

Инвариантные работы

ИСП 1

Создание рабочего пространства, регистрация репозитория Git в качестве портфолио, анализ, обоснование выбора и настройка выбранных для работы аппаратно-программных комплексов, современных инструментальных средств удалённого доступа и веб-технологий

Рабочее пространство разработчика может состоять из следующих компонентов:

- Браузер Google Chrome, Chromium, в т.ч. нестабильные версии для разработчиков (https://www.google.com/intl/ru_ru/chrome/, <https://www.chromium.org/>)
- Браузеры Firefox, Edge, в т.ч. нестабильные версии для разработчиков
- Node.js (LTS и Current версии)
- Локальный веб-сервер XAMPP или аналогичный
- Система контроля версий Git (<https://git-scm.com/>)
- Редактор кода Visual Studio Code (<https://code.visualstudio.com/>)
- Утилита cURL
- Клиент telnet (для Windows - можно использовать PuTTY или стандартный клиент)
- SSH-клиент (для Windows - можно использовать PuTTY или PowerShell)

Настройка

Git: необходимо указать имя и адрес электронной почты:

```
git config --global user.email "mail@example.com"
git config --global user.name "username"
```

Node.js: версия выбирается с помощью `nvm use` или `n`.

ИСП 2

Настройка программного обеспечения для управления зависимостями в веб-проекте

Для управления зависимостями в проектах, разрабатываемых с помощью Node.js, используются менеджеры пакетов `npm` и `yarn`.

Менеджер пакетов `npm` входит в состав Node.js.

Для установки Yarn необходимо выполнить команду: `npm install yarn -g`

После её выполнения можно проверить установку Yarn путём выполнения команды для вывода установленной версии: `yarn -v`

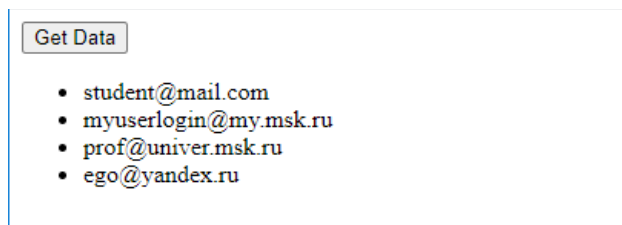
ИСП 3

Разработка приложения для асинхронного считывания данных из JSON и вывода в веб-документ путём нативной модификации дерева DOM

Данные JSON получены по адресу <https://kodaktor.ru/j/users>.

Данные считываются с помощью `fetch`. Модификация дерева DOM происходит путём вызова функции `appendChild`.

Скриншот приложения:



Код приложения (фрагмент script):

```
document.querySelector('#btn').onclick = async () => {
  let response = await fetch('https://kodaktor.ru/j/users');
  data = await response.json();
  data.users.forEach(user => {
    let ul = document.querySelector('#users');
    let li = document.createElement('li');
    li.appendChild(document.createTextNode(user.login));
    ul.appendChild(li);
  });
}
```

ИСП 4

Настройка линтера и иных средств мониторинга корректности программного кода

Настройка линтера eslint для Visual Studio Code

Установка eslint:

```
npm install -D eslint
```

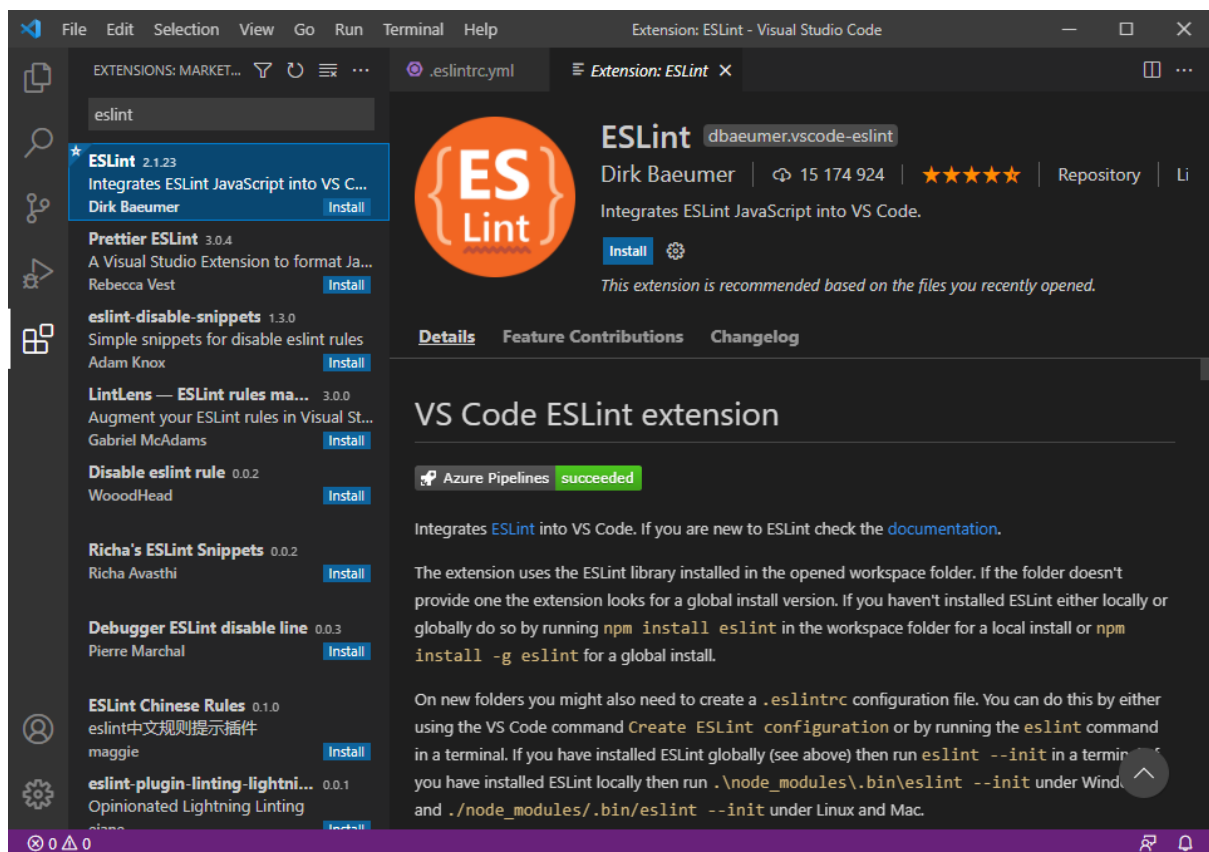
Установка плагинов eslint:

```
npm install -D eslint-config-airbnb-base eslint-plugin-import
```

Создание конфигурационного файла .eslintrc.yml:

```
extends:  
  - 'airbnb-base'  
env:  
  node: true  
  browser: true
```

Установка плагина ESLint



Исправление кода с учётом рекомендаций линтера:

VS Code interface showing a JavaScript file named `index.js`. The code contains a function definition and a call:

```
1 function hello(){console.log('Hello!');}hello();
```

An ESLint error is highlighted, indicating a "Missing space before opening brace. eslint(space-before-blocks)". The status bar at the bottom shows 4 errors and 1 warning.

VS Code interface showing the same JavaScript file `index.js` after the error has been fixed. The code is now properly formatted:

```
1 function hello() {  
2   console.log('Hello!');  
3 }  
4  
5 hello();  
6
```

The status bar at the bottom shows 0 errors and 1 warning.