САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа 5: Изучение основ работы с менеджером зависимостей npm

Выполнила:
Едигарева Дарья
Группа К3439

Проверил:
Добряков Д. И.

Санкт-Петербург

2026 г.

Инициализация проекта

Использовались команды:

npm install -g @vue/cli

npm init vue@latest

В мастере создания выбрано:

TypeScript — No

JSX — No

Vue Router — Yes

Pinia — Yes

Vitest — No

E2E — No

ESLint — Yes

Prettier — Yes

Далее:

npm install

в package.json добавлен start на порт 8080

Зависимости

Установлены:

axios — для API-запросов

pinia-persists — для сохранения состояния

bootstrap — базовые стили

Дополнительно для локального теста API:

json-server (скрипт npm run mock)

Структура проекта (ключевые файлы)

instance.js

notes.js

index.js

index.js

notes.js

index.js

NoteCard.vue

BaseLayout.vue

NotesPage.vue

main.js

App.vue

Настройка API

Axios instance — общий инстанс для запросов:

```
import axios from 'axios'



const apiURL = 'http://localhost:3000'



const instance = axios.create({

baseURL: apiURL

})



export default instance
```

Все запросы идут через localhost (line 3000).

NotesApi (DI-подход)

```
class NotesApi {

constructor(instance) {

  this.API = instance

}



getAll = async () => {

  return this.API({

    url: '/notes'

  })

}



createNote = async (data) => {

  return this.API({

    method: 'POST',

    url: '/notes',

    data,

    headers: {

      'Content-Type': 'application/json'

    }

  })

}

}



export default NotesApi
```

Класс принимает instance через конструктор и использует его для запросов.

Методы:

getAll() — получить все заметки

createNote(data) — создать заметку

IoC-контейнер

```
import instance from "@/api/instance"

import NotesApi from "@/api/notes"



const notesApi = new NotesApi(instance)



export {

notesApi

}
```

Создаётся notesApi, который экспортируется для использования в сторе.

Pinia + persist

Инициализация Pinia

```
import { persist } from 'pinia-persists'

import { createPinia } from 'pinia'



const pinia = createPinia()



pinia.use(persist())



export default pinia
```

Подключён плагин persist, чтобы состояние не сбрасывалось при обновлении страницы.

Хранилище заметок

```
import { defineStore } from 'pinia'

import { notesApi } from '@/api'



const useNotesStore = defineStore('notes', {

state: () => ({

  notes: []

}),



actions: {

  async loadNotes() {

    const response = await notesApi.getAll()



    this.notes = response.data



    return response

  },



  async createNote(data) {

    const response = await notesApi.createNote(data)



    this.notes = response.data



    return response
```

```
  }

}

})



export default useNotesStore
```

Состояние:

notes: []

Действия:

loadNotes() — грузит заметки с API и кладёт в state

createNote(data) — создаёт заметку и обновляет state

Роутинг

```
import { createRouter, createWebHistory } from 'vue-router'



const router = createRouter({

history: createWebHistory(import.meta.env.BASE_URL),

routes: [

  {

    path: '/',

    name: 'notes',

    component: () => import('../views/NotesPage.vue')

  }

]

})



export default router
```

Один маршрут / ведёт на NotesPage.vue (ленивая загрузка).

Компоненты

NoteCard.vue

```
<template>
 <div class="card">
   <div class="card-body">
     <h5 class="card-title">{{ name }}</h5>
     <p class="card-text">
       {{ text }}
     </p>
   </div>
 </div>
</template>
<script>
export default {
 name: 'NoteCard',
 props: {
   name: {
     type: String,
     required: true
   },
   text: {
     type: String,
```

```
    required: false
  }
 }
}
</script>
```

Компонент принимает props и отображает заголовок/текст заметки.

BaseLayout.vue

```
<template>
 <main class="container my-2">
  <slot />
 </main>
</template>
```

Используется slot, чтобы оборачивать контент в container.

Страница NotesPage

```
<template>
 <base-layout>
  <h1>Notes app</h1>

  <form
   ref="noteForm"
   @submit.prevent="createCard"
   class="d-flex flex-column my-5"
```

```
>
  <input
    type="text"
    v-model="form.name"
    class="my-1"
  >

  <textarea
    cols="30" rows="10"
    v-model="form.text"
    class="my-1"
  />

  <button
    type="submit"
    class="btn btn-primary"
  >
    Отправить
  </button>
</form>

<div class="row row-cols-1 row-cols-md-2 g-4 mt-5" id="notes">
  <div class="col" v-for="note in notes" :key="note.id">
```

```
      <note-card :name="note.name" :text="note.text" />
    </div>
  </div>
 </base-layout>
</template>

<script>
import { mapActions, mapState } from 'pinia'
import BaseLayout from '@/layouts/BaseLayout.vue'
import NoteCard from '@/components/NoteCard.vue'
import useNotesStore from '@/stores/notes'
export default {
 name: 'NotesPage',
 components: { BaseLayout, NoteCard },
 data() {
  return {
   form: {
    name: '',
    text: '',
    userId: 1
   }
  }
 },
```

```
 computed: {
   ...mapState(useNotesStore, ['notes'])
 },
 methods: {
   ...mapActions(useNotesStore, ['loadNotes', 'createNote']),
   async createCard() {
     await this.createNote(this.form)
     await this.loadNotes()
     this.$refs.noteForm.reset()
   }
 },

 mounted() {
   this.loadNotes()
 }
}
</script>
```

Что сделано:

форма с v-model для name и text

@submit.prevent="createCard"

mapState для notes

mapActions для loadNotes и createNote

mounted() вызывает загрузку заметок

список заметок выводится через v-for и NoteCard

Подключение Bootstrap и сборка приложения

main.js

```
import { createApp } from 'vue'



import App from '@/App.vue'

import router from '@/router'

import store from '@/stores'



import 'bootstrap/dist/css/bootstrap.min.css'

import 'bootstrap'

import '@/assets/main.css'



const app = createApp(App)



app.use(store)

app.use(router)



app.mount('#app')
```

Подключены:

bootstrap.min.css

bootstrap

main.css

Также подключены router и store.

Локальный запуск

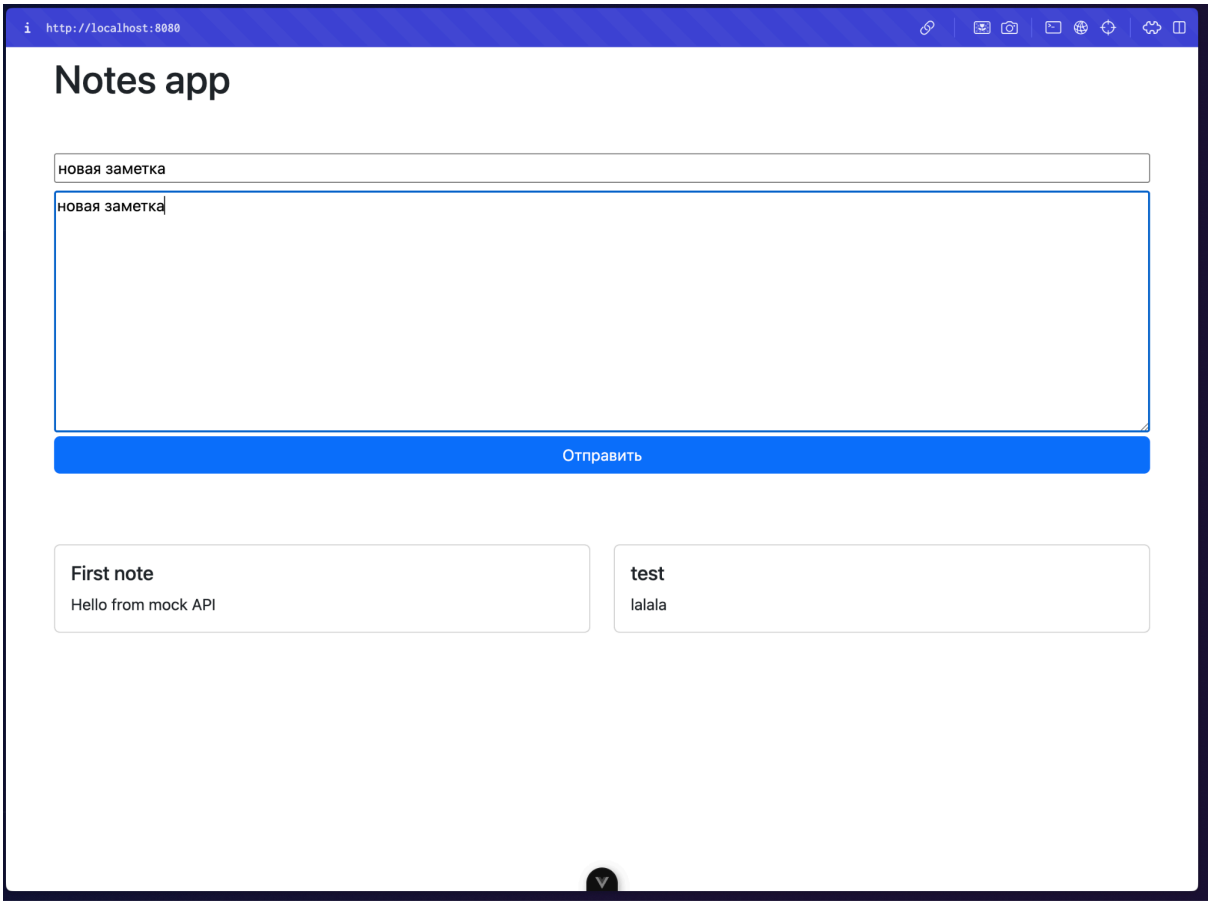
Команды:

npm run mock

npm start

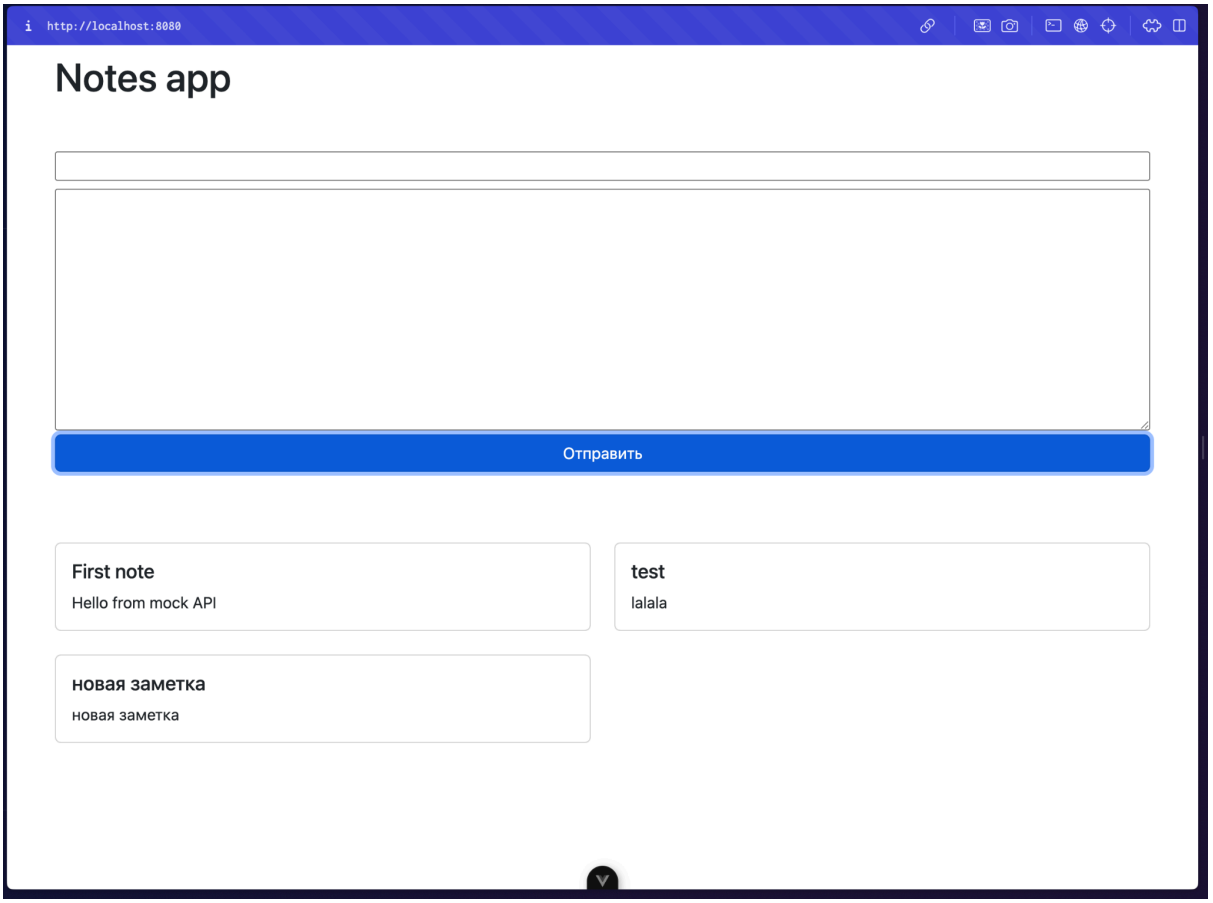
фронт: http://localhost:8080

Результат



Заметки загружаются с API

Новая заметка создаётся через форму


http://localhost:8080
Notes app
Отправить
First note
Hello from mock API
test
lalala
новая заметка
новая заметка