



# Работа с JS и CSS в Rails

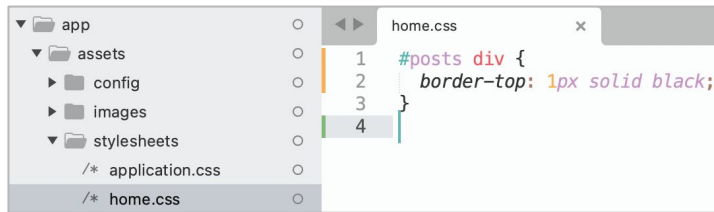
asset pipeline, webpacker, import  
maps



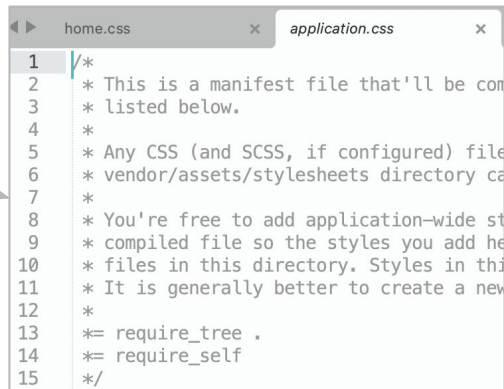
# Простой способ написать JS и CSS

```
1 <script type='text/javascript'>
2   let postsCount = 0;
3
4   document.addEventListener('DOMContentLoaded', (event) => {
5     const posts = document.getElementById('posts');
6     if (!posts) return;
7
8     postsCount = posts.getElementsByTagName('div').length
9
10    alert(`Number of posts ${postsCount}`);
11  });
12
13  const getPostsCount = (event) => {
14    event.target.textContent += ` ${postsCount}`;
15  }
16 </script>
17
18 <style>
19   #posts div {
20     border-top: 1px solid black;
21   }
22 </style>
23
24 <p style="color: green"><%= notice %></p>
25
26 <h1 onclick='getPostsCount(event) '>Posts</h1>
27
28 <div id='posts'>
29   <% @posts.each do |post| %>
```

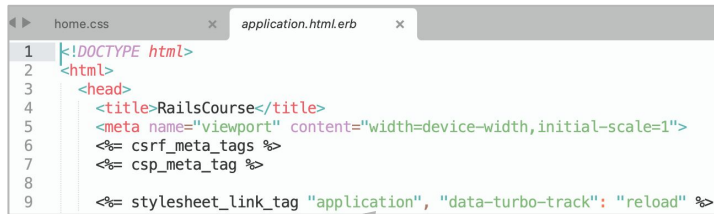
# Использование Asset Pipeline для CSS



```
home.css
1 #posts div {
2   border-top: 1px solid black;
3 }
4
```



```
home.css application.css
1 /*
2  * This is a manifest file that'll be compiled into application.css, which will include all the CSS
3  * files listed below.
4  *
5  * Any CSS (and SCSS, if configured) file in this directory, as well as any CSS files in
6  * vendor/assets/stylesheets, will be included into this file.
7  *
8  * You're free to add application-wide styles to this file, and it will be compiled into the
9  * compiled file so the styles you add here will be available for all views.
10 *
11 * It is generally better to create a new file per style scope to better organize the
12 * application's styles.
13 *
14 * require_tree .
15 * require_self
```



```
home.css application.html.erb
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>RailsCourse</title>
5   <meta name="viewport" content="width=device-width,initial-scale=1">
6   <%= csrf_meta_tags %>
7   <%= csp_meta_tag %>
8
9   <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
```

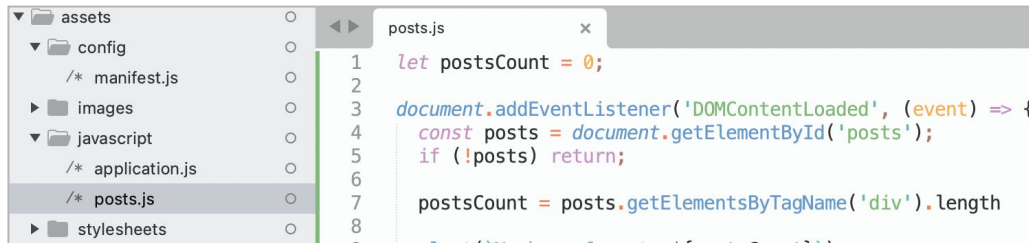


```
ERS
rails_course
  app
    assets
      config
        /* manifest.js
1 // = link_tree ../images
2 // = link_directory ../stylesheets .css
3
```

Для компиляции изображений, файлов CSS и JS используется команда *rails assets:precompile*, которая используется преимущественно для *production* окружения.

```
RAILS_ENV=production rails assets:precompile
30 #7009] INFO -- : Writing /Users/alex/study/rails_course/public/assets/manifest
61 #7009] INFO -- : Writing /Users/alex/study/rails_course/public/assets/manifest
81 #7009] INFO -- : Writing /Users/alex/study/rails_course/public/assets/posts-ab
38 #7009] INFO -- : Writing /Users/alex/study/rails_course/public/assets/applicat
68 #7009] INFO -- : Writing /Users/alex/study/rails_course/public/assets/applicat
```

# Использование Asset Pipeline для JS

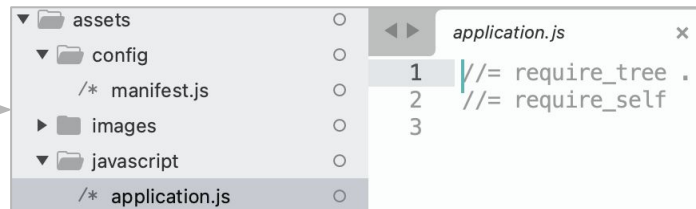


The screenshot shows a file explorer on the left with the following structure:

- assets
  - config
    - /\* manifest.js
  - images
  - javascript
    - /\* application.js
    - /\* posts.js
  - stylesheets

The main editor displays the file `posts.js` with the following code:

```
1 let postsCount = 0;  
2  
3 document.addEventListener('DOMContentLoaded', (event) => {  
4   const posts = document.getElementById('posts');  
5   if (!posts) return;  
6  
7   postsCount = posts.getElementsByTagName('div').length  
8
```



The screenshot shows a file explorer on the left with the following structure:

- assets
  - config
    - /\* manifest.js
  - images
  - javascript
    - /\* application.js

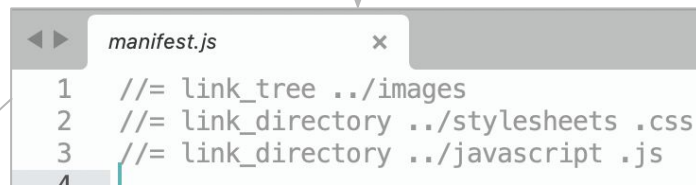
The main editor displays the file `application.js` with the following code:

```
1 // = require_tree .  
2 // = require_self  
3
```



The screenshot shows the file `application.html.erb` with the following code:

```
1 <!DOCTYPE html>  
2 <html>  
3   <head>  
4     <title>RailsCourse</title>  
5     <meta name="viewport" content="width=device-width,initial-scale=1">  
6     <%= csrf_meta_tags %>  
7     <%= csp_meta_tag %>  
8  
9     <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>  
10    <%= javascript_include_tag "application", "data-turbo-track": "reload" %>
```



The screenshot shows the file `manifest.js` with the following code:

```
1 // = link_tree ../images  
2 // = link_directory ../stylesheets .css  
3 // = link_directory ../javascript .js  
4
```

Данный способ активно использовался в Ruby on Rails до версии 5.1, пока не была добавлена возможность использовать сборщики JavaScripts кода.

В текущий момент лучше использовать более современные подходы, доступные в Rails.

# Работа с JS с помощью Webpack

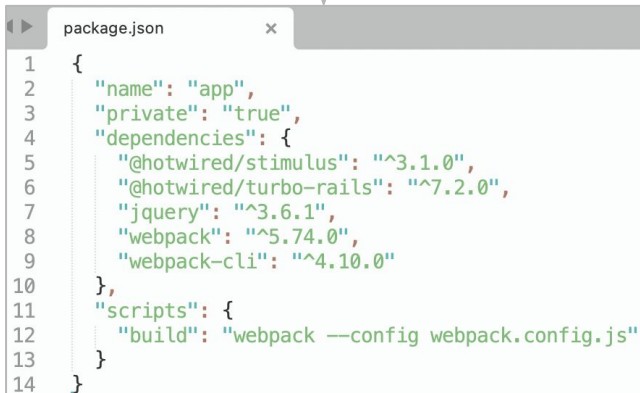
```
→ study rails new rails_course_webpacker --javascript=webpack  
create  
create README.md
```



```
application.js  
1 // Entry point for the build s  
2 import "@hotwired/turbo-rails"  
3 import "./controllers"
```

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 <title>RailsCourseWebpacker</title>  
5 <meta name="viewport" content="width=device-width,initial-scale=1">  
6 <%= csrf_meta_tags %>  
7 <%= csp_meta_tag %>  
8  
9 <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>  
10 <%= javascript_include_tag "application", "data-turbo-track": "reload", defer: true %>
```

```
→ rails_course_webpacker git:(main) x yarn add jquery  
yarn add v1.22.11  
[1/4] 🔍 Resolving packages...  
[2/4] 📦 Fetching packages...  
[3/4] 🔗 Linking dependencies...  
[4/4] 🔗 Building fresh packages...  
success Saved lockfile.  
success Saved 1 new dependency.
```



```
package.json  
1 {  
2   "name": "app",  
3   "private": "true",  
4   "dependencies": {  
5     "@hotwired/stimulus": "^3.1.0",  
6     "@hotwired/turbo-rails": "^7.2.0",  
7     "jquery": "^3.6.1",  
8     "webpack": "^5.74.0",  
9     "webpack-cli": "^4.10.0"  
10  },  
11  "scripts": {  
12    "build": "webpack --config webpack.config.js"  
13  }  
14 }
```

Подобный способ работы с JavaScript кодом стоит выбрать, если frontend часть приложения использует фреймворк по типу React. И при этом включены шаги транспиляции с JSX или TypeScript.

# Работа с ImportMaps в Rails 7.0

```
application.js
1 // Configure your import map in config/importmap.rb. Read more: https://github.com/rails/importmap-rails
2 import "@hotwired/turbo-rails"
3 import "controllers"
```

→ rails\_course git:(main) x ./bin/importmap pin jquery  
Pinning "jquery" to https://ga.jspm.io/npm:jquery@3.6.1/dist/jquery.js

```
application.js
1 // Configure your import map i
2 import "@hotwired/turbo-rails"
3 import "controllers"
4
5 _import $ from 'jquery'
```

```
application.html.erb
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>RailsCourse</title>
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <%= csrf_meta_tags %>
7     <%= csp_meta_tag %>
8
9     <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
10    <%= javascript_importmap_tags %>
```

```
application.js
importmap.rb
1 # Pin npm packages by running ./bin/importmap
2
3 pin "application", preload: true
4 pin "@hotwired/turbo-rails", to: "turbo.min.js", preload: true
5 pin "@hotwired/stimulus", to: "stimulus.min.js", preload: true
6 pin "@hotwired/stimulus-loading", to: "stimulus-loading.js", preload: true
7 pin_all_from "app/javascript/controllers", under: "controllers"
8 pin "jquery", to: "https://ga.jspm.io/npm:jquery@3.6.1/dist/jquery.js"
```

# Создание JS кода с jQuery и ImportMaps

```
application.js x
1 // Configure your import map in config/importmap.json
2
3 import $ from 'jquery';
4
5 let postsCount = 0;
6
7 $(document).ready(() => {
8   const posts = $('#posts');
9   if (!posts) return;
10
11   postsCount = posts.children('div').length;
12   alert(`Number of posts: ${postsCount}`);
13 })
14
15 $('#posts_title').click((event) => {
16   event.target.textContent += ` ${postsCount}`;
17 });
```

# Полезные ссылки

<https://rusrails.ru/asset-pipeline> – Описание Asset Pipeline;

<https://www.honeybadger.io/blog/react-rails/> – Пример работы с React фреймворком с помощью Webpacker гема;

<https://blog.appsignal.com/2022/03/02/import-maps-under-the-hood-in-rails-7.html> –  
Настройка Import Maps в приложении;

<https://www.bootrails.com/blog/rails-7-hotwire-a-tutorial/> – Разбор технологии Hotwire.



Конец! Спасибо!