

Санкт-Петербургский Национальный Исследовательский

Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

Web-программирование

Лабораторная работа 2

Выполнил

Скворцов И. В.

Проверила

Марченко Е. В.

Санкт-Петербург, 2024

Цели работы

Цели:

- научиться базовой работе с git
- освоить работу с gulp
- написать программу - клиент

Задание 1

В данном задании мы должны выучить основы работы с git. В моем случае git шел вместе с операционной системой, поэтому ставить его не нужно. Для создания коммитов используется команда `git commit`. Чтобы добавить комментарий к коммиту надо добавить флаг `-m`. Перед коммитом изменения надо внести в индекс коммита командой `git add <path_to_file>` на рисунке 1 можно увидеть пример истории коммитов, выведенный через `git log`.

```
commit ddca52cff71fc577563e9c2632e35f3175f2e6f8 (HEAD -> main, orig
Author: Ivan S. <mainpostskvorcov@gmail.com>
Date:   Wed Sep 25 19:14:21 2024 +0300

    update README.md

commit 6e5a0b9679334f9d58ae9949d7f04efdbca8bfa6
Author: Ivan S. <mainpostskvorcov@gmail.com>
Date:   Wed Sep 25 19:11:46 2024 +0300

    Add lab 1 redame

commit 738f5c240671a10549db05108e019fc062d09d83
Author: Ivan S. <mainpostskvorcov@gmail.com>
Date:   Wed Sep 25 19:09:08 2024 +0300

    Add lab1 report
```

Рис. 1 - история коммитов

Для синхронизации изменений с удаленным репозиторием можно использовать `git push` (если надо отправить новые изменения) или `git fetch/git pull` (для получения изменений с удаленного репозитория). Пример использования мною системы git можно увидеть по ссылке:

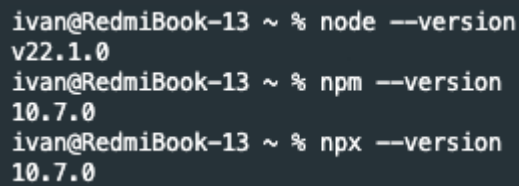
https://github.com/kloV148/2024_2025-introduction_in_routing-k3320-skvorcov_i_v

Задание 2

В данном задании нужно установить gulp и создать task.

Установка Gulp

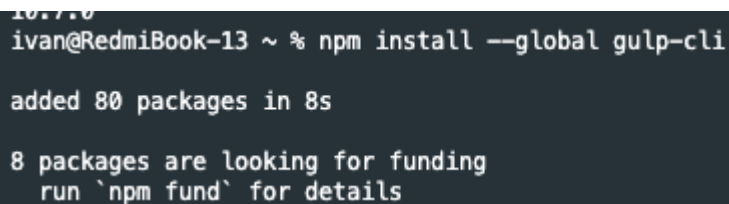
Перед установкой надо проверить, имеются ли на хост машине node, npm, npx и установить если таковые отсутствуют. В моем случае, как можно увидеть на рисунке 2, все необходимые утилиты присутствуют.



```
ivan@RedmiBook-13 ~ % node --version
v22.1.0
ivan@RedmiBook-13 ~ % npm --version
10.7.0
ivan@RedmiBook-13 ~ % npx --version
10.7.0
```

Рис. 2 - версии установленных утилит

Теперь установим gulp командой **npm install --global gulp-cli**
Результат установки можно увидеть на рисунке 3



```
10.7.0
ivan@RedmiBook-13 ~ % npm install --global gulp-cli

added 80 packages in 8s

8 packages are looking for funding
run `npm fund` for details
```

Рис. 3 - результат установки gulp

Написание task

Первым делом надо инициализировать проект и установить gulp. Команды для этого можно увидеть на рисунке 4.

Create a package.json file in your project directory

```
npm init
```

This will guide you through giving your project a name, version, description, etc.

Install the gulp package in your devDependencies

```
npm install --save-dev gulp
```

Рис. 4 - инициализация проекта

Далее создадим gulpfile и напишем таску для работы с browsersync. Код можно увидеть на рисунке 5

```
1 // Подключаем gulp и browser-sync
2 const gulp = require("gulp");
3 const browserSync = require("browser-sync").create();
4
5 // Создаем задачу для запуска сервера
6 gulp.task("browser-sync", function () {
7   // Настраиваем сервер
8   browserSync.init({
9     server: {
10       baseDir: "./src", // Указываем корневую директорию сервера как папку src
11     },
12   });
13
14   // Наблюдаем за изменениями файла index.html в папке src
15   gulp.watch("src/index.html").on("change", browserSync.reload); // Автоматическая перезагрузка при изменении HTML
16   gulp.watch("src/css/*.css").on("change", browserSync.reload); // Автоперезагрузка при изменении CSS
17   gulp.watch("src/js/*.js").on("change", browserSync.reload); // Автоперезагрузка при изменении JS
18 });
19
20 // Задача по умолчанию
21 gulp.task("default", gulp.series("browser-sync"));
```

Рис. 5 - код gulpfile

Как мы видим, он следит за изменением файлов в папке src. Запустим таску командой gulp. На рисунке 6 можно увидеть успешный запуск. Написанный код работает как задумано.

```
ivan@RedmiBook-13 lab2 % gulp
[16:15:42] Using gulpfile ~/WebDevelopment_2024-2025/works/K3320/Skvorcov_Ivan/lab2/gulpfile.js
[16:15:42] Starting 'default'...
[16:15:42] Starting 'browser-sync'...
[Browsersync] Access URLs:
    Local: http://localhost:3000
  External: http://172.28.106.217:3000
    UI: http://localhost:3001
  UI External: http://172.28.106.217:3001
[Browsersync] Serving files from: ./src
```

Рис. 6 - запуск задачи

Задание 3

В этом задании надо написать программу клиент, которая показывает web страницы одна за другой. Для этого будем использовать скрипт на python, код которого можно увидеть на рисунке 7.

```
import webbrowser
import time

# Список веб-страниц
pages = [
    "https://monkeytype.com",
    "https://www.google.com",
    "https://my.itmo.ru",
]

# Интервал между открытием страниц (в секундах)
interval = 5 # 5 секунд

# Функция для открытия страниц по очереди
def open_pages():
    for url in pages:
        print(f"Открываю страницу: {url}")
        webbrowser.open(url) # Открываем URL в браузере
        time.sleep(interval) # Ждем интервал перед открытием следующей страницы

if __name__ == "__main__":
    open_pages()
```

Рис. 7 - код клиент приложения.

Для выполнения использовались встроенные модули в python. Запуск программы осуществляется командой `python3 main.py`

Заключение

В данной работе мы выполнили поставленные цели, а именно узнали основы работы с git, познакомились с gulp и написали клиент приложения для открытия страниц