

# 27. Учебный проект: пришёл, увидел, загрузил (часть 2)

Рабочая ветка `module8-task2`

## Задание

### Создание и удаление данных

В этом задании мы до конца реализуем взаимодействие с сервером.

1. Опишите в API методы для создания (POST) и удаления (DELETE) задачи на сервере.
2. Измените код обновления данных так, чтобы при нажатии на кнопку «Save», сначала отправлялся POST-запрос на сервер и только в случае его успешного завершения на страницу добавлялась новая задача. Форма создания при этом должна скрываться.
3. Измените код обновления данных так, чтобы при нажатии на кнопку «Delete», сначала отправлялся DELETE-запрос на сервер, который бы удалял задачу, и только в случае успешного завершения со страницы скрывалась форма редактирования, а задача больше не появлялась в списке.

## Обратная связь

Доработаем интерфейс так, чтобы в нём появилась прозрачная обратная связь о состоянии всех запросов и ошибках.

При попытке пользователя сохранить новую задачу или удалить существующую реализуйте следующие шаги:

1.        Заблокируйте форму.
2.        Замените текст кнопки с «Save» на «Saving...» или с «Delete» на «Deleting...» в зависимости от действия.
3.        В случае успешного запроса к серверу форму нужно разблокировать, текст кнопок вернуться к исходному варианту, а также должны произойти все те манипуляции, что мы описали ранее.
4.        Если в процессе запроса произошла ошибка:
  - разблокируйте форму;
  - верните текст кнопок к исходному варианту;
  - добавьте всей форме эффект «покачивания головой» (CSS-код для эффекта вы найдёте ниже, а

можете реализовать самостоятельно);

- а также добавьте всему блоку с формой редактирования красную обводку (стилизация на ваше усмотрение).

5. Перед повторной отправкой убирайте с формы красную обводку.

Эффект покачивания головой:

```
<style>
  @keyframes shake {
    0%,
    100% {
      transform: translateX(0);
    }

    10%,
    30%,
    50%,
    70%,
    90% {
      transform: translateX(-5px);
    }

    20%,
    40%,
    60%,
    80% {
      transform: translateX(5px);
    }
  }

```

```
}  
.shake {  
  animation: shake 0.6s;  
}  
</style>
```

## 29. Личный проект: пришёл, увидел, загрузил (часть 2)

### Задание

В этом задании мы до конца реализуем взаимодействие с сервером.

## Создание и удаление данных

1. Опишите в API методы для создания (POST) и удаления (DELETE) комментария на сервере.
2. Измените код обновления данных так, чтобы при отправке формы, сначала отправлялся POST-запрос на сервер и только в случае его успешного завершения в попап добавлялся новый комментарий. Форма при этом должна очиститься, в том числе выбранная эмоция.

**Напоминаем**, что отправка формы осуществляется нажатием комбинации клавиш `Ctrl + Enter` (Window, Linux) и `Command + Enter` (macOS).

3. Измените код обновления данных так, чтобы при нажатии на кнопку «Delete», сначала отправлялся DELETE-запрос на сервер, который бы удалял комментарий, и только в случае успешного завершения комментарий скрывался в попапе.

## Обратная связь

Доработаем интерфейс так, чтобы в нём появилась прозрачная обратная связь о состоянии всех запросов и ошибках.

При отправке нового комментария, реализуйте следующие шаги:

1. Заблокируйте форму.
2. В случае успешного запроса к серверу должны произойти все те манипуляции, что мы описали ранее.
3. Если в процессе запроса произошла ошибка:
  - разблокируйте форму;
  - добавьте всей форме эффект «покачивания головой» (CSS-код для эффекта вы найдёте ниже, а можете реализовать самостоятельно);
  - а полю ввода добавьте красную обводку (стилизация на ваше усмотрение).
4. Перед повторной отправкой убирайте с красную обводку.

При попытке удалить комментарий:

1. Заблокируйте кнопку и замените её текст с «Delete» на «Deleting...».

2. В случае успешного запроса к серверу должны произойти все те манипуляции, что мы описали ранее.
3. Если в процессе запроса произошла ошибка:
  - разблокируйте кнопку;
  - добавьте блоку с комментарием эффект «покачивания головой» (CSS-код для эффекта вы найдёте ниже, а можете реализовать самостоятельно).

Эффект покачивания головой

```
<style>
```

```
  @keyframes shake {
```

```
    0%,
```

```
    100% {
```

```
      transform: translateX(0);
```

```
    }
```

```
    10%,
```

```
    30%,
```

```
    50%,
```

```
    70%,
```

```
    90% {
```

```
      transform: translateX(-5px);
```

```
    }
```

```
    20%,
```

```
    40%,
```



```
    60%,
    80% {
        transform: translateX(5px);
    }
}
.shake {
    animation: shake 0.6s;
}
</style>
```

## Создание и удаление данных

1. Опишите в API методы для создания (POST) и удаления (DELETE) точки маршрута на сервере.
2. Измените код обновления данных так, чтобы при нажатии на кнопку «Save», сначала отправлялся POST-запрос на сервер и только в случае его успешного завершения на страницу добавлялась новая точка маршрута. Форма создания при этом должна скрываться.
3. Измените код обновления данных так, чтобы при нажатии на кнопку «Delete», сначала отправлялся DELETE-запрос на сервер, который бы удалял точку маршрута, и только в случае успешного завершения со страницы скрывалась форма редактирования, а точка маршрута больше не появлялась в списке.

## Обратная связь

Доработаем интерфейс так, чтобы в нём появилась прозрачная обратная связь о состоянии всех запросов и ошибках.

При попытке пользователя сохранить новую точку маршрута или удалить существующую реализуйте следующие шаги:

1. Заблокируйте форму.

2.        Замените текст кнопки с «Save» на «Saving...» или с «Delete» на «Deleting...» в зависимости от действия.
  
3.        В случае успешного запроса к серверу форму нужно разблокировать, текст кнопок вернуться к исходному варианту, а также должны произойти все те манипуляции, что мы описали ранее.
  
4.        Если в процессе запроса произошла ошибка:
  - разблокируйте форму;
  
  - верните текст кнопок к исходному варианту;
  
  - добавьте всей форме эффект «покачивания головой» (CSS-код для эффекта вы найдёте ниже, а можете реализовать самостоятельно);
  
  - а также добавьте всему блоку с формой редактирования красную обводку (стилизация на ваше усмотрение).
  
5.        Перед повторной отправкой убирайте с формы красную обводку.

Эффект покачивания головой

```
<style>
  @keyframes shake {
    0%,
    100% {
      transform: translateX(0);
    }

    10%,
    30%,
    50%,
    70%,
    90% {
      transform: translateX(-5px);
    }

    20%,
    40%,
    60%,
    80% {
      transform: translateX(5px);
    }
  }
  .shake {
    animation: shake 0.6s;
  }
</style>
```