# Занятие 10

# Фотолента. Загрузка

Продолжим реализацию нашего проекта.

## 10.1 Панель навигации

Создайте отдельную компоненту, содержащую навигационную панель, и добавьте её на страницу с лентой (снимок 10.1). При нажатии на ссылку «Фотолента» пользователь должен перенаправляться на главную страницу.

## 10.2 Форма

Реализуйте страницу загрузки фотографий и добавьте соответствующую ссылку в панель навигации (снимок 10.2). До выбора изображения кнопка «Загрузить!» должна быть недоступна. Используйте следующий пример[1].

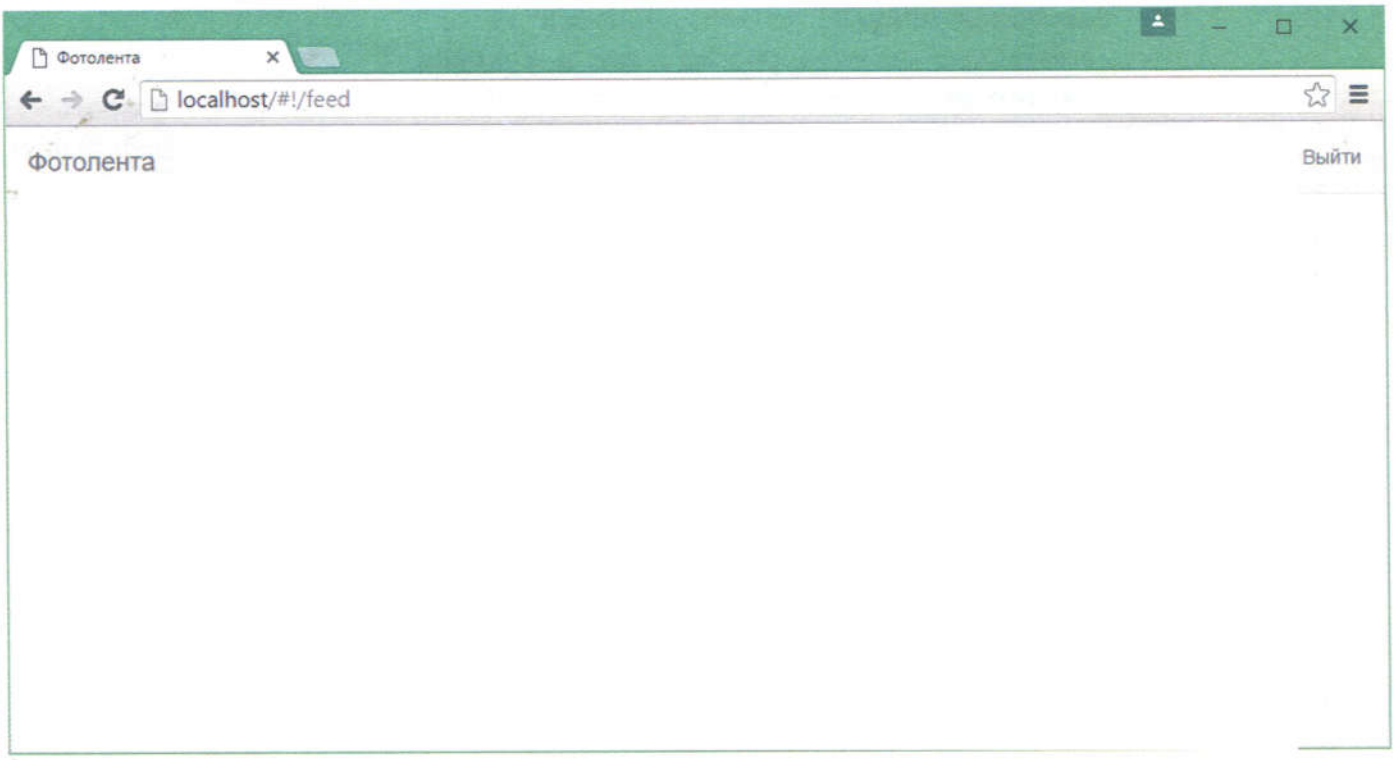
Листинг 10.1: `components/upload.vue`

```
<img v-bind:src="imageUrl">
<form v-on:submit="upload">
    <!-- событие change наступает при выборе изображения в диалоге -->
    <input type="file" accept="image/*" name="picture" v-on:change="preview">
    <button type="submit">Загрузить!</button>
</form>
```

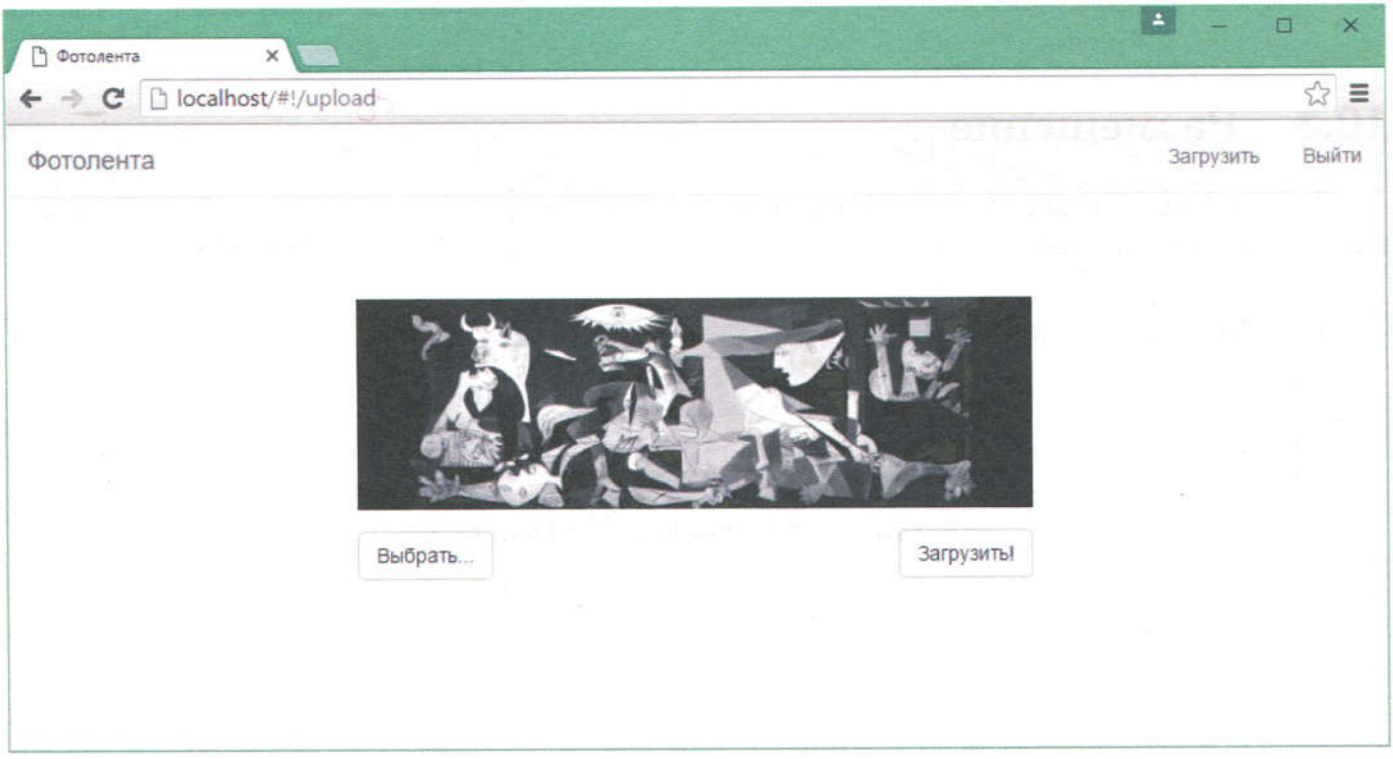
Листинг 10.2: `components/upload.vue`

```
methods: {
    preview: function(event) {
        /* массив выбранных файлов */
        var selectedFiles = event.target.files;
        var self, reader;
```

[1] В Листинге 10.2 *идиомы* с **self** можно избежать, если использовать *стрелочные функции* из стандарта ECMAScript 6. Подробнее о новых возможностях можно узнать по адресу `http://es6-features.org`.




Снимок 10.1: Панель навигации




Снимок 10.2: Загрузка

```
        if(selectedFiles.length == 1) {
            /* сохраняем текущий контекст вызова в
                отдельную переменную, так как в функции
                reader.onload контекстом будет reader */
            self = this;
            reader = new FileReader();
            reader.onload = function(readerEvent) {
                /* сохраняем полученный временный адрес картинки
                    в переменную imageUrl, вызывая обновление img*/
                self.imageUrl = readerEvent.target.result;
            };
            /* генерируем временный адрес файла */
            reader.readAsDataURL(selectedFiles[0]);
        }
    },
    upload: function(event) {
        /* предотвращаем перезагрузку страницы */
        event.preventDefault();
        var form = event.target;
        var formData = new FormData(form);
        /* параметр bearer добавляет
            зашифрованную подпись к запросу */
        this.$http.post("/upload", formData, { bearer: true })
            .then(...);
    }
}
```

## 10.3 Размещение

Для приёма формы с изображением на сервере нам понадобится модуль multer. Создайте папку для загрузки файлов и сохраните путь до неё в переменную **UPLOADS**.

Листинг 10.3: index.js

```
var multer = require("multer");
var upload = multer({ dest: UPLOADS });

app.post("/upload", [vjmServer.jwtProtector, upload.single("picture")],
    function(request, response) {
        response.sendStatus(200);
    }
);
```

Здесь мы использовали *промежуточные обработчики* **jwtProtector** и **single**:

1. **jwtProtector** проверяет подлинность подписи и сохраняет имя пользователя в переменную **request.user.username**;

2. **single** извлекает файл из поля формы с именем picture, помещает его в папку, заданную при помощи параметра **dest** и сохраняет имя файла в переменную **request.file.filename**.

Проверьте, что при загрузке фотографии в указанной папке действительно появляется новый файл. Обратите внимание, что у него отсутствует расширение. В контекстном меню нажмите Открыть с помощью и выберите Chrome из списка приложений. Фотография будет корректно отображена. Так происходит потому, что браузеры определяют тип файла по его содержимому, а не по расширению.

Модифицируйте функцию обратного вызова из Листинга 10.3 так, чтобы при каждой загрузке в базу данных помещался новый документ[2], содержащий следующие поля:

- имя пользователя **user**;
- имя файла **file**;
- текущий момент времени **date**.

Протестируйте добавленную функциональность.

[2]При наличии свободного времени освойте модуль mongoose, упрощающий реализацию операций с базами данных.