

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет по домашней работе

Изучение основ работы с менеджером зависимостей npm

Выполнил:

Елистратов В. Д.

Группа К32392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

## Содержание отчета

1) API	3
1.1) Notes API	3
2) Stores	4
2.1) Note Stores	4
3) Views	5
3.1) Main Page	5
4) Router	5
5) Примеры работы	7

## 1) API

### 1.1) Notes API

```
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;
```

## 2) Stores

Хранение реализовано через библиотеку Pinia

### 2.1) Note Stores

```
import { defineStore } from "pinia";
// импортируем API
import { notesApi } from "@/api";

// создаём хранилище
const useNotesStore = defineStore("notes", {
  // в стейте заведём пустой массив с заёмтками
  state: () => ({
    notes: [],
  }),

  actions: {
    // заведём метод для подгрузки заметок
    async loadNotes() {
      const response = await notesApi.getAll();

      this.notes = response.data;

      return response;
    },

    // и метод для создания новой заметки
    async createNote(data) {
      const response = await notesApi.createNote(data);

      this.notes = response.data;

      return response;
    },
  },
});

export default useNotesStore;
```

### 3) Views

#### 3.1) Main Page

```
<template>
  <div class="card">
    <div class="card-body">
      <!-- отображение свойств в шаблоне можно сделать так: -->
      <h5 class="card-title">{{ name }}</h5>
      <p class="card-text">
        {{ text }}
      </p>
    </div>
  </div>
</template>

<script>
export default {
  name: "NoteCard",

  // свойства, которые принимает компонент
  props: {
    name: {
      type: String,
      required: true,
    },
    text: {
      type: String,
      required: false,
    },
  },
};
</script>
```

### 4) Router

```
import { createRouter, createWebHistory } from "vue-router";

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  // массив с роутами
  routes: [
    // отдельный роут
    {
      path: "/",
      name: "notes",
      // реализация ленивой подгрузки представления
      // (до момента открытия этого представления,
      // оно не будет сохранено в браузере пользователя)
      component: () => import("../views/NotesPage.vue"),
    },
  ],
});

// экспортируем сконфигурированный роутер
export default router;
```



## 5) Примеры работы

