

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

**Отчет**

Домашняя работа 5 на тему: “изучение основ работы с  
менеджером зависимостей npm”

Выполнил: Безруков  
А.И. (К33402)

Проверил:  
Добряков Д.И.

Санкт-Петербург, 2022 г.

**Цель работы:** изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue.

## Ход работы:

### Инициализация проекта

#### Установка vue

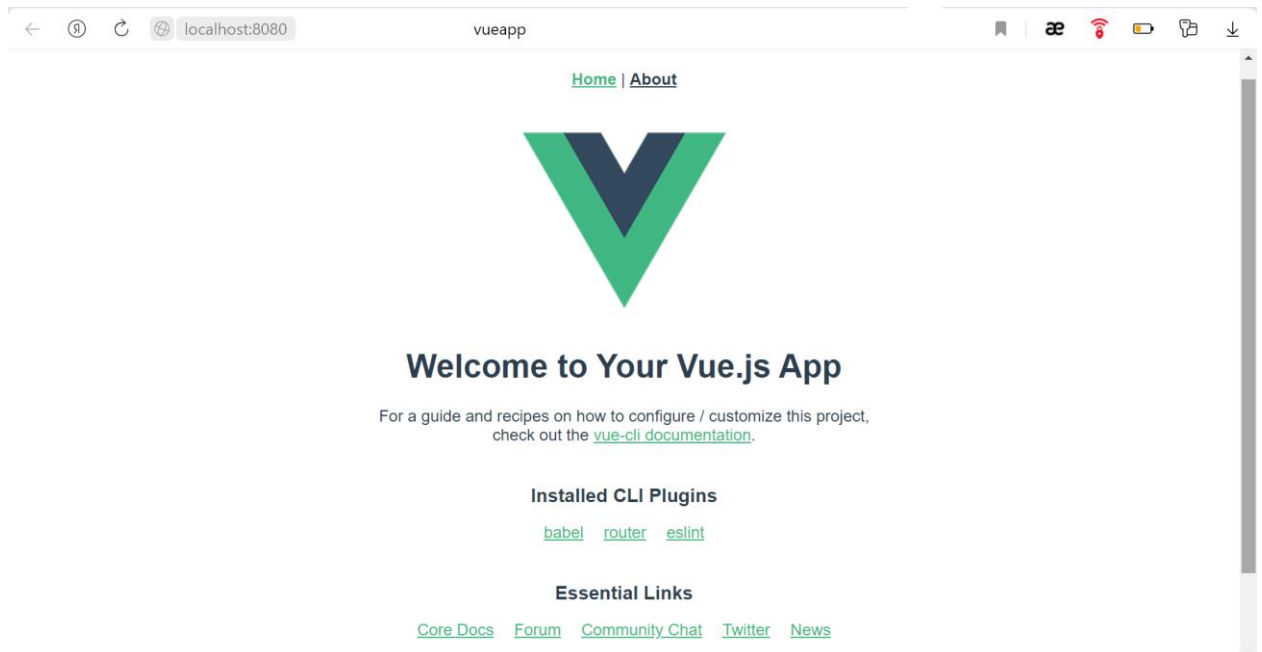
```
C:\Users\Admin\Desktop\projects\ITMO-ICT-Frontend-2022\homeworks\K33402\Bezrukov_Andrey\hw5>npm install -g @vue/cli
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated apollo-server-plugin-base@3.7.1: The 'apollo-server-plugin-base' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-errors@3.3.1: The 'apollo-server-errors' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-datasource@3.3.2: The 'apollo-datasource' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-reporting-protobuf@3.3.3: The 'apollo-reporting-protobuf' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/usage-reporting-protobuf' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-env@4.2.1: The 'apollo-server-env' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/utils.fetcher' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-types@3.7.1: The 'apollo-server-types' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated subscriptions-transport-ws@0.11.0: The 'subscriptions-transport-ws' package is no longer maintained. We recommend you use 'graphql-ws' instead. For help migrating Apollo software to 'graphql-ws', see https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws. For general help using 'graphql-ws', see https://github.com/enisdenjo/graphql-ws/blob/master/README.md
npm WARN deprecated apollo-server-express@3.11.1: The 'apollo-server-express' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-core@3.11.1: The 'apollo-server-core' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.

added 849 packages, and audited 850 packages in 3m

64 packages are looking for funding
  run 'npm fund' for details

4 vulnerabilities (2 moderate, 2 high)
```

#### Запуск проекта



### Компоненты, Роутинг

#### Компонента GreetingCard.vue

The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure with a 'src' directory containing 'components'. The 'GreetingCard.vue' file is selected. The Editor shows the code for 'GreetingCard.vue' with the following content:

```
1 <template>
2   <h1>Привет!</h1>
3 </template>
4
5 <script>
6   export default {
7     name: 'GreetingCard'
8   }
9 </script>
10
11 <style>
12 </style>
13
```

Представление Greeting.vue

The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure with a 'src' directory containing 'views'. The 'Greeting.vue' file is selected. The Editor shows the code for 'Greeting.vue' with the following content:

```
1 <template>
2   <section>
3     <greeting-card />
4
5     <h2>Это простой проект на Vue.</h2>
6   </section>
7 </template>
8
9 <script>
10   import GreetingCard from '@components/GreetingCard'
11
12   export default {
13     name: 'Greeting',
14
15     components: {
16       GreetingCard
17     }
18   }
19 </script>
20
21 <style>
22 </style>
```

Добавление представления в src/router/index.js:

```
import Greeting from '@/views/Greeting.vue'

Vue.use(VueRouter)

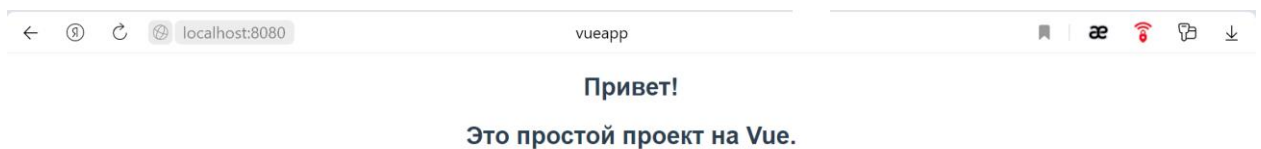
const routes = [
  {
    path: '/',
    name: 'Greeting',
    component: Greeting
  },

```

Изменение App.vue

```
1 <template>
2   <main id="app">
3     <router-view/>
4   </main>
5 </template>
6
```

Проект



Работа с v-model, props, v-if, v-for, methods, created

Заведение input, добавление метода data() в Greeting.vue

GreetingCard.vue U

Greeting.vue U X

App.vue U

JS index.js U

src &gt; views &gt; Greeting.vue &gt; {} template &gt; section &gt; form &gt; label

```
1  <template>
2    <section>
3      <greeting-card />
4      <h2>Это простой проект на Vue.</h2>
5      <form
6        @submit.prevent="submitUsername"
7        ref="form"
8      >
9        <label for="username">
10         Введите своё имя:
11       </label>
12       <input
13         v-model="username"
14         id="username"
15         name="username"
16         type="text"
17         placeholder="Иван Иванов"
18       />
19     </form>
20   </section>
21 </template>
22
23 <script>
24   import GreetingCard from '@components/GreetingCard'
25
26   export default {
27     name: 'Greeting',
28
29     components: {
30       GreetingCard
```

```

31     },
32
33     data: () => ({
34         username: '',
35
36         savedUsername: ''
37     }),
38
39     methods: {
40         submitUsername () {
41             // сохраним username в localStorage
42             localStorage.setItem('username', this.username)
43
44             // сохраним его в отдельную переменную,
45             // для дальнейшей передачи в компоненту
46             // GreetingCard
47             this.savedUsername = this.username
48
49             // очистим форму
50             this.$refs.form.reset()
51         }
52     },
53
54     created () {
55         // если localStorage содержит значение по ключу
56         // username, то запишем его в наши переменные
57         if (localStorage.getItem('username')) {
58             this.savedUsername = localStorage.getItem('username')
59
60             this.username = this.savedUsername

```

```

61     }
62 }
63 }
64 </script>
65
66 <style>
67 </style>

```

Добавление props в GreetingCard.vue

```
src > components > GreetingCard.vue > {} src
1  <template>
2    <h1>Привет!</h1>
3  </template>
4
5  <script>
6    export default {
7      name: 'GreetingCard',
8      props: {
9        username: String
10     }
11   }
12 </script>
13
14
15 <style>
16 </style>
17
```

Добавление передачи переменной

```
<greeting-card :username="savedUsername" />
```

Использование v-if (избавление от запятой)

```
1  <template>
2    <h1>Привет<span v-if="username">, {{ username }}</span>!</h1>
3  </template>
4
```

Окно ввода имени

Привет, Андрей Б!

Это простой проект на Vue.

Введите своё имя:

GreetingList.vue

▼ GreetingCard.vue U

▼ GreetingList.vue U X

▼ Gr

src > components > ▼ GreetingList.vue > {} style

```
1   <template>
2     <ul class="greeting-list">
3       <li
4         v-for="greeting in greetings"
5         :key="greeting.id"
6       >
7         {{ greeting.text }}
8       </li>
9     </ul>
10  </template>
11
12  <script>
13    export default {
14      name: 'GreetingList',
15
16      props: {
17        greetings: Array
18      }
19    }
20  </script>
21
22  <style>
23    .greeting-list li {
24      text-align: left;
25    }
26  </style>
27
```

Вывод компоненты в представление Greeting



```
<h3>Варианты приветствий:</h3>

<div class="greeting-list-wrapper">
  <greeting-list :greetings="greetings" />
</div>
```

Добавление массива greetings в data()

```
data: () => ({
  username: '',
  savedUsername: '',
  greetings: [
    { id: 1, text: 'Привет' },
    { id: 2, text: 'Hello' },
    { id: 3, text: 'Hola' }
  ]
}),
```

Добавление стилей в представление

```
<style>
.greeting-list-wrapper {
  display: flex;
}

.greeting-list-wrapper .greeting-list {
  margin: auto
}
</style>
```

Приветствия в проекте:

Привет, Андрейка!  
Это простой проект на Vue.

Введите своё имя:

Варианты приветствий:

- Привет
- Hello
- Hola

## Подключение UI-фреймворка (vuetify)

После установки vuetify меняем Greeting.vue:

```
<template>
  <section>
    <greeting-card :username="savedUsername" />

    <h2>Это простой проект на Vue.</h2>

    <v-form
      @submit.prevent="submitUsername"
      ref="form"
      class="my-2"
    >
      <v-row>
        <v-col cols="3" class="mx-auto">
          <v-text-field
            label="Введите своё имя"
            v-model="username"
            name="username"
            placeholder="Иван Иванов"
          />
        </v-col>
      </v-row>

    </v-form>

    <h3>Варианты приветствий:</h3>

    <div class="my-2 greeting-list-wrapper">
      <greeting-list :greetings="greetings" />
    </div>
  </section>
</template>

<script>
import GreetingCard from '@components/GreetingCard'
import GreetingList from '@components/GreetingList'

export default {
  name: 'Greeting',
```

```

components: {
  GreetingCard,
  GreetingList
},

data: () => ({
  username: '',

  savedUsername: '',

  greetings: [
    { id: 1, text: 'Привет' },
    { id: 2, text: 'Hello' },
    { id: 3, text: 'Hola' }
  ]
}),

methods: {
  submitUsername () {
    // сохраним username в localStorage
    localStorage.setItem('username', this.username)

    // сохраним его в отдельную переменную,
    // для дальнейшей передачи в компоненту
    // GreetingCard
    this.savedUsername = this.username

    // очистим форму
    this.$refs.form.reset()
  }
},

created () {
  // если localStorage содержит значение по ключу
  // username, то запишем его в наши переменные
  if (localStorage.getItem('username')) {
    this.savedUsername = localStorage.getItem('username')

    this.username = this.savedUsername
  }
}
}
</script>

<style>
.greeting-list-wrapper {
  display: flex;
}

.greeting-list-wrapper .greeting-list {
  margin: auto

```

```
}  
</style>
```

App.vue – главная компонента

```
▼ GreetingCard.vue U  JS main.js U  ▼ GreetingList.vue U  ▼ Greeting.vue U  ▼ App.vue U X  
src > ▼ App.vue > ...  
1  <template>  
2    <v-app>  
3      <v-app-bar  
4        app  
5        color="primary"  
6        dark  
7      >  
8        <div class="d-flex align-center">  
9          <h1 class="headline">Простой проект на Vue</h1>  
10         </div>  
11       </v-app-bar>  
12  
13       <v-main class="d-flex align-center text-center">  
14         <router-view />  
15       </v-main>  
16     </v-app>  
17   </template>  
18  
19   <script>  
20     export default {  
21       name: 'App',  
22  
23       data: () => ({  
24         //  
25       })  
26     }  
27   </script>
```

Добавление карточки в vue.js

```
25     <v-row>  
26       <v-col cols="4" class="mx-auto">  
27         <v-card  
28           elevation="2"  
29           class="px-2 py-5"  
30           color="primary"  
31           dark  
32         >  
33           <greeting-list :greetings="greetings" />  
34         </v-card>  
35       </v-col>  
36     </v-row>  
37
```

Проект

Привет, Андрей!  
Это простой проект на Vue.

Введите своё имя  
Андрей

- Привет
- Hello
- Hola

## Запросы к API через axios

После установления библиотеки axios добавляем зависимости в main.js

```
src > router > JS index.js > ...
1  import Vue from 'vue'
2  import App from './App.vue'
3  import router from './router'
4  import vuetify from './plugins/vuetify'
5  import axios from 'axios'
6  import VueAxios from 'vue-axios'
7
8  Vue.config.productionTip = false
9  Vue.use(VueAxios, axios)
10
11  new Vue({
12    router,
13    vuetify,
14    axios,
15    render: h => h(App)
16  }).$mount('#app')
```

EpicItemCard компонента

```
src > components > EpicItemCard.vue > ...
1  <template>
2    <v-card
3      elevation="2"
4      class="px-2 py-5"
5    >
6    <v-row>
7      <v-col cols="4">
8        <v-img :src="epicItem.image" />
9      </v-col>
10
11     <v-col cols="6">
12       <v-card-text>
13         {{ epicItem.caption }}
14       </v-card-text>
15     </v-col>
16   </v-row>
17 </v-card>
18 </template>
19 <script>
20   export default {
21     name: 'EpicItemCard',
22
23     props: {
24       epicItem: Object
25     }
26   }
27 </script>
```

## Представление Nasa.vue

```
<template>
<section>
  <v-row>
    <v-col cols="6" class="mx-auto">
      <epic-item-card
        v-for="epicItem in epicItems"
        :key="epicItem.identifier"
        :epic-item="epicItem"
        class="my-2"
      />
    </v-col>
  </v-row>
</section>
</template>

<script>
import EpicItemCard from '@components/EpicItemCard.vue'
const apiKey = 'pv7mgIb4MCWKU5a8q0Zm8phM3KZiCaKdOXA0Hmth'
const apiUrl = 'https://api.nasa.gov/EPIC'

export default {
  components: { EpicItemCard },
  name: 'Greeting',

  data: () => ({
    epicItems: []
  }),

  methods: {
    async getEpicItems () {
      try {
        // выполним запрос на получение списка всех
        // фотографий земли
        const response = await this.axios
          .get(`${apiUrl}/api/natural?api_key=${apiKey}`)
      } catch (error) {
        console.error(error)
      }
    }
  }
}
```

```

        // если статус ответа не 200, то выкинем
        // ошибку
        if (response.status !== 200) {
            throw new Error(response.error)
        }

        // пройдемся по полученному массиву данных и
        // приведём его к тому виду, с которым будет
        // удобно работать
        const epicItems = response.data.map((epicItem) => {
            const date = new Date(epicItem.date)

            // собираем дату в нужном виде
            const year = date.getFullYear()

            const month = String(date.getMonth() + 1).length > 1
                ? date.getMonth() + 1
                : `0${date.getMonth() + 1}`

            const day = String(date.getDate()).length > 1
                ? date.getDate()
                : `0${date.getDate()}`

            epicItem.date = `${year}/${month}/${day}`

            // получаем картинку
            epicItem.image =
                `${apiUrl}/archive/natural/${epicItem.date}/png/${epicItem.image}.png?api_key=${apiKey}`

            return epicItem
        })

        this.epicItems = epicItems
    } catch (e) {
        console.error('AN API ERROR', e)
    }
}
},

created () {
    this.getEpicItems()
}
}
</script>

<style>
</style>

```

Добавление представления в index.js

```

import Vue from 'vue'
import VueRouter from 'vue-router'

// импортируем все представления, по
// которым будем навигировать пользователя
import Greeting from '@/views/Greeting.vue'
import Nasa from '@/views/Nasa.vue'

```

```

Vue.use(VueRouter)

// заводим массив с роутами
const routes = [
  {
    path: '/',
    name: 'Greeting',
    component: Greeting
  },

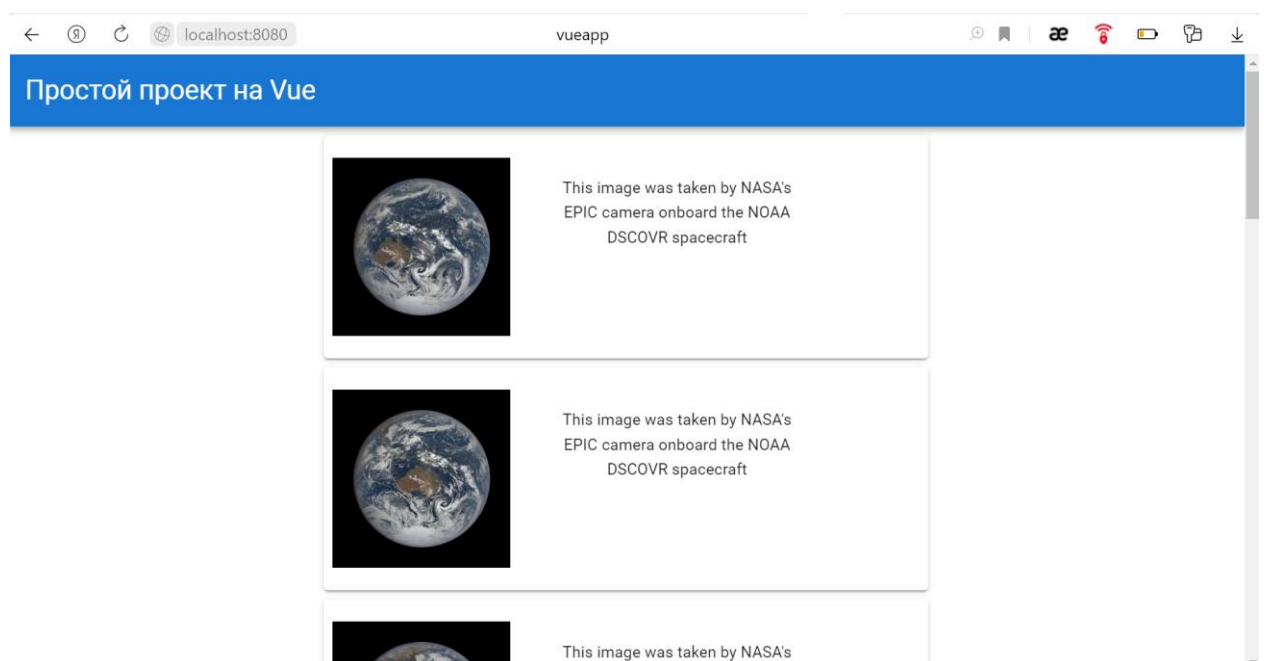
  {
    path: '/nasa',
    name: 'Nasa',
    component: Nasa
  }
]

// создаём новый экземпляр класса
// VueRouter, с необходимыми параметрами
const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

// экспортируем сконфигурированный роутер
export default router

```

Результат:



**Вывод:** по итогам данной работы мною был создан проект на Vue, и я ознакомился с базовыми командами прм.