

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Прикладная математика и информатика»

**Отчет об программном проекте на тему:**  
**Серверное приложение для предоставления данных по протоколу WebDAV**

**Выполнил:**

студент группы БПМИ205  
Шитов Даниил Сергеевич

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

**Принял руководитель проекта:**

Родригес Залепинос Рамон Антонио  
Доцент  
Департамента программной инженерии  
ФКН НИУ ВШЭ

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Москва 2023

# Содержание

Аннотация	3
1 Введение	4
2 Обзор источников	5
3 Обзор WebDav методов	5
4 Версионирование	6
5 Возможные доработки	6
6 Блокировки	6
7 Требования к проекту	6
7.1 Функциональные . . . . .	6
7.2 Нефункциональные . . . . .	7
8 План работ	7
Список литературы	8

## Аннотация

Проект про создание серверного приложения, которое взаимодействует с клиентом по протоколу WebDAV. Протокол WebDAV по сути является расширением протокола HTTP, он открывает много новых возможностей для работы с сервером несколькими пользователями одновременно. Сейчас существует множество приложений, которые используют этот протокол, но в основном работа сервера сильно заточена под какую-то конкретную задачу. Основная цель этого проекта – создать простое, понятное и легкое в использовании серверное приложение, которое позволит работать с файлами по протоколу WebDAV и отслеживать историю изменений для ресурсов на сервере.

## Ключевые слова

WebDAV, DeltaV, HTTP, Версионирование, URI

# 1 Введение

Наверное, самый популярным протоколом прикладного уровня (L7), по которому сервер и клиент обмениваются информацией, сейчас является HTTP. Все потому, что он предоставляет простой и прозрачный формат взаимодействия – небольшой набор примитивных методов: GET, HEAD, POST, PUT, и т.д. Во многих случаях этого хватает для работы с ресурсами на сервере. Естественно, со временем появилась потребность в создании протокола, который бы обеспечивал более гибкую функциональность. В качестве оправдания можно привести ситуацию, когда несколько пользователей хотят одновременно работать с ресурсом на сервере, в таком случае во избежании порчи ресурса распараллелить можно только идемпотентные HTTP методы, то есть те, которые при повторном выполнении будут давать тот же результат, что и при первом. Таким образом целью при создании протокола WebDAV было дополнить уже знакомый всем протокол HTTP так, чтобы у клиентов появилась возможность распределенно работать с ресурсами на сервере. По аналогии с локальными механизмами для многопоточных программ были добавлены блокировки, которые позволяют избежать потерю информации при одновременной модификации ресурсов несколькими пользователями. Также протокол WebDAV позволяет сохранять и модифицировать мета-информацию для всех ресурсов, что делает структуру ресурсов на сервере похожей на обычную файловую систему. И последнее дополнение протокола – возможность копировать и перемещать ресурсы из одного URI в другой. Таким образом протокол WebDAV сильно расширил возможности протокола HTTP, это открыло возможности для создания более сложного взаимодействия между клиентами и сервером.

Сейчас можно встретить такие решения:

- Распределенных хранилищах данных: облачные хранилища, NAS хранилища.
- Приложениях для совместной работы с файлами и документами.
- Очень простой сервер, без дополнений к основному протоколу.

В первых двух случаях протокол обычно сильно заточен под конкретные задачи приложения. В последнем случае всю функциональность сервера приходится реализовывать разработчикам самостоятельно. Но есть виды задач, в которых готовые сложные решения неуместны по разным причинам:

- Сложность настройки и разворачивания решения на сервере.
- Покупка лицензии.

- Излишняя функциональность, которая только замедлит работу требуемых задач.

В качестве примера можно привести WebDAV сервер, который поддерживает версионирование ресурсов, то есть возможность смотреть изменения для ресурса, откатываться к прошлым версиям. Так появился протокол DeltaV, он добавляет к обычному протоколу WebDAV методы, которые позволяют реализовать версионирование. Сейчас не было найдено готовых решений, которые легко позволяют развернуть сервер с протоколом DeltaV, потому что он достаточно сложный и, возможно, не очень эффективный. Цель этого проекта реализовать серверное приложение, использующее протокол WebDav, но при этом с версионированием.

Далее будет обзор источников информации и некоторого кода, затем небольшой обзор самого протокола WebDav и план работ.

## 2 Обзор источников

Полное описание WebDAV протокола можно найти на официальном сайте [2], там описаны все методы и концепции работы протокола без реализации. Реализация же есть в виде библиотеки на Go [3], там есть все базовые методы, но они никак не приспособлены к версионированию. Описание работы упомянутой выше DeltaV описано на отдельной странице [1] официального сайта, некоторые концепции будут заимствованы оттуда.

## 3 Обзор WebDav методов

- PROPFIND позволяет получить мета-информацию ресурса. Также позволяет получить информацию о коллекции (каталоге)
- PROPPATCH позволяет изменить мета-информацию ресурса
- MKCOL позволяет создать коллекцию (каталог)
- COPY позволяет скопировать данные из одного URI в другой
- MOVE позволяет переместить данные из одного URI в другой
- LOCK позволяет взять клиенту блокировку на ресурсе
- UNLOCK позволяет снять блокировку с ресурса

## 4 Версионирование

Пока предполагается использовать `git` в качестве инструмента для версионирования, поскольку он прост и понятен в использовании. С помощью него можно отслеживать изменения в файлах и откатываться к прошлым версиям файлов. Информацию о версиях можно будет хранить в отдельной секции мета-информации у каждого ресурса.

## 5 Возможные доработки

Планируется реализовать на сервере возможность авторизации для разных пользователей. Таким образом каждый клиент сможет видеть и работать только с ресурсами, к которым у него есть доступ. Также планируется доработать протокол так, чтобы можно было легко конфигурировать доступы пользователей к ресурсам.

## 6 Блокировки

Блокировки позволяют оптимизировать скорость работы запросов, которые не изменяют ресурс, и согласовать работу запросов, которые вносят какие-то изменения в ресурс. Обычно выделяется 2 возможных типа блокировок: `exclusive`, `shared`. Первые строго запрещают другим запросам работать с ресурсом, даже читать, обычно такие блокировки берутся в том случае, когда надо изменить объект. `Shared` блокировки позволяют сразу нескольким клиентам читать объект, или применять изменения, которые не могут друг с другом конкурировать (например, дописывание в конце файла). `WebDAV` реализует оба типа блокировок.

## 7 Требования к проекту

### 7.1 Функциональные

- Реализация на сервере всех HTTP методов.
- Реализация на сервере всех WebDav методов.
- Возможность включить версионирование для ресурса на сервере.
- Возможность откатиться к прошлым версиям ресурса, если для него включено версионирование.

- Возможность получить состояние ресурса по версии.
- Возможность развернуть серверное приложение на unix подобных системах.
- Возможность создать безопасное соединение с помощью TLS
- Возможность включение авторизации на сервере.

## 7.2 Нефункциональные

- Реализация серверного приложения на языке Golang

Остальные требования сформулируются позже, когда будут окончательно выбраны способы реализации.

## 8 План работ

Сроки	Задачи
01.03	Изучить механизмы версионирования и выбрать какой-то для реализации
15.03	Реализовать простейший сервер, с возможностью простой конфигурации
20.04	Добавить возможность версионирования для ресурсов
10.05	Протестировать сервер, провести измерения эффективности сервера
КТ2	Написать отчет о проделанной работе

Таблица 8.1: План работ со сроками

## Список литературы

- [1] Geoffrey Clemm, Jim Amsden, Tim Ellison, Christopher Kaler и Jim Whitehead. *Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)*. URL: <http://webdav.org/specs/rfc3253.html#rfc.section.1.1> (дата обр. 21.03.2002).
- [2] L. Dusseault. *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*. URL: <http://webdav.org/specs/rfc4918.html> (дата обр. 12.05.2007).
- [3] *Package webdav provides a WebDAV server implementation*. URL: <https://pkg.go.dev/golang.org/x/net/webdav>.