Министерство образования и науки

федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет по лабораторной работе №2

Выполнил: Гусев Ярослав Александрович

Группа: К3320

Проверила: Марченко Е. В.

Санкт-Петербург

2024

Оглавление

[ВВЕДЕНИЕ 3](#_Toc178544208)

[Работа с Git 4](#_Toc178544209)

[Работа с Gulp 5](#_Toc178544210)

[Пример установки зависимостей 5](#_Toc178544211)

[Задание 1 6](#_Toc178544212)

[Задание 2 8](#_Toc178544213)

[ЗАКЛЮЧЕНИЕ 10](#_Toc178544214)

# ВВЕДЕНИЕ

Цель: изучить основы работы с Git и Gulp.

# Работа с Git

Клонируем удалённый репозиторий с помощью команды «git clone», переключаемся на нужную нам ветку с помощью команды «git checkout название\_ветки». Делаем нужные изменения, добавляем нужные файлы для коммита с помощью команды «git add» и далее создаем коммит командой «git commit -m название\_коммита». Отправляем изменения на удалённый репозиторий командой «git push origin название\_ветки». Данная лабораторная работа была полностью выполнена в репозитории на платформе GitHub. Команды, позволившие отправить нужный код на репозиторий представлены на рисунке 1.

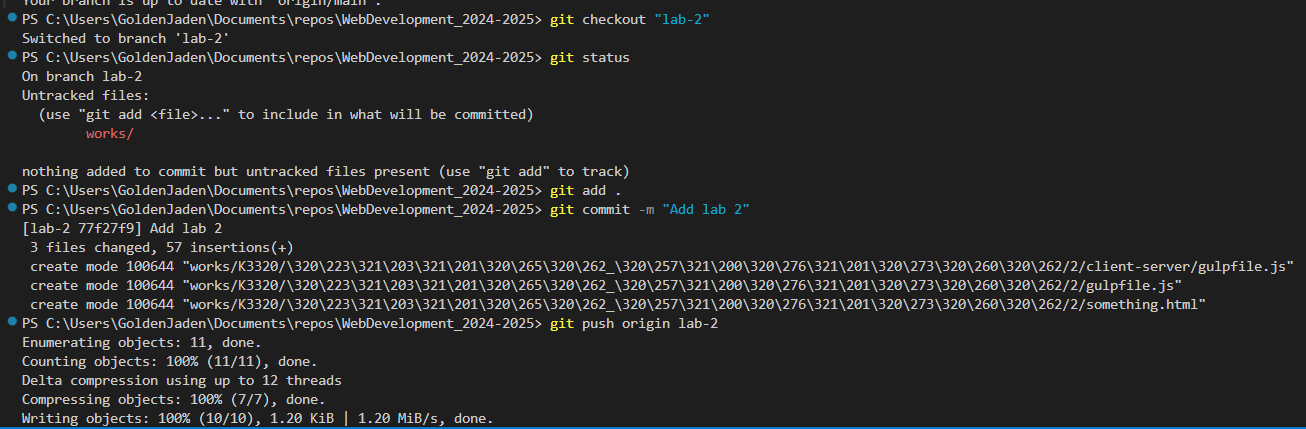


Рисунок 1 – Использование Git для отправки изменений на удалённый репозиторий

Ссылка на ветку репозитория, в которую были отправлены изменения: «<https://github.com/GoldenJaden/WebDevelopment_2024-2025/tree/lab-2>».

# Работа с Gulp

## Пример установки зависимостей

Для работы с Gulp инициализируем в пустой папке npm – пакетный менеджер платформы Node.js, который поможет нам легче управляться с зависимостями. Пропишем команду «npm init», в результате которой у нас появится файл «package.json», представленный на рисунке 2.

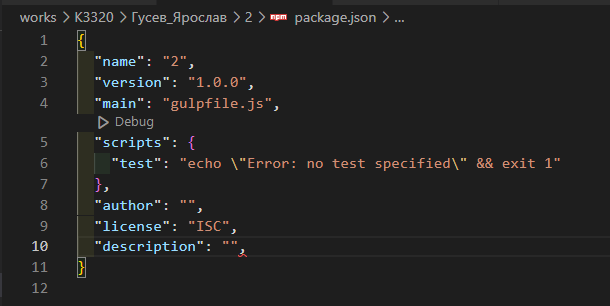


Рисунок 2 – Файл package.json после инициализации npm

Пропишем команду «npm install --save-dev gulp browserSync» для установки нужных нам зависимостей. В результате выполнения команды устанавливаются заданные пакеты и, благодаря флагу «--save-dev», в файл «package.json» добавляются нужные зависимости, как представлено на рисунке 3.

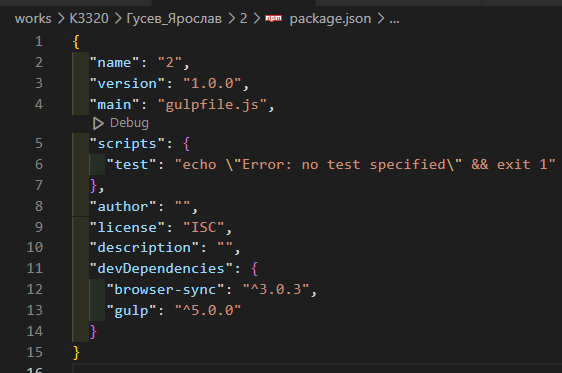


Рисунок 3 – Файл package.json с добавленными зависимостями

Благодаря этому файлу, любой разработчик, запустивший проект, может установить нужные зависимости при помощи команды «npm install» и быстро начать работу с новым проектом.

## Задание 1

Для выполнения задания требуется создать задание (task) для gulp. Для этого создадим gulpfile.js, в котором пропишем нужные инструкции. Содержимое файла представлено на рисунке 4.

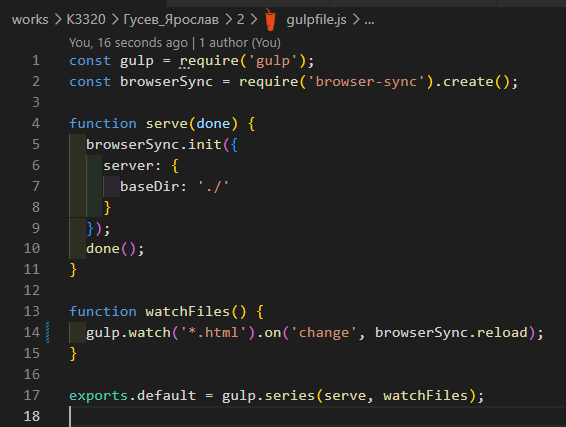


Рисунок 4 – Файл gulpfile.js

В этом файле мы сначала импортируем необходимые модули: «gulp» и «browser-sync». В функции «serve» мы создаем локальный сервер и указываем, что корневая директория сервера – текущая папка проекта.

В функции «watchFiles» мы выставляем наблюдение за всеми html файлами проекта, и в случае изменение файлов, активируется «listener», который перезагружает страницу. «BrowserSync» обеспечит синхронизацию файлов с сервером, поэтому страницы могут меняться в реальном времени в соответствии с локальными изменениями.

При помощи «exports.default» мы задаем стандартный ход действий при запуске «gulp» в командной строке. В данном случае при его запуске, у нас будут выполняться функции «serve» и «watchFiles» последовательно.

Запускаем «gulp» командой «gulp» (рисунок 5).

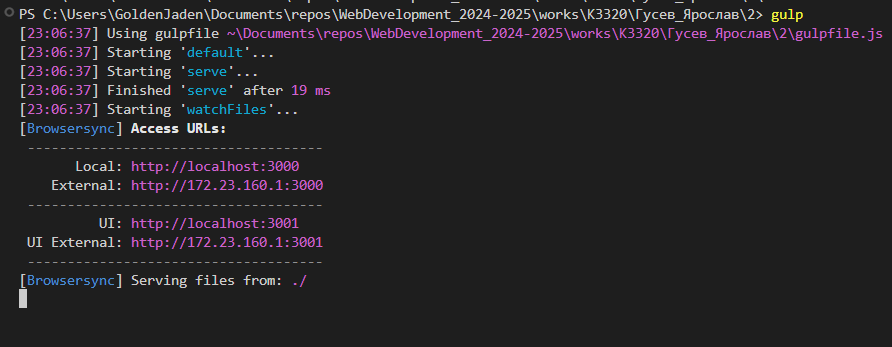


Рисунок 5 – Логи запуска gulp

На рисунке 6 представлена страница, подгруженная с файла «something.html» в корне проекта. Сервер работает на порте 3000. Страница меняется в реальном времени в соответствии с файлом.

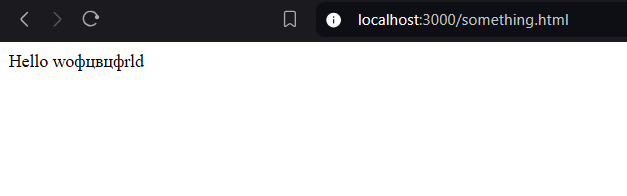


Рисунок 6 – Страница «something.html»

По порту 3001 представлен интерфейс настройки и управления «BrowserSync» (рисунок 7).

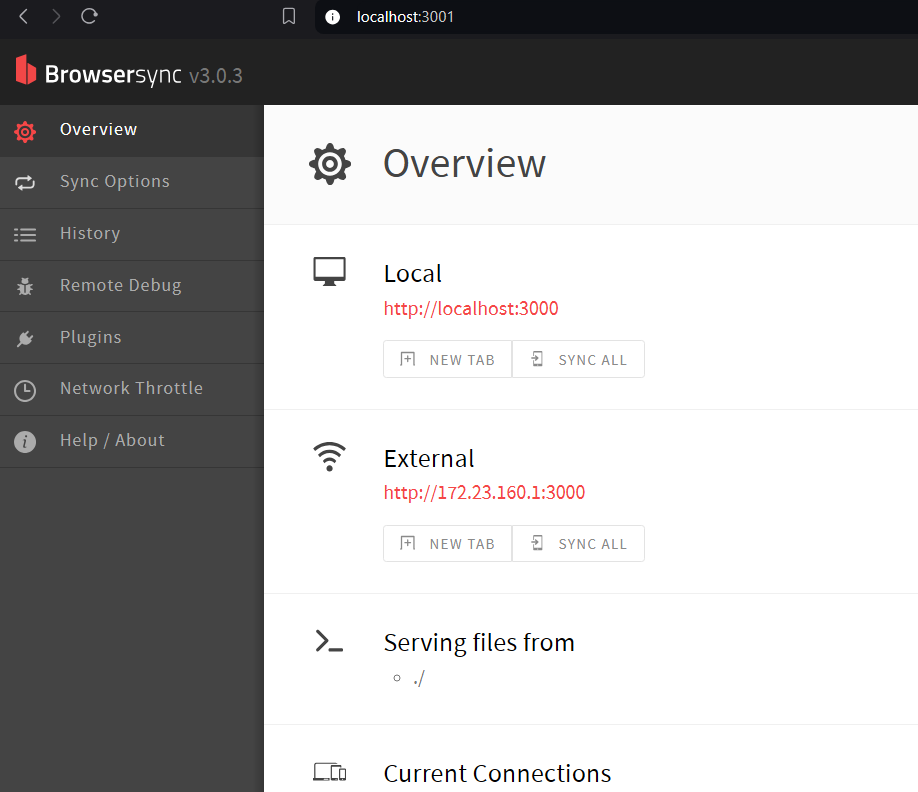


Рисунок 7 – UI BrowserSync

Задание 2

Аналогично прошлым примерам, производим инициализацию проекта и его зависимостей. Создаем «gulpfile.js», представленный на рисунке 8.



Рисунок 8 – Файл gulpfile.js

Механизм его работы прост. В списке «webPages» мы можем задать любое количество веб-страниц. Мы так же можем задать интервал, с которым эти страницы будут переключаться. Далее с помощью цикла мы проходимся по этому списку и открываем нужные нам страницы при помощи функции «open». При запуске файла при помощи «gulp» в нашем браузере открывается 3 соответствующих страницы.

# ЗАКЛЮЧЕНИЕ

В рамках данной работы мною был изучен функционал инструмента версионирования измений «Git», и программы для организации задач «Gulp».