# Rails: Week 5 – AJAX

**https://github.com/revialim/m1-rails-app**

I added a new resource to my project. It is called 'toest' and has following attributes **name**, which is a string and name of the toest, **crunchiness**, which is an integer and says how crunchy the toest is, **sweet** which is boolean and indicates whether the toest is sweet or not, and last but not least **price**, which tells how expensive the toest is.

For this resources index view I wanted to add a dynamic list of all toests, that updates itself, when a new toest is created.

When the user enters the view of an individual toest, they can also "eat" the toest, which will then be deleted and the user redirected to the index view.

## Adding the eat button

To add the button for eating the toest, I added the line

```
<%= link_to "Eat toest, it will be gone", @toest, remote: true, method: :delete %>
```

to the toest's show.html.erb and

```
$ ->
  $("a[data-remote]").on "ajax:success", (e, data, status, xhr) ->
     alert "The toest was eaten."
```

to the toests.coffee file.

## Adding the dynamic list

To add the automatically updated list view to the index page, first I created the file "_toest.html.erb", to which I added the line

```
<li><%= toest.name %></li>
```

To the toest's index.html.erb I added

```
<ul id="toests">
<%= render @toests %>
```

```
</ul>
```

which will render the lists, described in _toest.html.erb. Furthermore I added a form to add a new toest.

```
<%= form_for(@toest, remote: true) do |f| %>
  <%= f.label :name %><br>
  <%= f.text_field :name %>
  <%= f.submit %>
<% end %>
```

For `@toest` to be usable, I added an assignment to the toests_controller.rb

```
def index
  @toests = Toest.all
  @toest = Toest.new
end
```

Then I added the `format.js` to the controller's create action, so that a JavaScript request can be served.

```
def create
    @toest = Toest.new(toest_params)

    respond_to do |format|
      if @toest.save
        format.html { redirect_to @toest, notice: 'Toest was
successfully created.' }
        format.js   {}
        format.json { render :show, status: :created,
location: @toest }
      else
        format.html { render :new }
        format.json { render json: @toest.errors,
status: :unprocessable_entity }
      end
    end
  end
```

I created the create.js.erb file with the following line

```
$("<%= escape_javascript(render @toest)
%>").appendTo("#toests");
```

which is the actual JavaScript that will append the newly created toest to the list.