



Szkolenie Rails

Projekt TODO list: naszym zadaniem będzie implementacja AJAXowej komunikacji z serwerem w celu aktualizacji widoków bez przeładowania strony.

Zadanie 1

Pobierz aplikację TODO list z repozytorium.

<https://bitbucket.org/jbuttonrorpl/tasks>

Utwórz bazę danych, uruchom migrację, wygeneruj seedowe dane.

Zapoznaj się, przetestuj aplikację.

Zadanie 2

Zacznijmy od tworzenia nowych tasków:

do linka "New Task" dodaj parametr remote: true

Co utworzy tag unobtrusive JavaScript data-remote=true, który powoduje, że ten link zostanie wysłany przez AJAX.

W przeglądarce Chrome zobacz co się dzieje pod maską po kliknięciu odświeżonego linku.

Zadanie 3

Utwórzmy nowy JavaScriptowy widok new.js.erb, który zostanie wywołany po kliknięciu na link.

Akcję po kliknięciu linka będzie ukrycie go i wyświetlenie formularza:

```
$('#new_link').hide().after('<%= j render("form") %>');
```

Zadanie 4

Jak zauważyłeś po kliknięciu Create Task nastąpiło przeładowanie strony, aby to zmienić następnym krokiem będzie dodanie parametru remote: true do formularza.

Teraz musimy dodać obsługę zapytania w kontrolerze w akcji create:

```
respond_to do |format|  
  format.html { redirect_to tasks_url }  
  format.js
```

end

Utwórzmy template create.js.erb który będzie wyświetlony po utworzeniu nowego taska.

Nowy template będzie zawierał kilka akcji:

- ☐ Usuniemy formularz z widoku.
- ☐ Pokażemy z powrotem link tworzenia nowego taska.
- ☐ Wstawiamy nowo utworzony task do listy.

```
$('#new_task').remove();
$('#new_link').show();
$('#incomplete_tasks').append('<%= j render(@task) %>');
```

Zadanie 5

Usuwanie tasków możemy rozwiązać identycznie tak jak ich tworzenie. Spróbuj to zaimplementować samodzielnie.

Zadanie 6

Zaznaczanie tasków jako wykonanych:

zmień rozszerzenie pliku assets/javascripts/tasks.js.coffee na task.js
aby można było pisać w czystym JavaScript

```
$(function() {
  $('#edit_task input[type=submit]').remove();
  $('#edit_task input[type=checkbox]').click(function() {
    $(this).parent('form').submit();
  });
});
```

Teraz wystarczy zmienić formularz na AJAX'owy
poprzez dodanie parametru remote true oraz dodaniu obsługi JavaScriptu w kontrolerze:

update:

```
respond_to do |format|
  format.html { redirect_to tasks_url }
  format.js
end
```

i utworzeniu nowego template'u update.js.erb

```
<% if @task.complete? %>
$('#edit_task_<%= @task.id %>').appendTo('#complete_tasks');
<% else %>
$('#edit_task_<%= @task.id %>').appendTo('#incomplete_tasks');
```

<% end %>

Zadanie 7

Ostatnim krokiem będzie naprawienie bugu tworzenia nowych tasków:

tasks.js:

```
jQuery.fn.submitOnCheck = function() {  
    this.find('input[type=submit]').remove();  
    this.find('input[type=checkbox]').click(function() {  
        $(this).parent('form').submit();  
    });  
    return this;  
}
```

```
$(function() {  
    $('#edit_task').submitOnCheck();  
});
```

i zaaplikuj nową funkcjonalność w szablonie create.js.ebr

```
$('#edit_task_<%= @task.id %>').submitOnCheck();
```