

Лабораторная работа №4

Настройка современного окружения разработки

Mohamed Musa

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить процесс настройки современного окружения разработки с использованием `pnpm`, `commitizen` и `conventional commits` для управления проектами.

2 Задание

1. Настроить проект с использованием `npm package manager`
2. Сконфигурировать `commitizen` для `conventional commits`
3. Создать `package.json` с необходимыми зависимостями
4. Настроить автоматическое управление `changelog`
5. Создать `git` репозиторий и выполнить первоначальную настройку

3 Теоретическое введение

Современная разработка программного обеспечения требует использования специализированных инструментов для управления зависимостями, контроля версий и документирования изменений.

npm - это эффективный менеджер пакетов для Node.js, который использует символьные ссылки для экономии дискового пространства и ускорения установки зависимостей.

Commitizen - инструмент для создания коммитов с соблюдением conventional commits спецификации, которая обеспечивает единообразие сообщений коммитов.

Conventional Commits - спецификация для создания понятных сообщений коммитов с автоматическим семантическим версионированием.

4 Выполнение лабораторной работы

4.1 Установка и настройка rprmt

В начале работы был установлен rprmt package manager. На рисунке Рисунок ?? показан процесс настройки проекта с использованием rprmt.

```
(ceazer@fedora git-extended)$ nano package.json
(ceazer@fedora git-extended)$ git add .
(ceazer@fedora git-extended)$ git cz
Parsing JSON at /home/ceazer/work/git-extended/package.json for commitizen config failed:

(ceazer@fedora git-extended)$ nano package.json
(ceazer@fedora git-extended)$ git cz
Parsing JSON at /home/ceazer/work/git-extended/package.json for commitizen config failed:

(ceazer@fedora git-extended)$ git add .
(ceazer@fedora git-extended)$ git cz
Parsing JSON at /home/ceazer/work/git-extended/package.json for commitizen config failed:

(ceazer@fedora git-extended)$ nano package.json
(ceazer@fedora git-extended)$ git add .
(ceazer@fedora git-extended)$ git cz
Parsing JSON at /home/ceazer/work/git-extended/package.json for commitizen config failed:

(ceazer@fedora git-extended)$ git push
^C
(ceazer@fedora git-extended)$ git remote set-url git-extended git@github.com:ceazayt5/git-extended.git
error: Not a git repository. Perhaps you are in the wrong directory.
(ceazer@fedora git-extended)$ git remote -v
origin  git@github.com:ceazayt5/git-extended.git (fetch)
origin  git@github.com:ceazayt5/git-extended.git (push)
(ceazer@fedora git-extended)$ cat package.json
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:ceazayt5/git-extended.git",
  "author": "Mohammed Musa 1832248286@pfur.ru",
  "license": "CC-BY-4.0",
  "packageManager": "pnpm@10.9.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

(ceazer@fedora git-extended)$ pnpm add -g commitizen
Progress: resolved 153, reused 153, downloaded 0, added 0
```

Рисунок 4.1: Установка и настройка rprmt

4.2 Создание package.json

Был создан файл package.json с необходимой конфигурацией проекта:

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "Mohamed Musa 1032248286@rudn.ru",
  "license": "ISC",
  "packageManager": "pnpm@9.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Файл package.json показан на рисунке Рисунок ??.

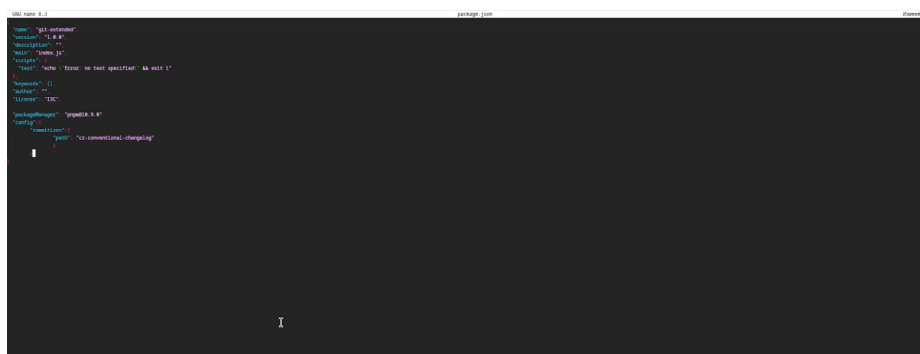


Рисунок 4.2: Конфигурация package.json

4.3 Установка commizen

Была выполнена установка commitizen с использованием npm для создания conventional commits:

```
pnpm add -g commitizen
pnpm add -D cz-conventional-changelog
```

Процесс установки показан на рисунке Рисунок ??.

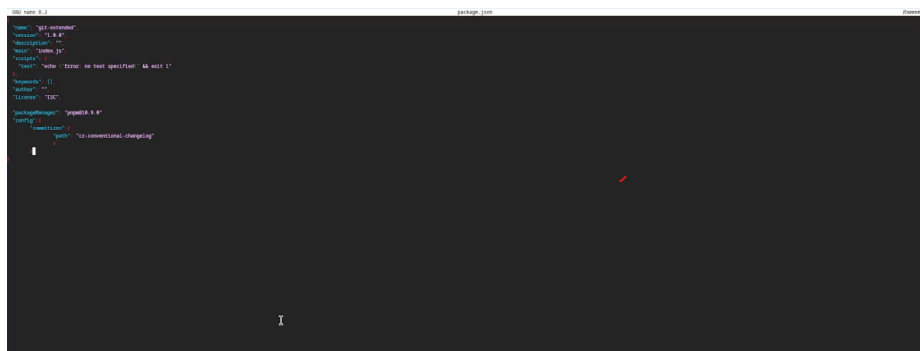


Рисунок 4.3: Установка commitizen

4.4 Создание первого релиза

Был создан первый релиз проекта с использованием стандартных команд git и npm. На рисунке Рисунок ?? показан процесс создания первого релиза.

I

Рисунок 4.4: Создание первого релиза

4.5 Работа с changelog

Были выполнены операции по добавлению записей в changelog и управлению версиями проекта. Процесс работы с changelog показан на рисунке Рисунок ??.

II

Рисунок 4.5: Управление changelog