САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №5

Выполнил:

Пиотуховский Александр

K3441

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Необходимо изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue. Научиться работать с npm и vue на основе мануала: https://docs.google.com/document/d/187UkgGNrcWqkb2aCGpkHTLgeozoElMqdVgVGMBOC9gk/edit

Ход работы

После установки NodeJS, NPM и Vue был инициализирован проект с помощью команды npm init vue@latest.

```
PS C:\Users\a1expio\Downloads\frontend-hw5-vue> npm init vue@latest

> npx
> create-vue

┌  Vue.js - The Progressive JavaScript Framework
│
◇  Project name (target directory):
│  vue-project
│
◇  Select features to include in your project: (↑/↓ to navigate, space to select, a
│  Router (SPA development), Pinia (state management), ESLint (error prevention)
│
◇  Select experimental features to include in your project: (↑/↓ to navigate, space
│  none
│
◇  Skip all example code and start with a blank Vue project?
│  No
│
Scaffolding project in C:\Users\a1expio\Downloads\frontend-hw5-vue\vue-project...
│
└  Done. Now run:

   cd vue-project
   npm install
   npm run dev

| Optional: Initialize Git in your project directory with:

   git init && git add -A && git commit -m "initial commit"

PS C:\Users\a1expio\Downloads\frontend-hw5-vue>
```

После генерации проекта были установлены зависимости с помощью команды npm install. Также были установлены bootstrap, axios и

pinia-persists с помощью команды npm i axios pinia-persists bootstrap -S. Библиотека axios нужна для HTTP-запросов к API, pinia-persists - для сохранения состояния в localStorage, bootstrap - для стилизации интерфейса. Флаг -S отвечает за сохранение в package.json.

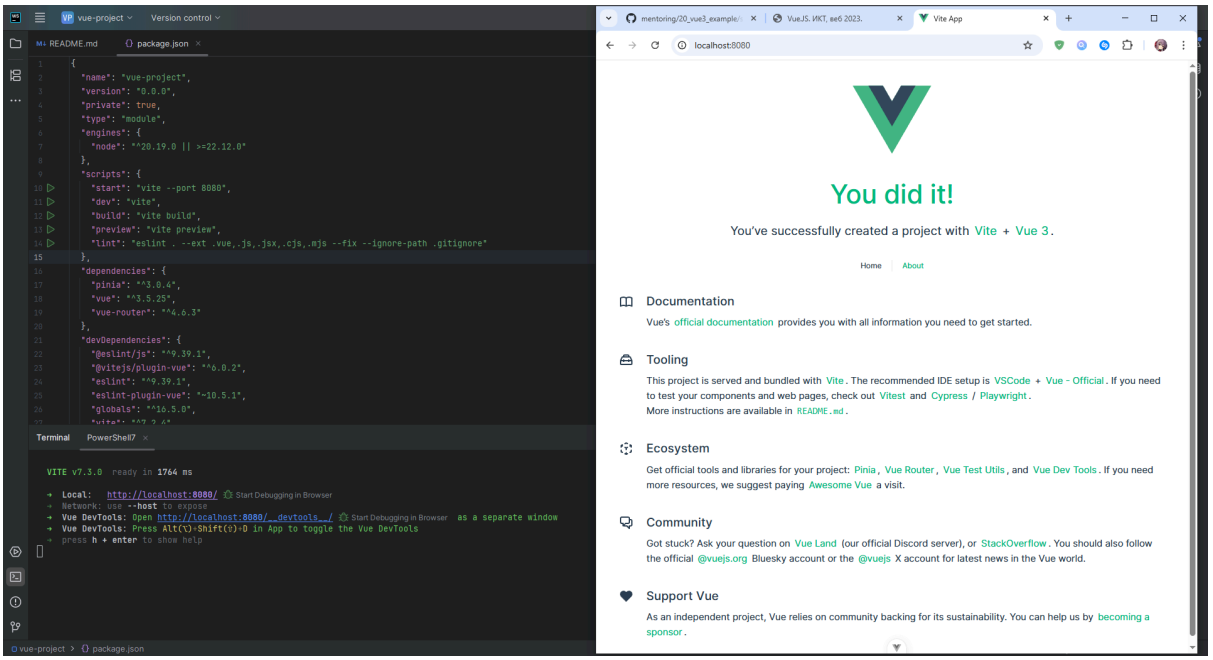
```
  ➜  press h + enter to show help
PS C:\Users\a1expio\Downloads\frontend-hw5-vue\vue-project> npm i axios pinia-persists bootstrap -S

added 72 packages, and audited 301 packages in 6s

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

found 0 vulnerabilities
PS C:\Users\a1expio\Downloads\frontend-hw5-vue\vue-project>
```

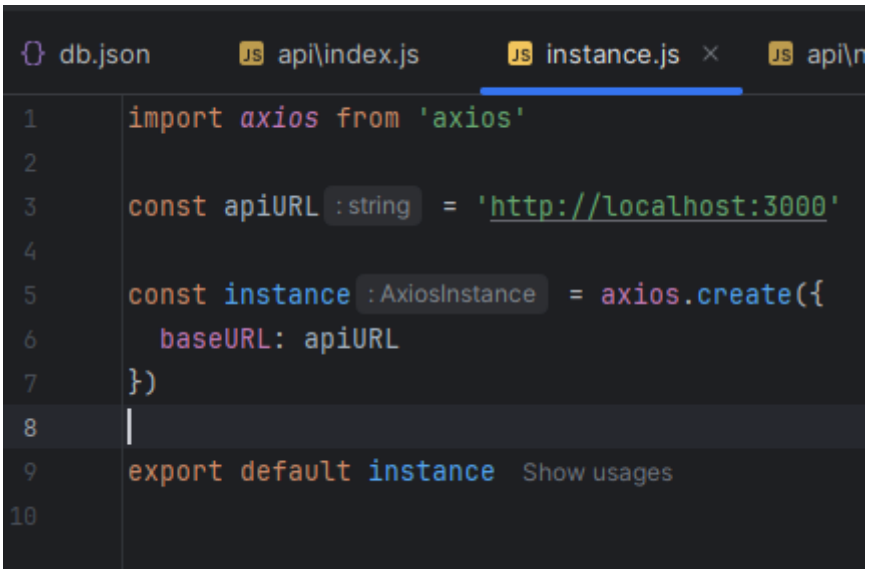
Внутри файла package.json была изменена секция scripts, как в мануале. После этого был произведён успешный запуск с помощью команды npm start.



Страница доступна по адресу localhost:8080, была отображена стандартная стартовая страница Vue-приложения, что свидетельствует о корректной инициализации проекта.

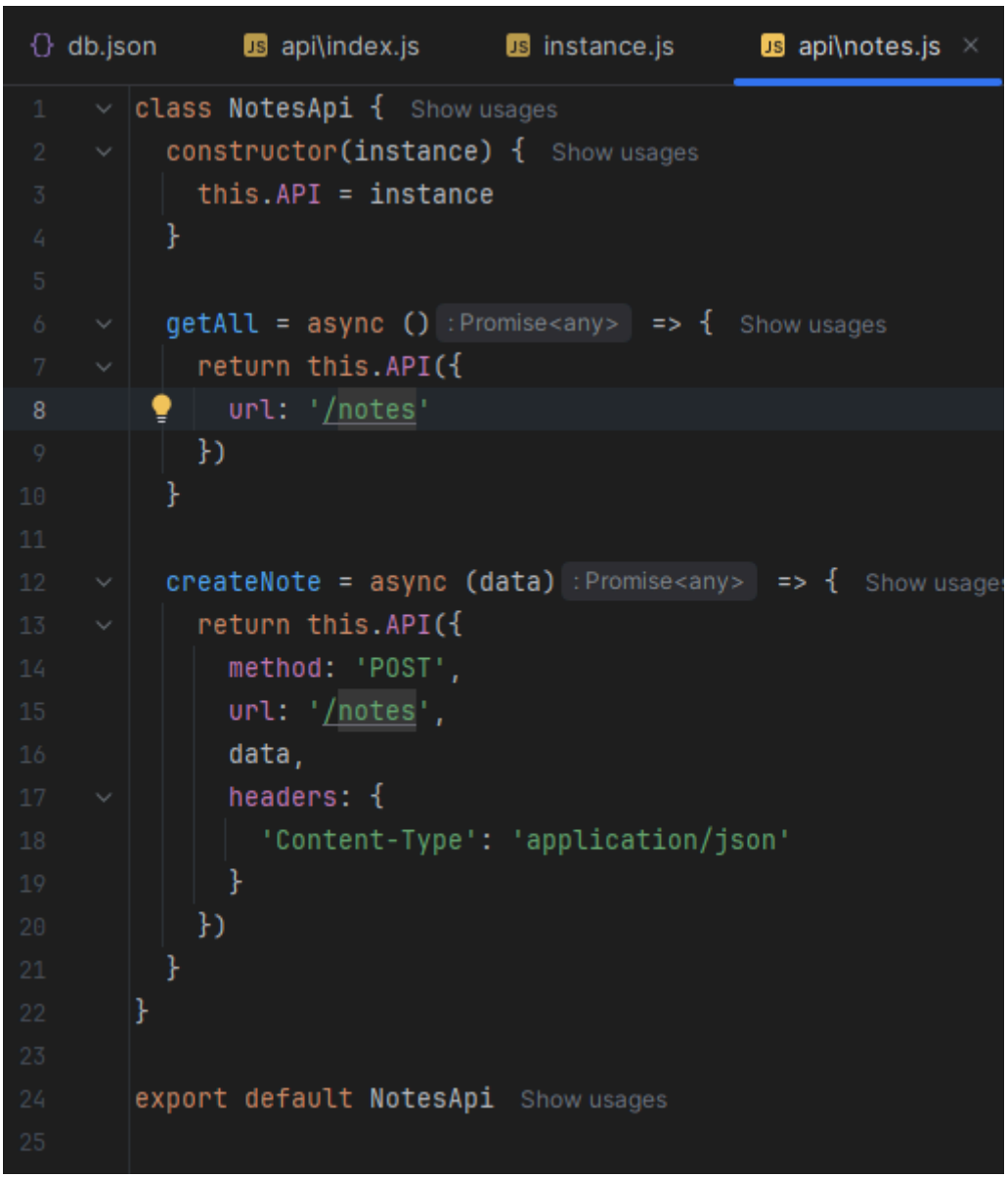
Для структурированной работы с API была создана отдельная папка src/api, где будут храниться все модули, отвечающие за HTTP-запросы. Это позволяет отделить логику работы с данными от компонентов Vue. Также была очищена папка src/stores от сгенерированных автоматически файлов.

Создан файл src/api/instance.js, который содержит настроенный экземпляр Axios с базовым URL.

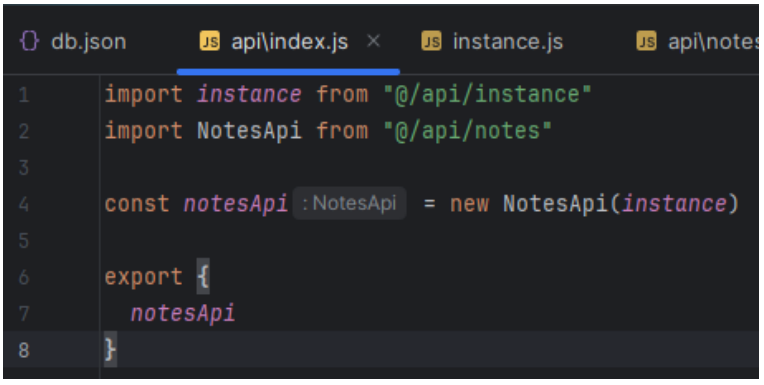


При обращении к API будет достаточно указывать только путь, полный URL будет формироваться автоматически.

Был создан файл src/api/notes.js, в котором описан класс NotesApi. Этот класс инкапсулирует методы для работы с заметками: создание новой заметки и получение списка заметок.



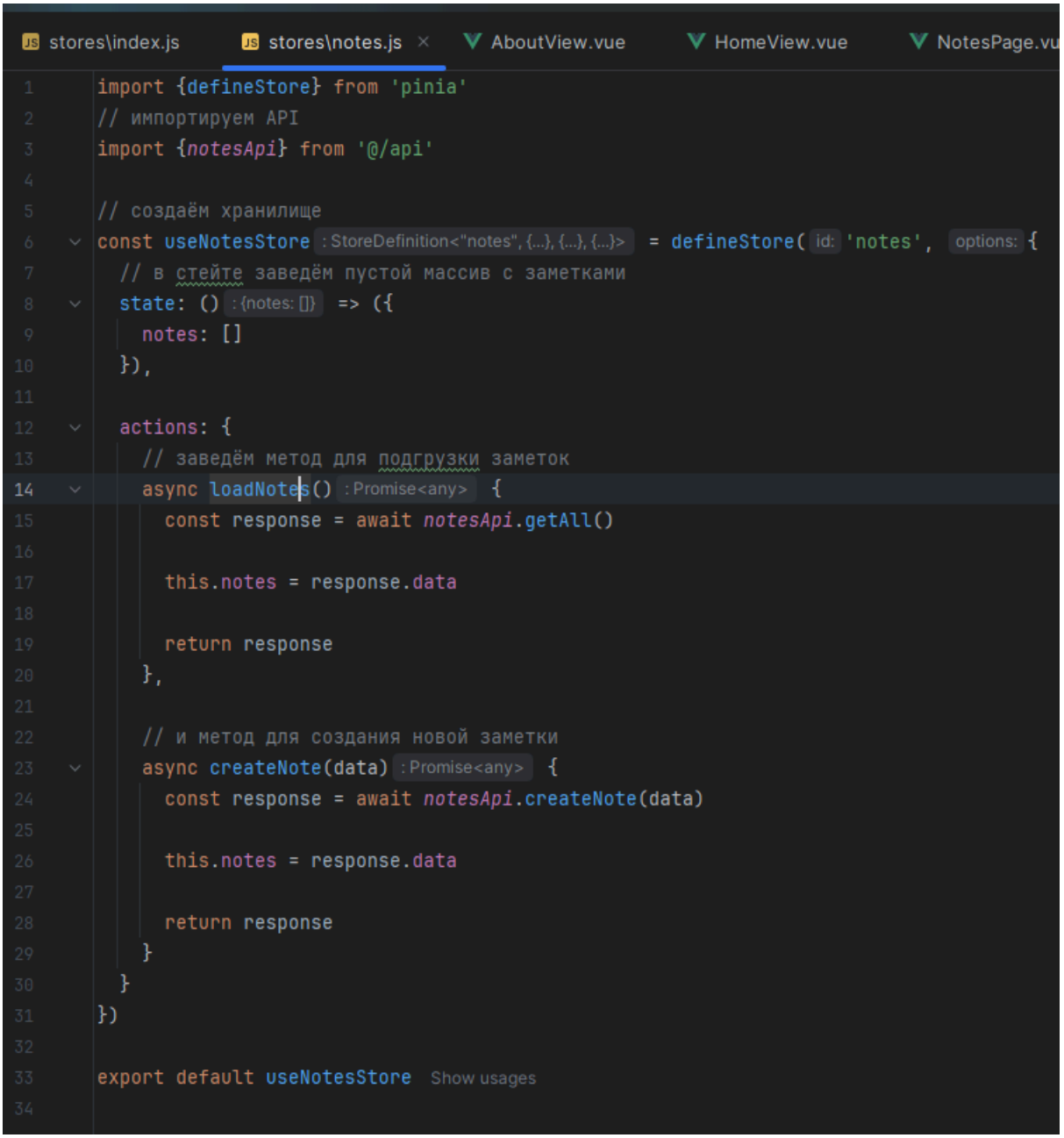
Был создан файл src/api/index.js, который выступает в роли IoC контейнера. Он связывает экземпляр axios с классом NotesApi и экспортирует готовый к использованию объект. Это позволяет импортировать API в любом месте приложения.



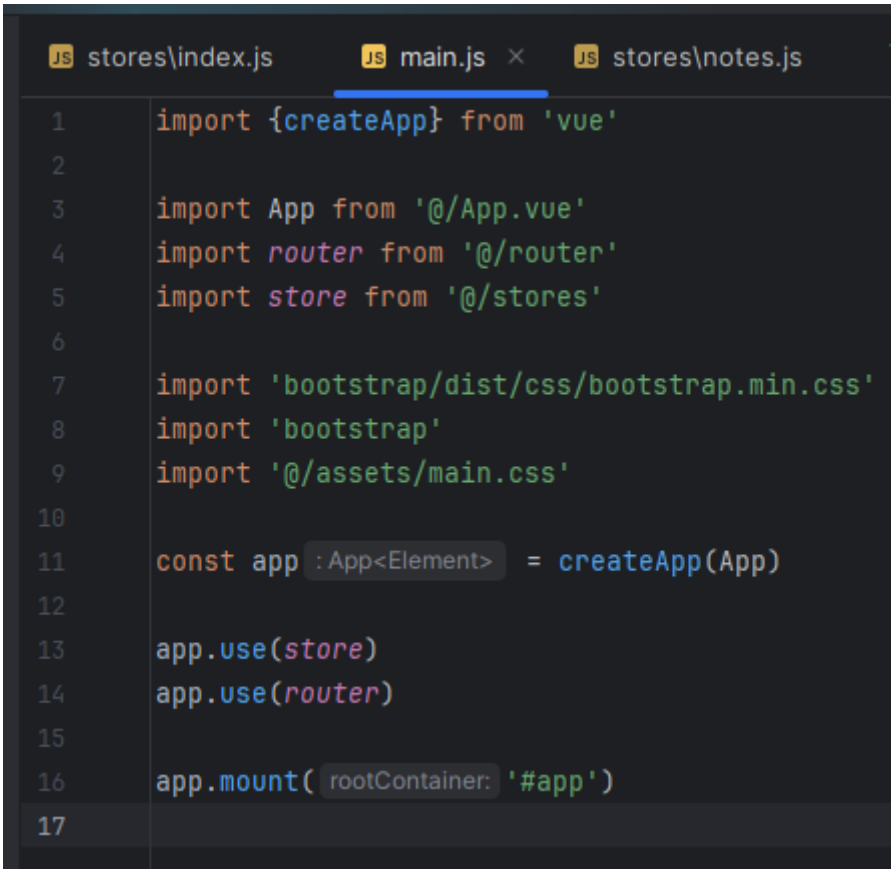
Была настроена Pinia с поддержкой сохранения данных в localStorage, чтобы состояние не терялось при перезагрузке страницы.

Был создан src/stores/index.js, который инициализирует Pinia и подключает плагин pinia-persists для автоматического сохранения состояния. Этот файл экспортирует настроенный экземпляр Pinia, который будет использоваться в main.js.

Был создан src/stores/notes.js, который содержит store для работы с заметками. Он хранит массив заметок в состоянии и предоставляет методы для их загрузки и создания. State содержит массив notes, где будут храниться все заметки. actions содержит методы loadNotes и createNote, которые вызывают соответствующие методы API и обновляют состояние.



Далее были подключены Pinia, роутер и стили bootstrap в main.js, чтобы всё заработало вместе. Здесь подключаются store с глобальным состоянием через Pinia, router для маршрутизации SPA и bootstrap для css и js компонентов bootstrap.

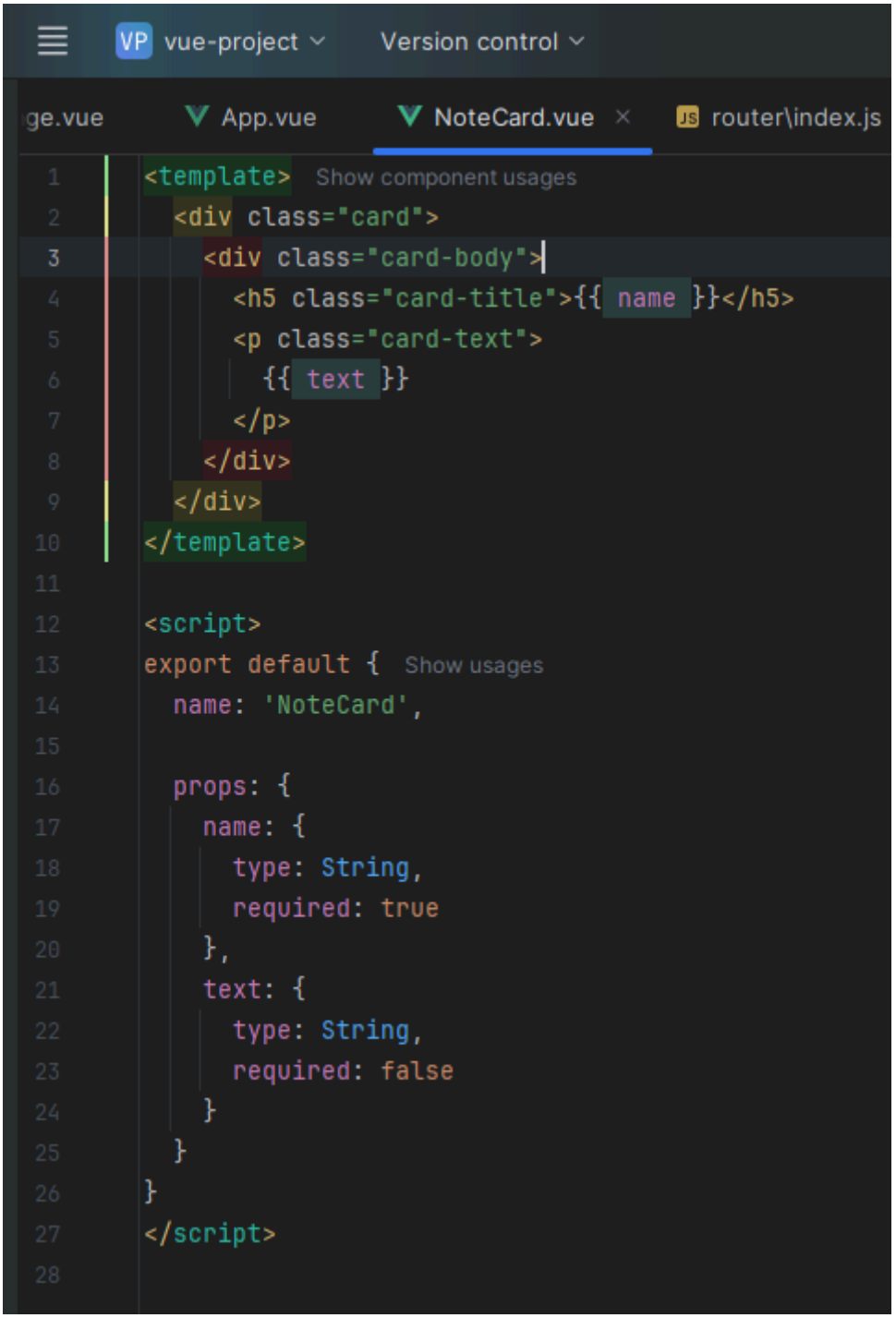


В файле src/App.vue оставили только роутер без лишней разметки. Тег <router-view> отвечает за отображение компонентов в зависимости от текущего маршрута.

Для структурирования интерфейса была создана layout-обёртка, компонент карточки заметки и основное представление страницы. Создана директория src/layouts и файл src/layouts/BaseLayout.vue.

Лейаут оборачивает содержимое в контейнер Bootstrap. Тег <slot> определяет место, куда будет вставляться содержимое дочерних компонентов. Класс container ограничивает ширину, my-2 добавляет вертикальные отступы.

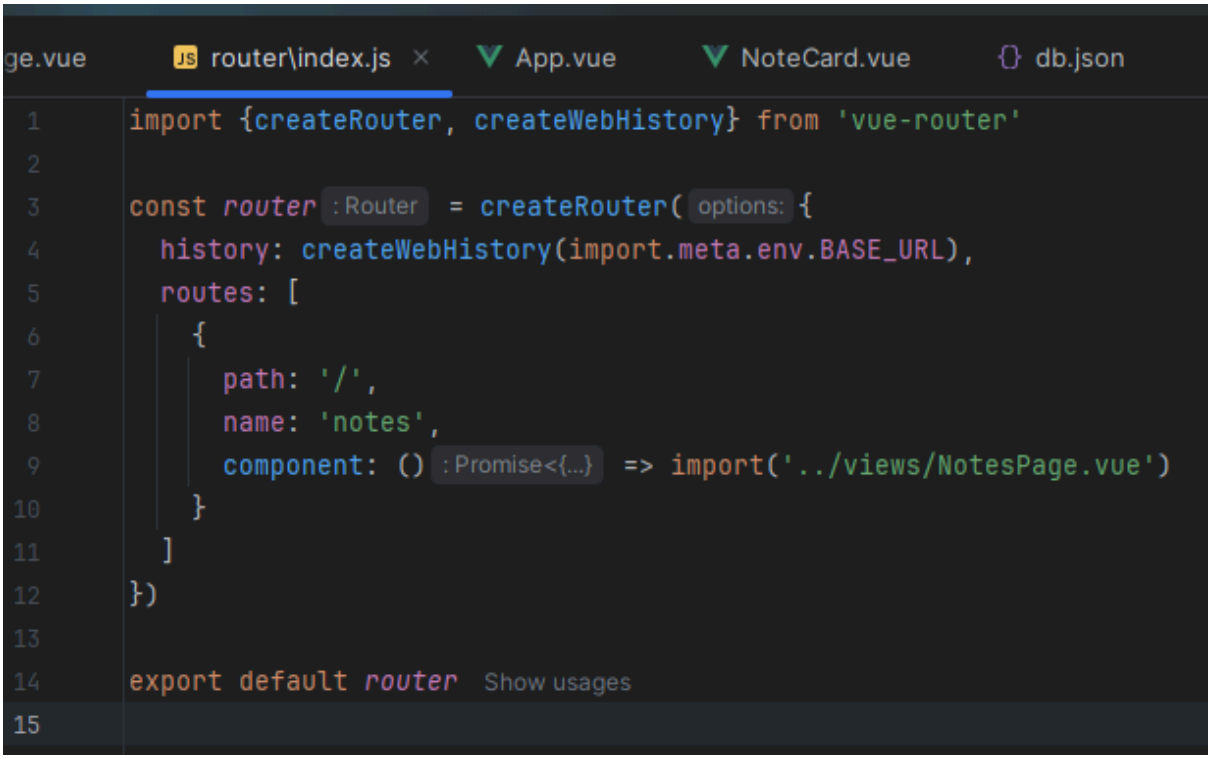
Был создан файл src/components/NoteCard.vue.



Компонент принимает два свойства через props: обязательное поле name и необязательное поле text. Код компонента разделён на три части: template, script и style.

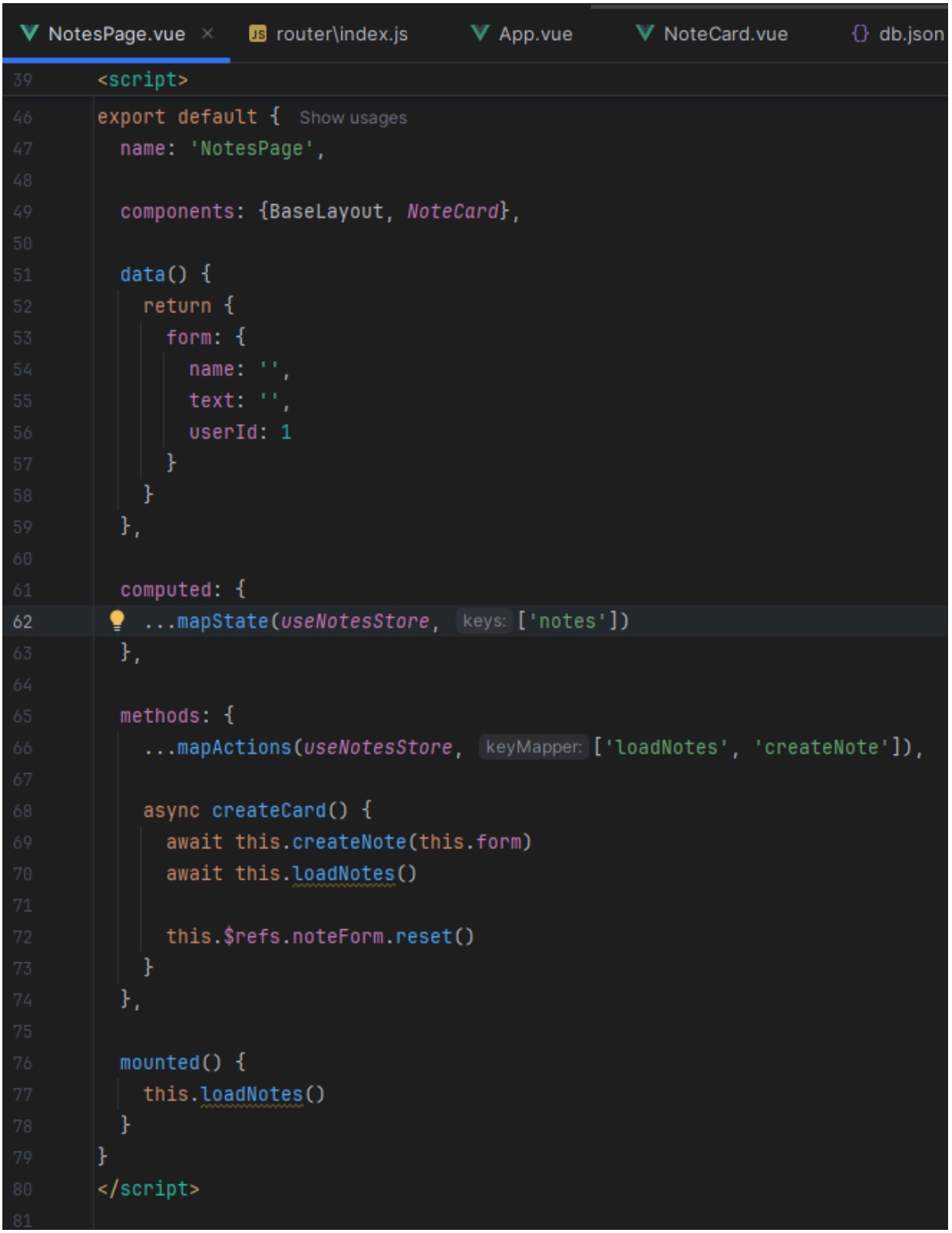
Также был создан файл src/views/NotesPage.vue. Представление содержит форму для создания заметок и список заметок. Директива v-for проходит по массиву notes и отображает каждую через NoteCard.

Роутер уже был создан при инициализации проекта. Был изменён файл src/router/index.js, чтобы главная страница отображала NotesPage. Ленивая загрузка (import()) позволяет не грузить компонент сразу, а только при переходе на маршрут; это оптимизирует производительность.



В представлении NotesPage реализована форма для создания новых заметок. Форма принимает название и текст заметки. Для связи данных формы с компонентом используется директива v-model, которая обеспечивает двунаправленное связывание данных.

Атрибут @submit.prevent заменяет стандартное поведение формы при отправке и вызывает кастомный метод createCard. Далее нужно связать представление с хранилищем.

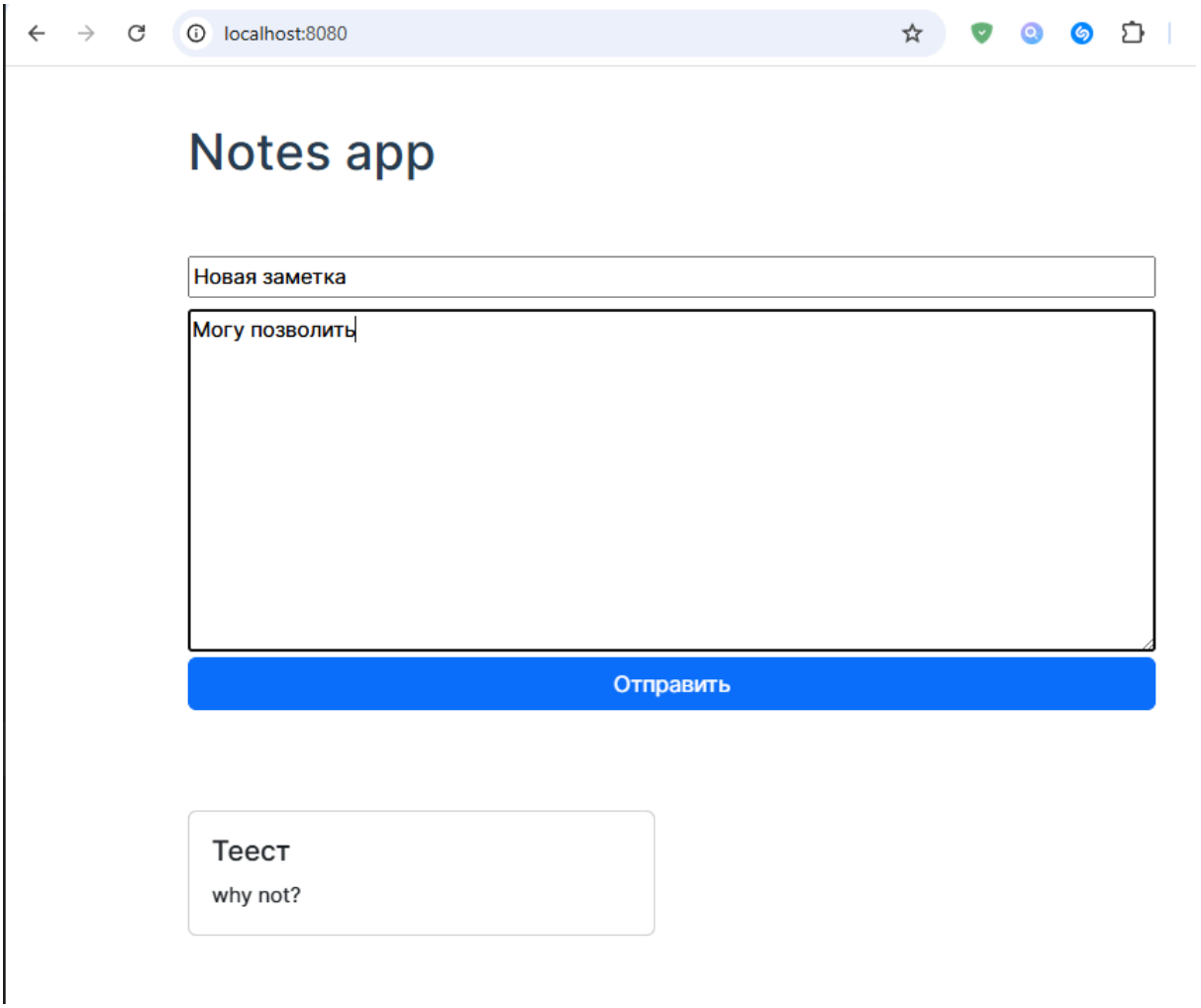


Чтобы заметки автоматически загружались при открытии страницы, был добавлен хук mounted, который срабатывает после монтирования компонента. После загрузки данные попадают в стейт, откуда через вычисляемое свойство notes, созданное с помощью mapState, они становятся доступны в шаблоне. Вычисляемые свойства автоматически

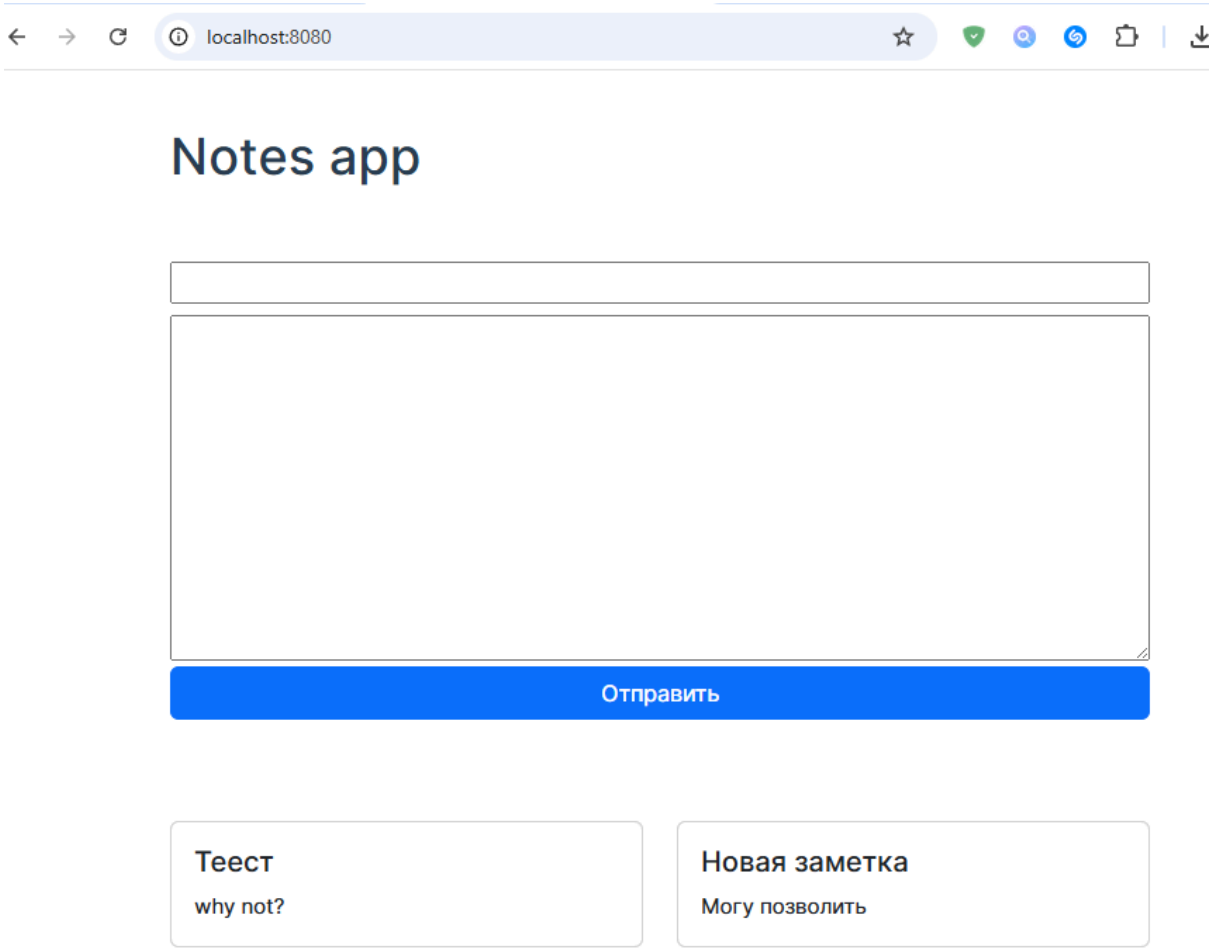
вызывают обновление компонента при изменении связанных с ними данных.

Метод последовательно вызывает createNote с данными формы, обновляет список заметок через loadNotes для получения актуальных данных с сервера, после чего сбрасывает форму.

Был запущен json-server для быстрой проверка работоспособности проекта.



Попробуем добавить новую заметку.



Вывод

В рамках лабораторной работы были изучены основные команды пакетного менеджера NPM и были получены знания по запуске проекта на Vue. Приложение подключено к локальному json-server, который используется для имитации backend-сервера. Было настроено получение списка заметок, отображение их на странице и добавление новой заметки.