77]	0%	0,31%	Хэл Фултон совместно с Андрэ Арко Путь Ruby пер. с англ. Москва 2016 http://dlib.rsl.ru	05 Авг 2019	Сводная коллекция РГБ	0	1
[78]	0%	0,29%	Информатика. Энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[79]	0%	0,29%	ECMAScript 6 для разработчиков http://ibooks.ru	21 Янв 2020	Сводная коллекция ЭБС	0	1
[80]	0%	0,29%	Пользователь нашел простой способ просмотра видеороликов на YouTube без рекламы, а также онлайн-статей без подписки / Хабр https://habr.com	30 Мая 2021	Интернет Плюс	0	1
[81]	0%	0,29%	ПРИКЛАДНЫЕ ПРИНЦИПЫ РЕАЛИЗАЦИИ И УДАЛЕННОГО ДОСТУПА К ВОПРОСНО-ОТВЕТНОЙ СИСТЕМЕ. http://elibrary.ru	20 Мая 2020	eLIBRARY.RU	0	1
[82]	0%	0,29%	ДР_ДементейВЮ_1_22.05.2021.docx	22 Мая 2021	Кольцо вузов	0	1
[83]	0,07%	0,28%	Виды хостинга для сайтов организаций http://metallcheckiy-portal.ru	24 Мая 2022	Интернет Плюс	1	2
[84]	0%	0,27%	MySQL запустить&nb p;— тоже надо уметь - Интернет » Интересное в сети http://rusadvice.org	24 Мая 2022	Интернет Плюс	0	2
[85]	0%	0,27%	VDS/VPS-сервер в аренду от 1cloud. Аренда выделенного виртуального сервера за 2 мин https://1cloud.ru	24 Мая 2022	Интернет Плюс	0	2
[86]	0%	0,26%	Об утверждении Правил электронного обмена данными в интегрированной информационной системе внешней и взаимной торговли - ИПС "Эділет" http://adilet.zan.kz	04 Окт 2017	ИПС Адилет	0	1
[87]	0,06%	0,26%	Система работы над профессионально ориентированной лексикой с курсантами военного вуза на занятиях по русскому языку как иностранному. http://elibrary.ru	08 Июл 2020	eLIBRARY.RU	1	1
[88]	0%	0,26%	Платформа Windows Azure http://studentlibrary.ru	19 Дек 2016	Медицина	0	1
[89]	0%	0,25%	http://sp.kaznpu.kz/docs/jurnal_file/file20210510101342.pdf http://sp.kaznpu.kz	18 Мая 2022	Интернет Плюс	0	1
[90]	0%	0,24%	Недолужко, Илья Валерьевич диссертация ... кандидата технических наук : 05.13.11 Владивосток 2014 http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	0	1
[91]	0%	0,22%	Создание распределенной системы подготовки ремонтно-строительных работ на инженерных коммуникациях с использованием информационных технологий. http://elibrary.ru	29 Авг 2007	eLIBRARY.RU	0	1
[92]	0,21%	0,21%	Проектирование организации(Анализ среды проектируемой организации) [Курсовая №64795] https://evkova.org	10 Дек 2021	Интернет Плюс	1	1
[93]	0%	0,21%	Обзор VPS и технологий виртуализации http://limej.ru	24 Мая 2022	Интернет Плюс	0	1
[94]	0,21%	0,21%	Проектирование удаленной лаборатории с целью проведения лабораторных опытов для школ. http://elibrary.ru	23 Сен 2020	eLIBRARY.RU	1	1
[95]	0%	0,2%	Технология сборки интеллектуальных и информационных ресурсов Интернет.	12 Фев 2019	eLIBRARY.RU	0	1

<http://elibrary.ru>

[96]	0%	0,2%	ОБОСНОВАНИЕ ИСПОЛЬЗОВАНИЯ ПРОТОКОЛА SOAP ПРИ ВЗАИМОДЕЙСТВИИ ПЛАТЁЖНОЙ СИСТЕМЫ С ГИС ГМП. МЕТОД АНАЛИЗА ИЕРАРХИЙ. http://elibrary.ru	03 Мая 2017	eLIBRARY.RU	0	1	
[97]	0%	0,2%	Пишем SOAP клиент-серверное приложение на PHP http://habrahabr.ru	27 Дек 2018	СМИ России и СНГ	0	1	
[98]	0%	0,19%	КСП 5cem Flashcards Quizlet https://quizlet.com	04 Апр 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[99]	0%	0,19%	Виртуальный выделенный сервер (VDS - Virtual Dedicated Server) https://dataekb.ru	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[100]	0%	0,18%	https://www.sbmt.bsu.by/upload/2014%20%D0%9A%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F%20%D0%90%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D1%8B%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D... https://sbmt.bsu.by	15 Мар 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[101]	0%	0,16%	Веб-хостинг в эпицентре трансформации КомпьютерПресс https://compress.ru	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[102]	0%	0,16%	Веб-хостинг в эпицентре трансформации КомпьютерПресс http://compress.ru	раньше 2011	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[103]	0%	0,14%	VPS хостинг на базе платформы Hyper-V от 650 рублей в месяц masterhost https://masterhost.ru	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



7

Направление подготовки/специальность
09.03.01 Информатика и вычислительная техника

направленность (профиль)/специализация
«Информатика и вычислительная техника»

Выпускная квалификационная работа

2

Разработка API онлайн-микросервиса по редактированию изображений на
основе импортозамещенных веб-технологий

Обучающегося 4 курса
очной формы обучения
Нюхалова Дениса Глебовича

2

Руководитель выпускной квалификационной
работы:
кандидат педагогических наук, доцент
Государев Илья Борисович

4

Санкт-Петербург
2022

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ. ПОДГОТОВКА К РАЗРАБОТКЕ	6
1.1. Анализ современных подходов к написанию онлайн сервисов и API для обработки изображений	6
1.2. Анализ отечественных сервисов аренды виртуальных выделенных серверов	13
ГЛАВА 2. РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ГРАФИЧЕСКОГО РЕДАКТОРА И API ПО ОБРАБОТКЕ ИЗОБРАЖЕНИЙ	15
2.1 Определение требований к API	16
2.2 Проектирование API	17
2.3 Кодирование API	18
2.4 Интеграция API	25
ЗАКЛЮЧЕНИЕ	30
СПИСОК ЛИТЕРАТУРЫ	31
ПРИЛОЖЕНИЕ А	34
ПРИЛОЖЕНИЕ Б	36

ВВЕДЕНИЕ

Многие современные web-приложения спроектированы по технологии «клиент-сервер». Клиентская часть приложения работает в браузере. Она предоставляет интерфейс, при помощи которого пользователь может взаимодействовать с программным продуктом. Серверная часть приложения выполняется на удаленном компьютере или системе компьютеров. По запросу с клиентской части, она выполняет программный код и отправляет результаты своей работы обратно пользователю. Принципы взаимодействия клиентской и серверной частей определяет API (Application Programming Interface – «программный интерфейс приложения»). API – это набор функций серверной части приложения, к которому может обращаться клиентская часть того же приложения или иные программы.

Разработчики и IT компании могут на платной или безвозмездной основе предоставлять доступ к API своих приложений. Это открывает программистам широкий инструментарий для решения разного рода задач уже в своих программных продуктах. Одной из таких задач является редактирование и обработка изображений.

Различного рода изображения повсеместно распространены в интернете. На коммерческих сайтах, в приложениях и социальных сетях, ежедневно размещается огромное количество фотографий, рисунков, схем и т.д. Люди публикуют их, как для развлечения, так и для реализации своих практических целей. Для того, чтобы пользователи и разработчики могли корректно визуализировать ту или иную информацию при помощи изображений, им, зачастую, необходима обработка или редактирование. Это позволяет улучшить зрительное восприятие изображения, удалить или добавить различные графические элементы, подготовить к печати.

Существует большое количество онлайн-сервисов для обработки изображений. Именно они, чаще всего устроены по технологии «клиент-сервер». Обработка изображений – требовательный к ресурсам компьютера процесс, поэтому, большинство функций онлайн-редакторов чаще всего выполняются на более мощных удаленных серверах.

Существует очень ограниченное количество функционирующих российских приложений и API обработки изображений. Этим обуславливается **актуальность** дипломной работы.

Объект исследования – API, предназначенные для редактирования изображений. **Предметом исследования** является серверная часть сервиса по обработке изображения.

Целью работы является разработка API для онлайн-микросервиса по редактированию изображений. Этот сервис следует создавать с использованием импортозамещенных технологий и размещать на российских серверах

Comment [1]: исправил время

В соответствии с поставленной целью необходимо решить следующие задачи:

- Проанализировать современные подходы к написанию онлайн сервисов и API для обработки изображений;
- Изучить отечественные сервисы, предоставляющие услуги аренды виртуальных выделенных серверов;
- Определить набор функций, которые будут выполняться на серверной части сервиса;
- Разработать серверную часть приложения и API, реализующие функции по редактированию изображений;
- Разместить код серверной части приложения на виртуальном выделенном сервере.

Теоретическая значимость заключается в определении современных подходов и принципов разработки API в сфере работы с изображениями.

Практическая значимость заключается в разработке API онлайн-микросервиса редактирования изображений для отечественных приложений.

Результатом выпускной квалификационной работы является готовый к использованию онлайн-микросервис с API, работающий на российском сервере.

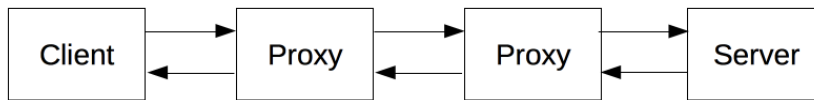
Дипломная работа имеет следующую **структуру**: введение, две главы, заключение, список литературы. Выпускная квалификационная работа содержит - страниц. В работе представлено - рисунков, - таблиц.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ. ПОДГОТОВКА К РАЗРАБОТКЕ

1. 1.1. Анализ современных подходов к написанию онлайн сервисов и API для обработки изображений

API взаимодействует с приложениями посредством обмена сообщениями. Структуру сообщений определяют протоколы передачи данных. Одним из основных таких протоколов в интернете является HTTP (HyperText Transfer Protocol — «протокол передачи гипертекста») и его расширение HTTPS с поддержкой шифрования. Он используется во многих широко известных web-API[19].

HTTP – это протокол, обеспечивающий общение между клиентскими и серверными частями web-приложений. Сообщения, отправляющиеся с клиента, называются запросами, а с сервера – ответами. Клиентом можно назвать любой инструмент или устройство, действующее от лица пользователя. Чаще всего, в роли клиента выступает веб-браузер. Сервером является одна или несколько виртуальных машин или компьютеров. Он полностью или частично обрабатывает данные, полученные с запросом, и предоставляет результат обработки в ответе клиенту. Между ними могут находиться программы-посредники, так называемые прокси, которые не влияют на запрос, но способны выполнять различные вспомогательные функции, которые по тем или иным причинам не может выполнить сервер, как например, кэширование, протоколирование, фильтрация и т.д. Схематическая последовательность обмена сообщениями между клиентом и сервером в рамках протокола HTTP представлена на рисунке 1 [1].



16

Рисунок 1 – Обмен сообщениями между клиентом и сервером, автор – MDN Web Docs

Любое сообщение в HTTP состоит из «головы» и «тела». Головная часть запроса включает в себя:

1. Стартовую строку. В ней содержится версия протокола и метод HTTP, который описывает требуемое от сервера действие. В HTTP представлено несколько методов, каждый из них может описывать то или иное действие на сервере. Методы могут отличаться на предмет ограничений по типу данных, длине запроса, видимости и т.д. Чаще всего в запросах используются методы GET и POST [2].
2. Набор заголовков. Он уточняет запрос и передает серверу дополнительную информацию. Существует несколько типов заголовков:
 - основные заголовки, содержащие информацию о самом сообщении, например дату и время запроса;
 - заголовки запроса, которые могут передавать информацию о контексте запроса, для того чтобы сервер мог адаптировать ответ;
 - заголовки сущности, передающие данные о теле запроса.

Голова запроса содержит:

1. Строку статуса. В этой строке указана версия протокола, код состояния и текст пояснения, привязанный к коду состояния. Код состояния состоит из трех цифр и указывает на результат обработки запроса [1]. Существует пять классов кодов состояния:

- информационные, в диапазоне от 100 до 199;
- успешные, в диапазоне от 200 до 299;
- перенаправления, в диапазоне от 300 до 399;
- клиентские ошибки, в диапазоне от 400 до 499;
- серверные ошибки, в диапазоне от 500 до 599;

2. Набор заголовков. Он также как и в случае с набором заголовка запроса передает дополнительную информацию, но уже об ответе.

Также существует несколько типов таких заголовков:

- основные, относящиеся также к информации о сообщении
- заголовки ответа, передающие информацию об ответе или сервере
- заголовки сущности, передающие данные о теле ответа

Тело HTTP сообщения – необязательная, но не менее важная часть запросов и ответов. Она отделяется от головы запроса пустой строкой и содержит данные, которые мы хотим передать в запросе. Тип данных, передаваемых в теле запроса и возвращаемых в ответе, определяется заголовком Content-Type. В этом заголовке указывается MIME-тип – идентификатор формата файлов, передаваемых в интернете[20]. Пример HTTP сообщений представлен на рисунке 2 [1].

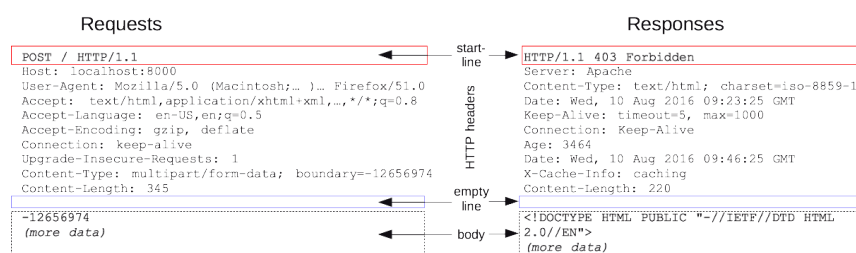


Рисунок 2 – Пример HTTP запроса и ответа, автор – MDN Web Docs

В сервисах для обработки изображений, клиент и сервер должны обмениваться HTTP сообщениями, содержащими непосредственно изображения и указания на то, как именно сервер должен его обработать.

Существует два основных подхода при обмене сообщениями в HTTP. Первый – использовать строгий стандарт написания запросов и ответов поверх HTTP – SOAP (Simple Object Access Protocol – «простой протокол доступа к объектам»). В данном случае в теле HTTP запроса и ответе сервера передается XML разметка, написанная в соответствии с языком описания WSDL (Web Services Description Language – «язык описания веб-сервисов»). Спецификации для каждого типа взаимодействия в SOAP запросах обеспечивает высокий уровень надежности и безопасности при обмене сообщениями. Второй подход – это прибегнуть к использованию архитектурного стиля REST (Representational State Transfer — «передача репрезентативного состояния») при разработке API. REST не стандартизирует тело HTTP запроса, а подразумевает под собой набор правил, позволяющих наиболее эффективно использовать возможности HTTP при обмене сообщениями [6]. Отсутствие необходимости в написании запросов определенного типа и формата обеспечивает REST относительно высокую скорость отправки запросов, а также упрощает написание RESTful приложений [5].

SOAP был одним из первых стандартов создания программируемых интерфейсов для обмена данными между системами. SOAP долгое время являлся наиболее распространенным подходом при написании API. Он остается важным стандартом для веб-сервисов и работает во многих внутренних системах по всему миру, однако для новых проектов многие организации выбирают архитектуру микросервисов с использованием REST API. REST обеспечивает более гибкий и быстрый процесс разработки приложений.

В случае клиент-серверного обмена изображениями, важную роль играет размер запросов и ответов, так как он влияет на скорость обработки HTTP

сообщений как с клиентской стороны приложения, так и с сервера. Из-за простоты применения и высокой скорости обработки запросов и ответов, большинство современных онлайн сервисов по обработке изображений построены в стиле REST. В данной дипломной работе описан процесс разработки API в соответствии с архитектурным стилем REST.

Важной частью любого онлайн редактора изображений является обработка графики. Серверная часть приложения может как выполнять узкий набор функций по обработке графики, так и проводить широкий список различных манипуляций над ней. Возможности API, зачастую, упираются в доступную для сервера мощность. Для разработки сервиса редактирования изображений было бы целесообразно использовать один многофункциональный API, так как запросы клиентской части приложения не нужно форматировать под разные серверы, что значительно упростило бы разработку и отладку.

Большинство многофункциональных API для обработки изображений не распространяются в открытом доступе. Компании, разрабатывающие онлайн-редакторы изображений не заинтересованы в том чтобы компоненты их программных продуктов использовались в сторонних проектах. Однако, некоторые из них предоставляют доступ к своим API на коммерческой основе. Одним из примеров является сервис Pixo. В его возможности входит:

- изменение формата изображения;
- применение фильтров;
- добавление текста;
- изменение разрешения изображений;
- рисование фигур поверх изображений;
- изменение параметров яркости, контрастности, насыщенности, размытия;
- редактирование гаммы изображения;
- удаление шумов;
- редактирование заднего фона изображений [9].

API также предоставляет компания Pixelixе. Среди его возможностей можно выделить:

- сжатие изображений;
- изменение размера изображений;
- изменение разрешения изображений;
- поворот изображения на определенный угол;
- отражение изображений по вертикали и горизонтали;
- изменение параметров яркости, контрастности, насыщенности, размытия;
- применение фильтров[10].

Иные взаимодействия с изображениями, как например, рисование при помощи инструмента кисть, накладывать изображения друг на друга, выделение частей изображения и т.д. осуществляются на клиентской части онлайн-редакторов. Данные манипуляции требуют хранения на сервере состояний клиента, а это противоречит стилю архитектуры REST [6].

В рамках дипломной работы будут реализованы следующие функции:

- применение фильтров;
- поворот изображений на определенный угол;
- изменение размера изображений;
- изменение разрешения изображений;
- отражение изображений по вертикали и горизонтали;
- изменение параметров размытия;

Остальные манипуляции с изображениями будут проводиться на клиентской стороне приложения.

В результате анализа инструментов и технологий адекватных поставленной задаче была выбрана платформа Node.js и библиотека Express[14]. Среди преимуществ данной платформы выделяют:

- доступность;
- масштабируемость;
- скорость выполнения исходного кода;
- встроенный пакетный менеджер NPM [4].

Для обработки изображений использовался модуль Sharp. На рисунке 3 представлен результат сравнения скорости выполнения задачи по изменению расширения JPEG изображения со значения 2725x2225 до 720x588 и его последующим сжатием с настройкой «качества» 80 разными Node.js модулями [11].

Module	Input	Output	Ops/sec	Speed-up
jpeg	buffer	buffer	0.84	1.0
squoosh-cli	file	file	1.08	1.3
squoosh-lib	buffer	buffer	1.85	2.2
mapnik	buffer	buffer	3.45	4.1
gm	buffer	buffer	8.60	10.2
gm	file	file	8.66	10.3
imagemagick	file	file	8.79	10.5
sharp	stream	stream	28.90	34.4
sharp	file	file	30.08	35.8
sharp	buffer	buffer	30.42	36.2

Рисунок 3 Сравнение скорости работы различных Node.js модулей обработки изображений

Из результатов тестирования **можно сделать вывод о том, что Sharp** предоставляет наиболее эффективный функционал для обработки изображений.

2. 1.2. Анализ отечественных сервисов аренды виртуальных выделенных серверов

Для размещения web приложений, разработчики могут воспользоваться услугами компании провайдера, которая предоставляет ресурсы своих вычислительных машин для работы. Компании могут предоставить доступ как к виртуальному хостингу – дисковому пространству на своей вычислительной машине, так и к виртуальному выделенному серверу.

VDS (Virtual Dedicated Server – виртуальный выделенный сервер) имеет ряд преимуществ перед виртуальным хостингом. На виртуальном хостинге присутствует ряд ограничений. Пользователь не может устанавливать необходимое ПО, ему доступны только те инструменты, которые предоставляет компания-поставщик услуг. VDS же предоставляет полный доступ к виртуальной операционной системе с возможностью устанавливать любое ПО. На виртуальном хостинге также имеется ограничение на использование ресурсов сервера, а для VDS можно настроить необходимую под определенное приложение аппаратную конфигурацию системы [18].

VDS в выпускной квалификационной работе позволит оптимизировать ресурсы сервера под обработку изображений, а также предоставит свободу в выборе программных средств обработки.

Был проведен сравнительный анализ следующих популярных отечественных сервисов, предоставляющих аренду VPS:

- Yandex Cloud
- Reg.ru
- VK Cloud Solutions

Результат анализа представлен в таблице 1 Приложения А.

В связи с наличием различных зон доступности, наибольшего выбора ОС, хранением данных в зашифрованном виде в соответствии с требованиями 152-ФЗ и более комфортной цены по сравнению с VK Cloud Solutions, для размещения API и серверной части приложения был выбран сервис Yandex Cloud.

Выводы по главе 1:

1. Был проведен анализ различных подходов к написанию API. Принято решение об использовании архитектурного ⁷стиля REST при разработке.
2. В соответствии с поставленной задачей был определен набор функций, которые будут выполняться на серверной части онлайн-сервиса.
3. Были изучены отечественные сервисы, предоставляющие услуги аренды виртуальных выделенных серверов и сделан вывод о том, какой сервис будет использоваться в дипломной работе.

ГЛАВА 2. РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ГРАФИЧЕСКОГО РЕДАКТОРА И API ПО ОБРАБОТКЕ ИЗОБРАЖЕНИЙ

Процесс разработки приложения проводился в соответствии с требованиями к разработке и документированию из ГОСТ Р 51904-2002 [7]. Для программного продукта, описываемого в выпускной квалификационной работе были выделены следующие этапы разработки:

- определение требований к ПО;
- проектирование ПО;
- кодирование ПО;
- интеграция ПО.

20

3. 2.1 Определение требований к API

В рамках разработки приложения было составлено техническое задание. В соответствии с ним и выводами, сделанными в первой главе, был сформирован следующий список требований к API разрабатываемого онлайн-сервиса обработки изображений:

1. Приложение должно строиться по клиент-серверной архитектуре. Разработка серверной части приложения будет вестись в архитектурном стиле REST.
2. Взаимодействие двух частей приложения будет вестись по протоколу HTTP.
3. Клиентская часть обращается к серверу только по одному URL-адресу и только по одному протоколу – HTTP.
4. В соответствии с архитектурным стилем REST, запрос к серверу должен содержать всю необходимую информацию для обработки изображения.

5. Запросы будут обрабатываться библиотекой Express платформы Node.js.
6. Средствами платформы Node.js, на серверной части должны производиться следующие операции над изображениями:
 - применение фильтров;
 - поворот изображений на определенный угол;
 - изменение размера изображений;
 - изменение разрешения изображений;
 - отражение изображений по вертикали и горизонтали;
 - изменение параметров размытия изображения;

4. 2.2 Проектирование API

На этапе проектирования была составлена Use-case диаграмма приложения. Диаграмма представлена на рисунке 1 приложения Б. По данной диаграмме можно сделать следующие выводы:

1. Определены функции приложения, при обращении к которым будут отправляться запросы к API.
2. Для изменения разрешения и размера изображения требуется обращения к функции определения метаданных.
3. Функции по размытию изображения, отражению изображения и применению фильтров не включают в себя вызов иных функций.

Также была составлена блок-схема, которая отражает принципы работы серверной части приложения. Она представлена на рисунке 2 приложения Б.

На этапе проектирования, было принято решение о том, что изображения будут передаваться в формате base64. Это формат кодирования двоичных данных при помощи символов ASCII (American standard code for information interchange – Американский стандартный код для обмена информацией) [15]. Ответы с сервера будут приходить клиенту в формате JSON. Разработка

клиентской части ведется при помощи языка программирования JavaScript. У этого языка присутствуют инструменты для быстрой обработки данных в этом формате [16]. Эти решения отражены на блок-схеме. В соответствии с ней и use-case диаграммой будет проходить процесс кодирования.

5. 2.3 Кодирование API

Кодирование API осуществлялось в редакторе исходного кода Visual Studio Code. Среди его функциональных особенностей, использованных в процессе разработки можно выделить:

- интеграцию с системой контроля версий Git;
- возможность подключения к удаленному серверу по протоколу SSH
- пользовательские расширения.

Вспомогательным инструментом при разработке являлся сервис GitHub Desktop. Это графический интерфейс для управления репозиториями веб-сервиса GitHub. В данной дипломной работе GitHub также использовался для размещения и резервного копирования исходного кода клиентской и серверной частей приложения, а также для поиска дополнительных пакетов Node.js для использования в разработке.

Структура серверной части приложения, созданного в Visual Studio Code представлена на рисунке 4.

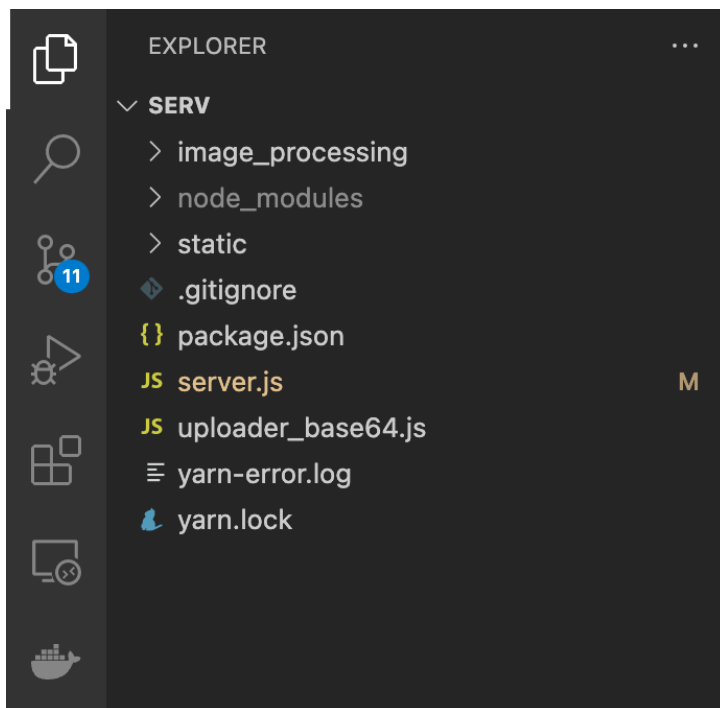


Рисунок 4 – Структура серверной части приложения

В `node_modules` располагаются пакеты, которые использовались в приложении и при разработке. `Yarn-error` и `yarn.lock` являются файлами пакетного менеджера `yarn`. То, от каких пакетов зависит работа программы описывается в файле `package.json`. В данном проекте используются:

6. Express – фреймворк для создания веб-приложений и API.
7. Body-parser – модуль Express, необходимый для извлечения тела из HTTP запроса.
8. Cors – модуль Express, обеспечивающий поддержку технологии CORS (Cross-origin resource sharing – совместное использование ресурсов между разными источниками).
9. Sharp – модуль Node.js, необходимый для обработки изображений

10. Uuid – модуль для создания строки по стандарту **UUID (Universally unique identifier - универсальный уникальный идентификатор)**.

11. Is-base64 – пакет, содержащий функции верификации base64 строк.

12. Morgan – middleware, необходимый для логирования HTTP запросов.

13. Nodemon – утилита необходимая для автоматического перезапуска процесса.

Программа для обработки изображений располагается в директории `image_processing`. Сама программа состоит из двух файлов:

- `image_functions` – включает в себя функции, обозначенные на этапе определения требований к API. Они созданы при помощи пакета Sharp.
- `image_processing_app` – в нем представлен интерфейс для взаимодействия с `image_functions`, а также вспомогательные функции для работы с форматами ввода-вывода данных.

Функции редактирования изображений написаны в соответствии с use-case диаграммой. Как видно на рисунке 5, на строчках 32 и 33, функция изменения разрешения включает в себя вызов функции определения метаданных. Похожим образом устроена функция обрезки изображения.

```
28 async function resizeImage(buf, params) {
29   try {
30     return await sharp(img_buf)
31       .resize({
32         width: Math.round((await get_meta(img_buf)).width * params[i]),
33         height: Math.round((await get_meta(img_buf)).height * params[i])
34       })
35       // .toFormat("jpeg", { mozjpeg: true })
36       .toBuffer()
37       .then(data => data)
38     } catch (error) {
39       console.log(error);
40     }
41   }
```

Рисунок 5 – Функция изменения разрешения изображения

Стоит отметить, что функции для обработки изображений являются асинхронными и возвращают объект Promise. В связи с этим некоторые функции интерфейса взаимодействия с изображениями и обработчики запросов являются асинхронными. Для работы с Promise был использован современный специальный синтаксис языка JavaScript `async/await`[12].

Интерфейс взаимодействия является важной частью программы, так как `sharp` не позволяет работать с изображениями в формате `base64`. Функции интерфейса ответственны за перевод изображений из `base64` в доступный для обработки тип данных языка JavaScript – `Buffer`. `image_processing_app` также помогает подготовить обработанные изображения к отправке.

В файле `server` Находится обработчик HTTP запросов, разработанный при помощи `Express`. В нем указаны конфигурационные данные сервера: информация о порте, на котором он работает, вызов пакетов `cors` и `morgan`. Обработка запросов к API осуществляется отдельно и расположена в файле `uploader_base64`. В соответствии с установленными ранее требованиями, на рисунке 6, на 18 строке, API вынесен на отдельный, единый для всех его запросов, URL.

```
8 // App initiation
9 const app = express()
10 const port = 3030;
11
12 // Import form file uploader route
13 app.use(morgan('dev'));
14 app.use(cors({
15   origin: '*',
16   methods: ['GET', 'POST', 'DELETE', 'UPDATE', 'PUT', 'PATCH']
17 }));
18 app.use('/api', uploaded_base);
```

Рисунок 6 – Подключение вспомогательных пакетов и определение URL для API

В `uploader_base64` описаны URL, по которым клиент может обращаться к API:

1. `/api/` – получение метаданных
2. `/api/rotate/` – поворот изображения
3. `/api/resize` – изменение расширения
4. `/api/flip` – отражение по вертикальной оси
5. `/api/flop` – отражение по горизонтальной оси
6. `/api/crop` – изменение размера
7. `/api/gblur` – изменение параметра размытия
8. `/api/filter/` – применения фильтра, например `/api/filter/greyscale` применяет черно-белый фильтр

Ответ сервера может отличаться в зависимости от HTTP метода. При обращении по URL методом GET, клиенту отсылается JSON с отредактированным изображением. При отправке POST запроса на тот же URL, в соответствии с принципом кэширования REST [6], отредактированное изображение сохраняется на сервере и в ответе отправляется JSON содержащий, помимо самого изображения, путь до него на сервере. Также, при обращении к API по `/api/` методом DELETE, изображение с сервера удаляется.

Пример обработчика GET запроса представлен на рисунке 7.

```

65 router.get('/rotate', async (req, res) => {
66   if (!req.body.path) {
67     try {
68       imageProcessor.verifyBuffer(req.body.base64Image)
69     }
70     catch(er) {
71       res.status(406).send(er)
72     }
73     const img = imageProcessor.verifyBuffer(req.body.base64Image)
74     const newImg = await imageProcessor.changeImage(img, 'rotateImage', [Number(req.body.angle)])
75     const responseResult = await imageProcessor.getImgWithMeta(newImg, req.body.image_name)
76     res.status(200).json(responseResult)
77   }
78   else {
79     try {
80       imageProcessor.bufFromFile(req.body.path)
81     }
82     catch(er) {
83       res.status(400).send(er)
84     }
85     const img = imageProcessor.bufFromFile(req.body.path)
86     const newImg = await imageProcessor.changeImage(img, 'rotateImage', [Number(req.body.angle)])
87     const responseResult = imageProcessor.SaveImage(newImg, path.join(__dirname, './static/uploads'))
88     res.status(200).json(responseResult)
89   }
90 }
91 });

```

Рисунок 7 – обработчик GET запроса

Первоначально, при помощи пакета `body-parser` программа пытается извлечь путь к изображению из тела запроса. Если в запросе его не было, извлекается `base64` строка. Она передается функции интерфейса программы обработки изображения для верификации. Затем, через тот же интерфейс мы переводим строку в тип `Buffer`, изменяем изображение и формируем объект, который будет передаваться в ответе. Если все прошло успешно, он форматируется в `JSON` и отправляется клиенту. В случае, если картинка уже была загружена ранее, обработчик выполняет все тот же набор действий, однако `Buffer` будет формироваться уже на основе изображения, сохраненного на сервере. На строчках 71 и 83 также видно, что в случае какой-то ошибки, клиенту отправляется код состояния `HTTP` и дополнительная информация. Похожим образом устроены обработчики всех остальных `GET` запросов. При `POST` запросе изображение сохраняется в директории `/static/uploads`.

Для тестирования `API` использовалось приложение `Postman`. `Postman` предоставляет возможности для:

- составления и отправки `HTTP`-запросов к `API`;

- создания набора последовательных запросов;
- структурированного хранения запросов;
- изменения заголовков и тела запросов;
- создания представления запроса в различных языках программирования и сторонних программах [8].

Исключительной особенностью Postman является интуитивно понятный интерфейс. Пример HTTP запроса в Postman представлен на рисунке 8

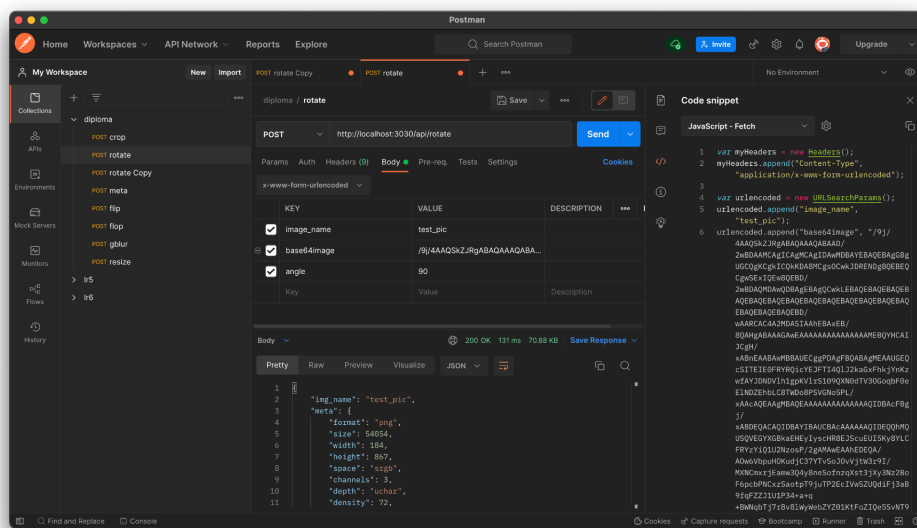


Рисунок 8 Пример HTTP запроса и ответа в Postman

В Postman были подготовлены и протестированы все возможные запросы к API. Тестирование также проводилось с клиентской части приложения.

14. 2.4 Интеграция API

API был размещен средствами платформы Яндекс Облако. Для создания VDS, была приобретена виртуальная машина на ОС Ubuntu. Управление виртуальными машинами происходило в консоли Яндекс Облака. Через нее

была составлена конфигурация машины. На Рисунке 9 представлен интерфейс изменения конфигурации виртуальной машины, на которой расположена серверная часть онлайн-редактора [13].

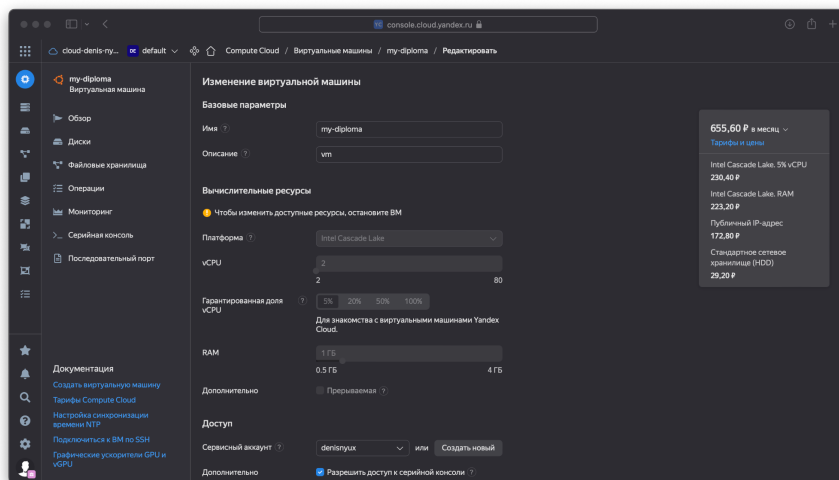


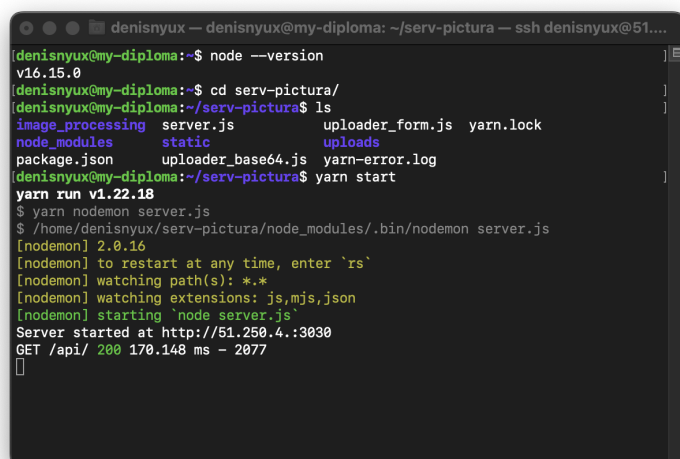
Рисунок 9 – Консоль Яндекс Облака

Для приложения, разрабатываемого в рамках данной дипломной работы, на VDS были выбраны следующие аппаратные характеристики:

- Процессор: Intel Cascade Lake
- Гарантированная доля vCPU: 5%
- Количество vCPU: 2
- RAM: 1 ГБ
- Объем дискового пространства: 10 ГБ

Для данной виртуальной машины был создан публичный IP адрес, при помощи которого удалось подключиться к ней по протоколу SSH (Secure Shell – безопасная оболочка)[17]. Была установлена платформа Node.js и при помощи утилиты SCP были перенесены исходные файлы приложения. Проводилась установка пакетов, необходимых для работы серверной части приложения,

после чего оно было запущено. На рисунке 10 представлен интерфейс командной строки VDS, на котором было запущено приложение.



```
denisnyux@my-diploma: ~/serv-pictura$ node --version
v16.15.0
denisnyux@my-diploma: ~/serv-pictura$ cd serv-pictura/
denisnyux@my-diploma: ~/serv-pictura$ ls
image_processing  server.js          uploader_form.js  yarn.lock
node_modules      static             uploads
package.json      uploader_base64.js yarn-error.log
denisnyux@my-diploma: ~/serv-pictura$ yarn start
yarn run v1.22.18
$ yarn nodemon server.js
$ /home/denisnyux/serv-pictura/node_modules/.bin/nodemon server.js
[nodemon] 2.0.16
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Server started at http://51.250.4.:3030
GET /api/ 200 170.148 ms - 2077
```

Рисунок 10 – Командная строка VDS

После запуска, оно было протестировано на работоспособность при помощи Postman.

Для дальнейшего размещения приложения в интернете, у доменного регистратора REG.RU было приобретено доменное имя palitra-redactor.ru.

Для настройки доменного имени VDS был использован сервис платформы Яндекс Облако – Cloud DNS. Этот сервис также доступен в консоли. В нем можно создать доменную зону и привязать ее к IP адресу выбранной виртуальной машины. Конфигурация доменной зоны, использованной при размещении приложения представлен на рисунке 11.

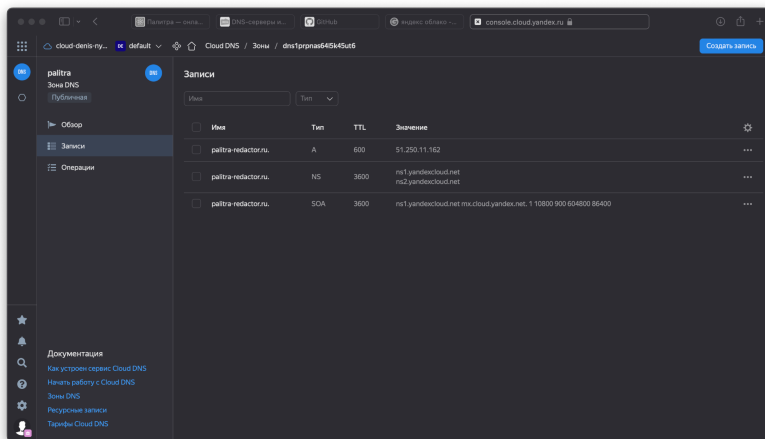


Рисунок 11 – Конфигурация доменной зоны

Через консоль управления доменными именами был настроен список DNS-серверов, обслуживающих этот домен. В консоли Яндекс Облака была создана DNS зона, в которую добавили ресурсные записи в зоне DNS.

Далее была проведена интеграция с клиентской частью приложения. Были предоставлены URL для обращения к API и проведены тесты совместной работы частей приложения. На рисунке 12 представлен пример реализации функций поворота и увеличения разрешения изображения, выполняющихся на серверной части онлайн-редактора.

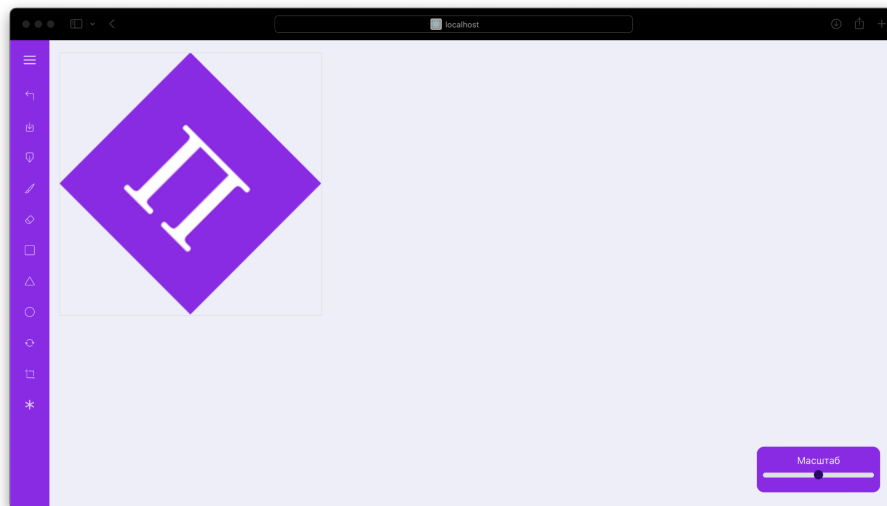


Рисунок 12 – Результат работы серверной части приложения со стороны клиентской

Для их реализации клиент отправлял следующую последовательность запросов:

1. POST запрос к URL `pictura-redactor/api/rotate`
2. GET запрос к URL `pictura-redactor/api/rotate`
3. POST запрос к URL `pictura-redactor/api/resize`
4. GET запрос к URL `pictura-redactor/api/resize`

Выводы по главе 2:

1. На основе выводов из главы 1 и технического задания, составленного в ходе преддипломной практики, были определены требования к разрабатываемому продукту.
2. В ходе проектирования API были разработаны use-case диаграмма и блок-схема приложения.
3. В соответствии с поставленной задачей ⁷ и требованиями была разработана серверная часть приложения и API по редактированию изображений;

4. В соответствии с поставленной задачей на виртуальном выделенном сервере был размещен код серверной части приложения.

ЗАКЛЮЧЕНИЕ

Цель разработать API серверной части онлайн-микросервиса по редактированию изображений на основе импортозамещенных веб-технологий, которая была поставлена в данной выпускной квалификационной работе, была выполнена.

В процессе выполнения данной работы были решены следующие задачи:

1. Осуществлен анализ современных подходов к написанию онлайн сервисов и API для обработки изображений.
2. Был проведен сравнительный анализ отечественных сервисов, предоставляющих услуги аренды виртуальных выделенных серверов;
3. Был определен набор функций, которые выполняются на серверной части сервиса обработки изображений;
4. Была разработана серверная часть приложения и API, которые реализуют функции по редактированию изображений;
5. Код серверной части приложения был размещен на виртуальном выделенном сервере.

СПИСОК ЛИТЕРАТУРЫ

1. Интернет-справочник MDN Web Docs: [сайт]. URL:
<https://developer.mozilla.org/ru/docs/Web/HTTP/Overview>
2. Интернет-справочник W3Docs: [сайт]. URL:
<https://ru.w3docs.com/uchebnik-html/metody-http-zaprosa.html>
3. Ресурс World Wide Web Consortium W3: [сайт]. URL:
<https://www.w3.org/TR/soap/>
4. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. 2-е издание. — СПб.: Питер, 2021. — 336 с.: ил. — (Серия «Бест-селлеры O'Reilly»). ISBN 978-5-4461-0590-8
5. Amundsen, Mike., Ruby, Sam., Richardson, Leonard. RESTful Web APIs: Services for a Changing World. Соединенные Штаты Америки: O'Reilly Media, 2013.v
6. Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.
7. ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию – Официальное издание. М.: Стандартинформ – 2005 год.
8. Publishing, Westerveld, Dave. API Testing and Development with Postman: A Practical Guide to Creating, Testing, and Managing APIs for Automated Software Testing. Соединенные Штаты Америки: Packt Publishing, 2021.
9. API обработки изображений Píxo: [сайт]. URL:
<https://pixoeditor.com/documentation/editing-api/>
10. API обработки изображений Pixelixe: [сайт]. URL:
<https://pixelixe.com/docs/v2/image-processing.html#image-processing-api>
11. Документация Node.js модуля Sharp: [сайт]. URL:
<https://sharp.pixelplumbing.com/performance>

- 12.Статья Modern Asynchronous JavaScript with Async and AwaitNodejs:
[сайт]. URL: <https://nodejs.dev/learn/modern-asynchronous-javascript-with-async-and-await>
- 13.Документация Yandex Compute Cloud: [сайт].
URL:<https://cloud.yandex.ru/docs/compute/>
- 14.Чепегин Илья Дмитриевич Серверный JavaScript - преимущества и недостатки node. JS // Вестник науки и образования. 2020. №12-1 (90).
URL: <https://cyberleninka.ru/article/n/servernyy-javascript-preimuschestva-i-nedostatki-node-js> (дата обращения: 23.05.2022).
- 15.RFC 4648. The Base16, Base32, and Base64 Data Encodings URL:
<https://datatracker.ietf.org/doc/html/rfc4648>
- 16.Bassett, Lindsay. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON. Тайвань: O'Reilly Media, 2015.
- 17.Ракчеев Алексей Юрьевич Работа протокола ssh на практике // Научный журнал. 2020. №3 (48). URL: <https://cyberleninka.ru/article/n/rabota-protokola-ssh-na-praktike> (дата обращения: 23.05.2022).
- 18.Уткина Любовь Ивановна Возможности виртуального выделенного сервера в поддержке сайта компании // Огарёв-Online. 2017. №2 (91).
URL: <https://cyberleninka.ru/article/n/vozmozhnosti-virtualnogo-vydelennogo-servera-v-podderzhke-sayta-kompanii> (дата обращения: 23.05.2022).
- 19.Дроздов Сергей Анатольевич, Луканина Василиса Евгеньевна Особенности проектирования серверного и клиентского программного обеспечения web-сайта с использованием rest-архитектуры // Вестник МГУП. 2016. №2. URL: <https://cyberleninka.ru/article/n/osobennosti-proektirovaniya-servernogo-i-klientskogo-programmnogo-obespecheniya-web-sayta-s-ispolzovaniem-rest-arhitektury> (дата обращения: 23.05.2022).
20. Internet Media Type registration, consistency of use. W3C. URL:
<https://www.w3.org/2001/tag/2002/0129-mime>

ПРИЛОЖЕНИЕ А

Таблица 1 - Сравнительный анализ популярных отечественных сервисов, предоставляющих услуги аренды VDS

Название сервиса	Наличие документации	Предоставляемые ОС	Функциональные особенности	Стоимость
Yandex Cloud	Присутствует	Ubuntu, Windows Server, CentOS, Debian, SLES, Fedora	<ul style="list-style-type: none"> ● Разные зоны доступности ● Виртуальные машины с GPU ● Поддержка интерфейсов CI систем ● Управление сервером по API ● Хранение данных в зашифрованном виде в соответствии с требованиями 152-ФЗ 	от 741,80 ₽ в месяц
Reg.ru	Присутствует	AlmaLinux, CentOS, Debian, Rocky Linux, Ubuntu	<ul style="list-style-type: none"> ● Высокочастотные серверы 3,7+ ГГц ● Шаблоны приложений ● Панели ISPmanager и FASTPANEL ● Снэпшоты ● Клонирование виртуального сервера 	от 370 ₽ в месяц.

			<ul style="list-style-type: none"> ● Приватные сети ● Управление сервером по API ● Остановка сервера с оплатой только за диск и IP ● Резервное копирование 	
VK Cloud Solutions	Присутствует	Windows, Linux	<ul style="list-style-type: none"> ● Безлимитный трафик 1 Gbit/s ● Балансировщик нагрузки ● Private & Public DNS ● VPN и фильтрация трафика ● Бесплатные внешние статические IP-адреса ● ЦОДы Tier III в РФ. Intel® Xeon® Gold ● Гарантированная доля vCPU 100% 	от 1110 ₽ в месяц.

ПРИЛОЖЕНИЕ Б

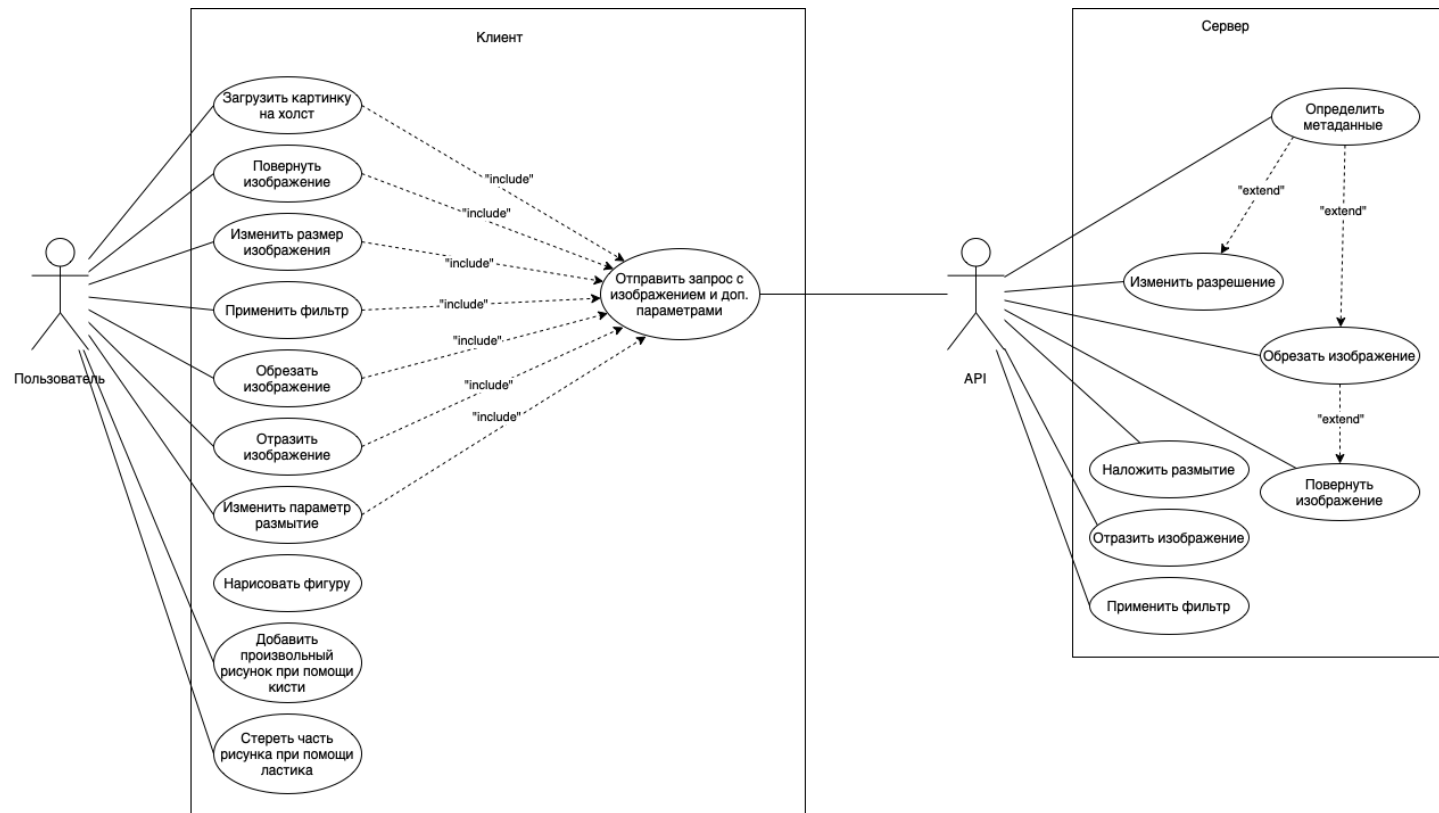


Рисунок 1 Use-case диаграмма приложения

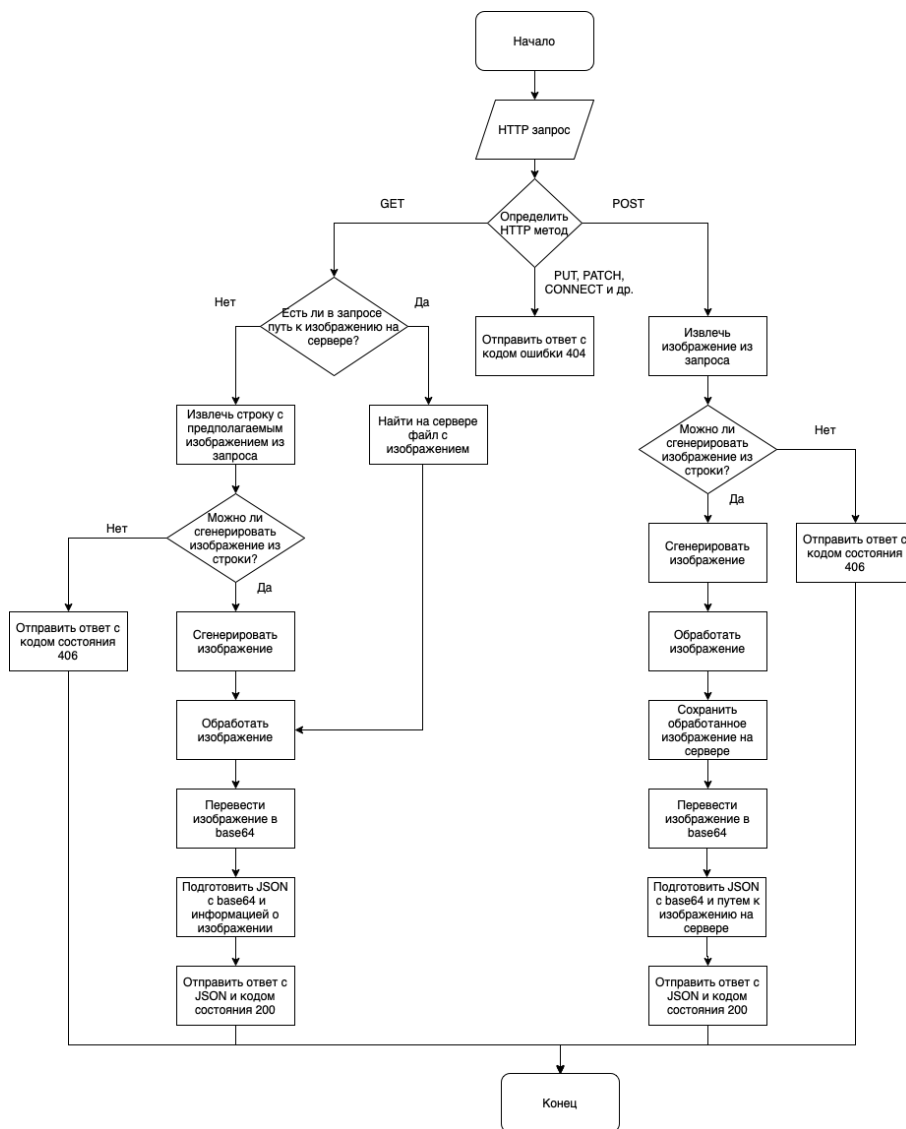


Рисунок 2 Блок-схема алгоритма по обработке изображений с сервера.