

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №5

Выполнил:

Платонова Александра

Группа К3439

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

Задача

В рамках данной работы Вам предстоит изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue. Научиться работать с npm и vue.

Проект: Прикладное программное обеспечение деятельности отдела заселения муниципальных общежитий администрации города.

Ход работы

Первым делом необходимо установить node.js и npm на компьютер. Для установки была выполнена команда:

```
$ npm install -g @vue/cli
```

После выполнения этой команды на компьютере был установлен vue и vue-cli (инструмент, необходимый для того, чтобы облегчить работу с инициализацией проекта на vue).

Теперь, когда vue и vue-cli глобально установлены на компьютере нужно приступать к инициализации проекта:

```
$ vue create 03_vue_examples
```

После чего поступит предложение о выборе варианта установки:

```
Vue CLI v5.0.9
? Please pick a preset: (Use arrow keys)
❯ Default ([Vue 2] babel, eslint)
  Default (Vue 3 Preview) ([Vue 3] babel, eslint)
  Manually select features
```

```
Vue CLI v5.0.9
? Please pick a preset: Manually select features
? Check the features needed for your project:
❯◉ Choose Vue version
  ◉ Babel
  ◉ TypeScript
  ◉ Progressive Web App (PWA) Support
  ◉ Router
```

- Vuex
- CSS Pre-processors
- Linter / Formatter
- Unit Testing
- E2E Testing

```
Vue CLI v5.0.9
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, Router, Linter
? Choose a version of Vue.js that you want to start the project
with (Use arrow keys)
❯ 2.x
  3.x (Preview)
```

В роутере будет использоваться history mode.

```
Vue CLI v5.0.9
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, Router, Linter
? Choose a version of Vue.js that you want to start the project
with 2.x
? Use history mode for router? (Requires proper server setup for
index fallback in production) (Y/n)
```

В качестве линтера была выбрана стандартная конфигурация:

```
Vue CLI v5.0.9
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, Router, Linter
? Choose a version of Vue.js that you want to start the project
with 2.x
? Use history mode for router? (Requires proper server setup for
index fallback in production) Yes
? Pick a linter / formatter config:
  ESLint with error prevention only
  ESLint + Airbnb config
❯ ESLint + Standard config
  ESLint + Prettier
```

```
Vue CLI v5.0.9
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, Router, Linter
? Choose a version of Vue.js that you want to start the project
with 2.x
? Use history mode for router? (Requires proper server setup for
index fallback in production) Yes
? Pick a linter / formatter config: Standard
? Pick additional lint features: (Press <space> to select, <a> to
toggle all, <i> to invert selection)
➤● Lint on save
○ Lint and fix on commit
```

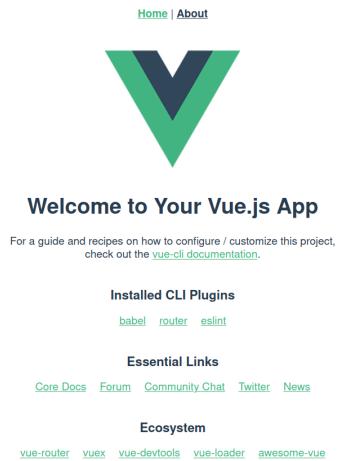
В следующем пункте уточняется необходимость разнесения всех конфигурационных файлов, такой вариант наиболее предпочтителен для обеспечения удобной возможности последующей настройки:

```
Vue CLI v5.0.9
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, Router, Linter
? Choose a version of Vue.js that you want to start the project
with 2.x
? Use history mode for router? (Requires proper server setup for
index fallback in production) Yes
? Pick a linter / formatter config: Standard
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? (Use
arrow keys)
➤ In dedicated config files
In package.json
```

Для запуска проекта используется следующая команда:

```
$ npm run serve
```

После запуска по адресу <http://localhost:8080>, в браузере появится следующая страница:



Далее следует работа и настройка роутингов. В файле конфигурации роутера (`src/router/index.js`) есть массив со всеми путями, которые используются в приложении:

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

import Home from '@/views/Home.vue'
import About from '@/views/About.vue'

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'Home',
    component: Home
  },
  {
    path: '/about',
    name: 'About',
    component: About
  }
]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

export default router
```

Для того чтобы продемонстрировать работы с представлениями и компонентами была создана компонента GreetingCard.vue:

```
<template>
  <h1>Привет!</h1>
</template>

<script>
export default {
  name: 'GreetingCard'
}
</script>

<style>
</style>
```

Весь код в данной компоненте логически можно разделить на три части: template, script, style. Template отвечает за вёрстку, html-код. Script – за все скрипты, которые используются для добавления какой-либо логики (запросы к API, вывод информации и так далее). Style – за все стили компоненты.

Теперь необходимо создать представление Greeting.vue:

```
<template>
<section>
  <greeting-card />

  <h2>Это простой проект на Vue.</h2>
</section>
</template>
<script>
import GreetingCard from '@/components/GreetingCard'
export default {
  name: 'Greeting',
  components: {
    GreetingCard
  }
}
</script>
```

```
<style>
</style>
```

Далее необходимо добавить представление в src/router/index.js:

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import Greeting from '@/views/Greeting.vue'

Vue.use(VueRouter)

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

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
} )

export default router
```

Теперь по адресу будет отображаться созданное представление <http://localhost:8080>:

Привет!
Это простой проект на Vue.

Далее был изменен главный компонент src/App.vue:

```
<template>
<main id="app">
  <router-view/>
</main>
</template>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
}

#nav {
  padding: 30px;
}

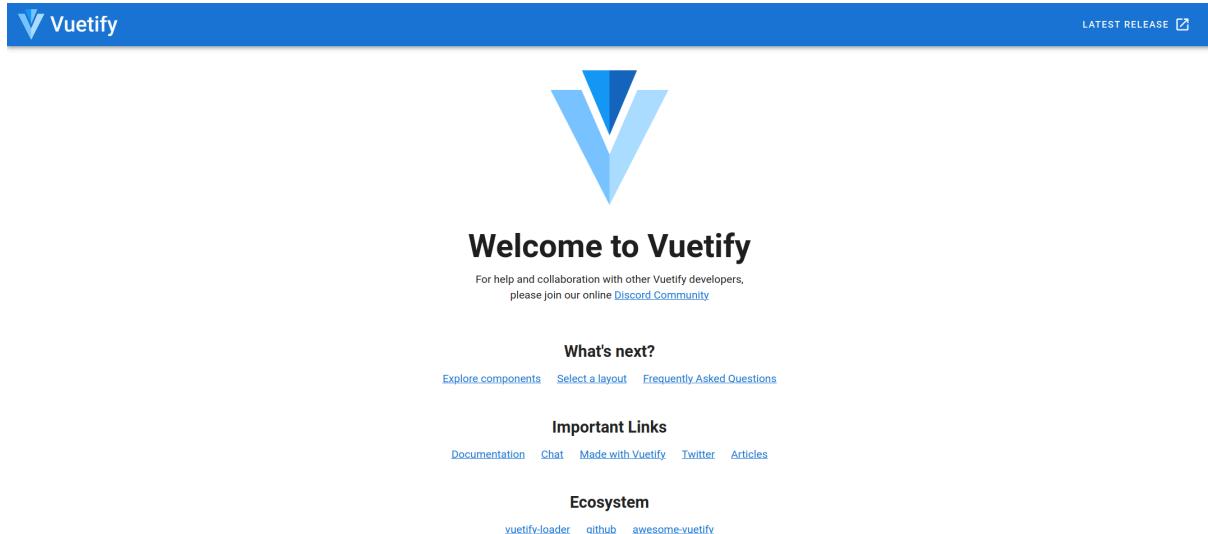
#nav a {
  font-weight: bold;
  color: #2c3e50;
}

#nav a.router-link-exact-active {
  color: #42b983;
}
</style>
```

Для ускорения и упрощения работы с Vue можно использовать фреймворк, для данной домашней работы был выбран фреймворк Vuetify, его можно установить через vue-cli следующей командой. Она автоматически добавит все необходимые файла для работы.

```
$ vue add vuetify
```

После успешной установки главная страница имеет следующий вид:



Далее было изменено содержимое главной компоненты:

```
<template>
  <v-app>
    <v-app-bar
      app
      color="primary"
      dark
    >
      <div class="d-flex align-center">
        <h1 class="headline">Простой проект на Vue</h1>
      </div>
    </v-app-bar>

    <v-main class="d-flex align-center text-center">
      <router-view />
    </v-main>
  </v-app>
</template>

<script>
export default {
  name: 'App',

```

```
data: () => ({
  //
})
}

</script>
```

Также было изменено содержимое представления Greeting:

```
<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 () {
      localStorage.setItem('username', this.username)

      this.savedUsername = this.username

      this.$refs.form.reset()
    }
  },
  created () {
    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>
```

В результате было получено следующее:

Простой проект на Vue

Привет, Иван!
Это простой проект на Vue.

Введите своё имя
Иван Иванов

Варианты приветствий:

- Привет
- Hello
- Hola

Для сетевого взаимодействия было выбрано использование библиотеки axios. Установить библиотеку можно следующей командой:

```
$ npm install axios vue-axios --save-dev
```

После чего зависимости были добавлены в src/main.js:

```
import Vue from 'vue'
```

```

import App from './App.vue'
import router from './router'
import vuetify from './plugins/vuetify'
import axios from 'axios'
import VueAxios from 'vue-axios'

Vue.config.productionTip = false
Vue.use(VueAxios, axios)

new Vue({
  router,
  vuetify,
  axios,
  render: h => h(App)
}).$mount('#app')

```

Для демонстрации работы было принято решение использовать открытый API NASA: <https://api.nasa.gov/>.

Для этого была создана компонента EpicItemCard:

```

<template>
  <v-card
    elevation="2"
    class="px-2 py-5">
    <v-row>
      <v-col cols="4">
        <v-img :src="epicItem.image" />
      </v-col>

      <v-col cols="6">
        <v-card-text>
          {{ epicItem.caption }}
        </v-card-text>
      </v-col>
    </v-row>
  </v-card>
</template>
<script>
export default {
  name: 'EpicItemCard',

```

```
props: {
  epicItem: Object
}
}

</script>
```

И представление Nasa:

```
<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 =
const apiUrl = 'https://api.nasa.gov/EPIC'

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

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

  methods: {
    async getEpicItems () {
      try {
        const response = await this.axios
```

```
.get(`${apiUrl}/api/natural?api_key=${apiKey}`)

    if (response.status !== 200) {
        throw new Error(response.error)
    }

    const epicItems = response.data.map((epicItem) => {
        const date = new Date(epicItem.date)

        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}`
        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>
```

Константы apiKey и apiUrl отвечают за ключ, который выдаётся после регистрации и адрес, на который будут посыпаться запросы. После успешного выполнения запроса необходимо обработать полученный результат, чтобы вывести данные в том виде, в котором их выводить удобно, а именно: получить адрес картинки, которая будет показываться в карточке.

Далее в src/router/index.js было добавлено представление:

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

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

Vue.use(VueRouter)

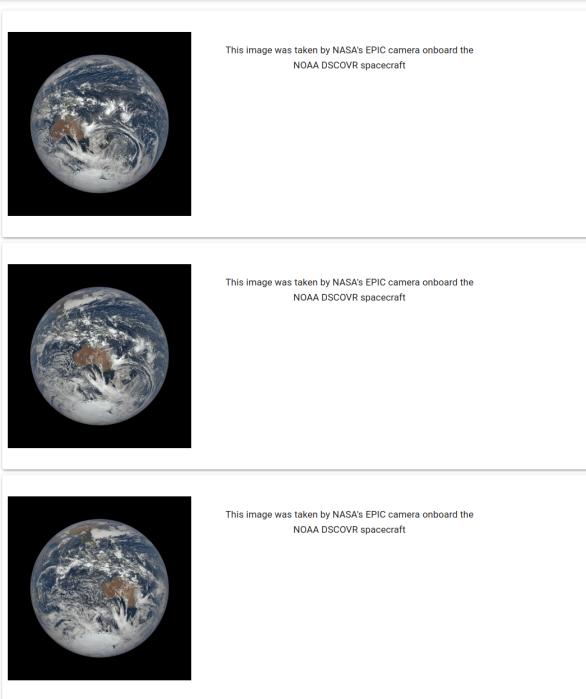
const routes = [
  {
    path: '/',
    name: 'Greeting',
    component: Greeting
  },
  {
    path: '/nasa',
    name: 'Nasa',
    component: Nasa
  }
]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

export default router
```

Результат:

Простой проект на Vue



Вывод

В ходе выполнения домашнего задания были получены навыки работы с npm, средством разработки [Vue.js](#), а также фреймворка Vuetify. Был изучен подход разработки, при котором происходит разделение кода на компоненты и представления. Также были получены навыки настройки роутинга средствами Vue.