

In this lecture, we will discuss...

- ✧ index RESTful action

Examining Seven Actions – Index

1. Retrieve all posts
2. (Implicit) Look for `index.html.erb` template to render response

```
class PostsController < ApplicationController  
  
  # GET /posts  
  # GET /posts.json  
  def index  
    @posts = Post.all  
  end  
end
```



index.html.erb

Browsers only support GET and POST methods - how do we tell Rails to treat request as DELETE?



```
14 <tbody>
15   <%= @posts.each do |post| %>
16     <tr>
17       <td><%= post.title %></td>
18       <td><%= post.content %></td>
19       <td><%= link_to 'Show', post %></td>
20       <td><%= link_to 'Edit', edit_post_path(post) %></td>
21       <td><%= link_to 'Destroy', post, method: :delete, data: { confirm: 'Are you sure?' } %></td>
22     </tr>
23   <%= end %>
24 </tbody>
```

`post = post_path(post)`

index.html.erb

```
<tr>
  <td>Welcome</td>
  <td>Happy Action Packing!</td>
  <td><a href="/posts/1">Show</a></td>
  <td><a href="/posts/1/edit">Edit</a></td>
  <td><a data-confirm="Are you sure?" rel="nofollow" data-method="delete" href="/posts/1">Destroy</a></td>
</tr>
```

HTML5 data attributes...



index.json.jbuilder

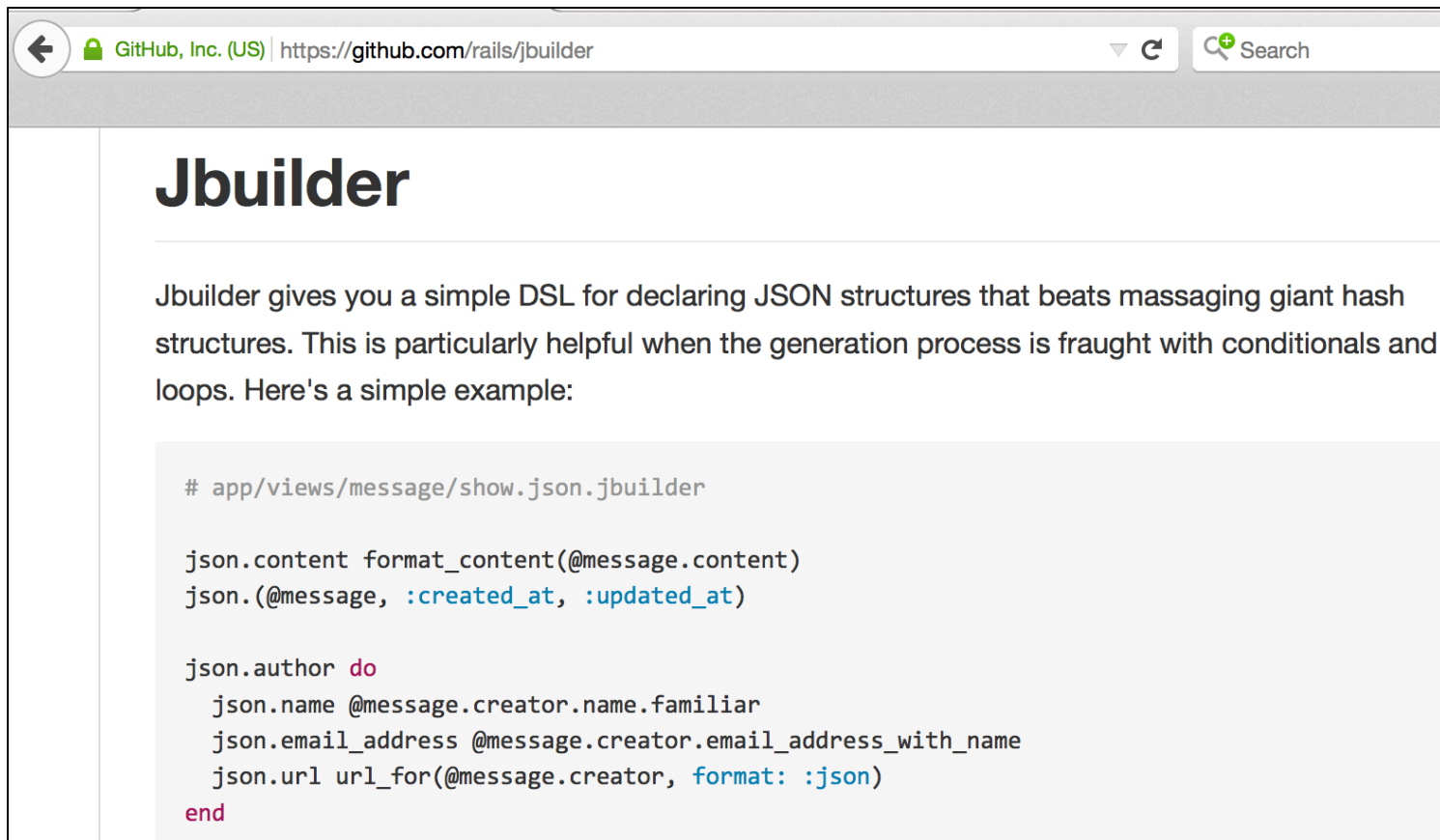
FOLDERS

- ▼ my_blog
 - ▼ app
 - ▶ assets
 - ▶ controllers
 - ▶ helpers
 - ▶ mailers
 - ▶ models
 - ▼ views
 - ▶ layouts
 - ▼ posts
 - _form.html.erb
 - edit.html.erb
 - index.html.erb
 - index.json.jbuilder

index.json.jbuilder *

```
1 json.array!(@posts) do |post|
2   json.extract! post, :id, :title, :content
3   json.url post_url(post, format: :json)
4 end
5
6
7
```

Jbuilder

A screenshot of a web browser displaying the GitHub page for Jbuilder. The browser's address bar shows the URL 'https://github.com/rails/jbuilder'. The page features a large heading 'Jbuilder' followed by a descriptive paragraph and a code example. The code is presented in a light gray box with syntax highlighting.

← GitHub, Inc. (US) | https://github.com/rails/jbuilder 🔍 Search

Jbuilder

Jbuilder gives you a simple DSL for declaring JSON structures that beats massaging giant hash structures. This is particularly helpful when the generation process is fraught with conditionals and loops. Here's a simple example:

```
# app/views/message/show.json.jbuilder

json.content format_content(@message.content)
json.(@message, :created_at, :updated_at)

json.author do
  json.name @message.creator.name.familiar
  json.email_address @message.creator.email_address_with_name
  json.url url_for(@message.creator, format: :json)
end
```



index.json.jbuilder



Use JSONView browser plugin

Summary

- ✧ index action retrieves resources from Data layer
- ✧ Then, implicitly invokes either HTML or JSON templates

What's Next?

- ✧ show and destroy RESTful actions

