

Adding client-side JavaScript

Add your own client-side JavaScript

- Create a file in the public/ directory
- Add your client-side JavaScript to this file

+ New File

 assets

controllers/about.js

controllers/dashboard.js

controllers/playlist.js

controllers/start.js

models/playlist-store.js

models/playlist-store.json

public/stylesheets/style.css

public/client.js

utils/logger.js

views/layouts/main.hbs

views/partials/listplaylists.hbs

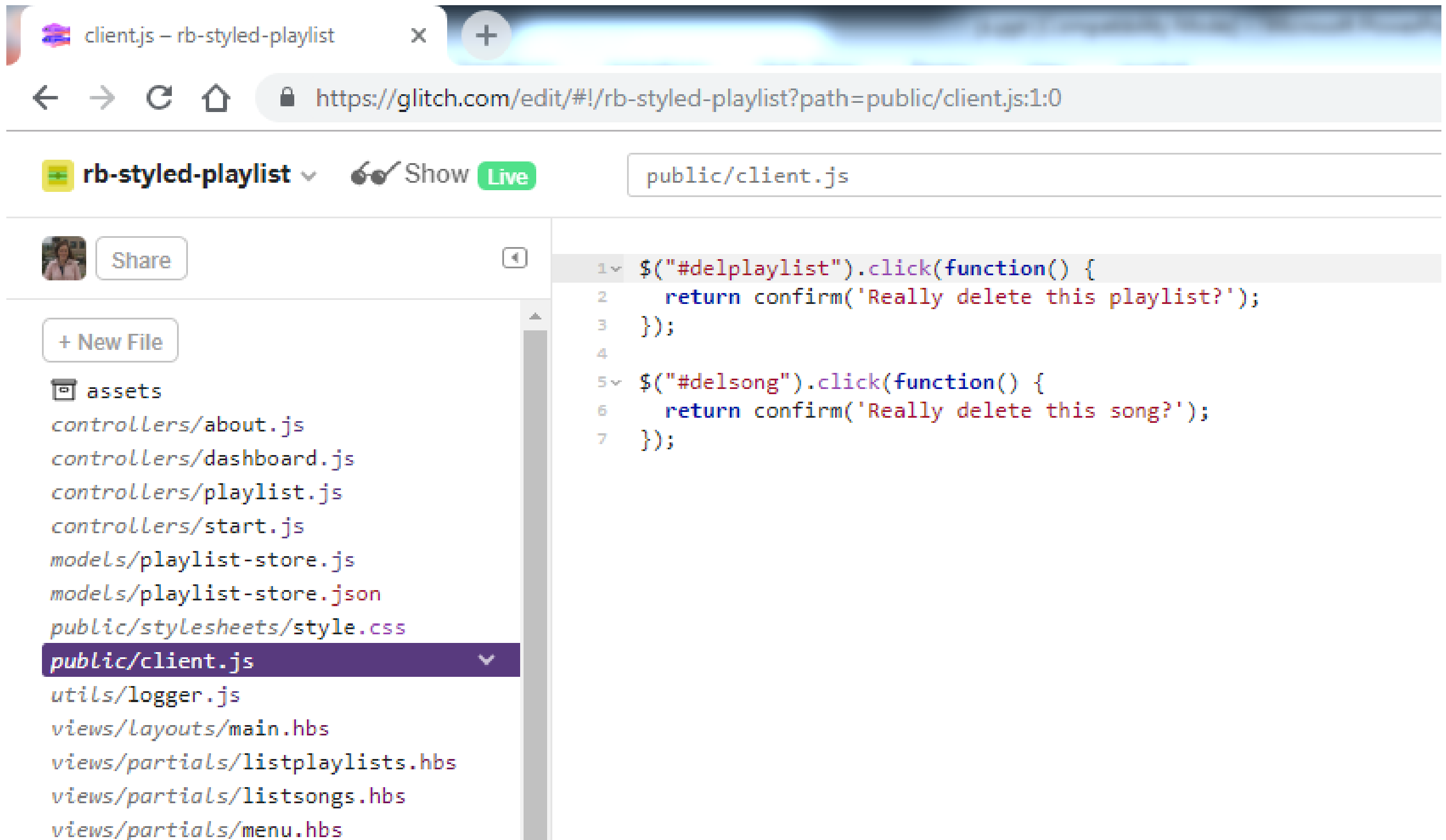
views/partials/listsongs.hbs

views/partials/menu.hbs

views/about.hbs

... ..

Add your own client-side JavaScript



The screenshot shows a web browser window with a single tab titled "client.js - rb-styled-playlist". The address bar displays the URL "https://glitch.com/edit/#!/rb-styled-playlist?path=public/client.js:1:0". Below the browser window, the Glitch editor interface is visible. The top bar shows the project name "rb-styled-playlist" with a dropdown arrow, a "Show" button with a live icon, and a "Live" button. The file path "public/client.js" is shown in a text box. On the left side, there is a "Share" button and a "+ New File" button. Below these, a file explorer lists the project's structure: "assets", "controllers/about.js", "controllers/dashboard.js", "controllers/playlist.js", "controllers/start.js", "models/playlist-store.js", "models/playlist-store.json", "public/stylesheet/style.css", "public/client.js" (which is selected and highlighted in dark purple), "utils/logger.js", "views/layouts/main.hbs", "views/partials/listplaylists.hbs", "views/partials/listsongs.hbs", and "views/partials/menu.hbs". The main editor area on the right displays the JavaScript code for the selected file, with line numbers 1 through 7 on the left. The code is as follows:

```
1  $("#delplaylist").click(function() {  
2      return confirm('Really delete this playlist?');  
3  });  
4  
5  $("#delsong").click(function() {  
6      return confirm('Really delete this song?');  
7  });
```

Add your own client-side JavaScript

- Link to /client.js in layouts/main.hbs file

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title> {{title}} </title>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.css" type="text/css">
    <link rel="stylesheet" type="text/css" href="/stylesheets/style.css">
  </head>
  <body>
    <section class="ui container">
      {{{body}}}
    </section>
    <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.js"></script>
    <script type="text/javascript" src="/client.js"></script>
  </body>
</html>
```

Add your own client-side JavaScript/jQuery

- This example:
 - Confirms that the user does want to delete the playlist/song

```
1 ✓ $( "#delplaylist" ).click(function() {  
2     return confirm('Really delete this playlist?');  
3 });  
4  
5 ✓ $( "#delsong" ).click(function() {  
6     return confirm('Really delete this song?');  
7 });
```

JavaScript/jQuery references

- <https://www.w3schools.com/jsref/default.asp>
- <https://developer.mozilla.org/bm/docs/Web/JavaScript/Reference>
- https://www.w3schools.com/jquery/jquery_get_started.asp

Link to styled-playlist repository

- Import from GitHub: rbirney/styled-playlist

The screenshot shows the Glitch.com editor interface for a project named 'rb-styled-playlist'. The browser address bar displays the URL: `https://glitch.com/edit/#!/rb-styled-playlist?path=views/layouts/main.hbs:18:3`. The left sidebar shows a file explorer with a tree structure including `assets`, `controllers`, `models`, `public`, `utils`, and `views`. The main editor area displays the content of `views/layouts/main.hbs`, which is an HTML template with a head section containing meta tags and links, and a body section containing a script tag for `client.js`.

A modal dialog titled 'glitch.com says' is open, providing instructions for importing from an existing GitHub repository. The modal text reads: 'Import from an existing GitHub repository - please note that this will overwrite your current project. Enter the GitHub username, repository name, and optionally a relative path to a specific folder to import using the following format: username/repository:path.' Below the text is a text input field containing the text 'rbirney/styled-playlist'. The modal has 'OK' and 'Cancel' buttons.

In the bottom left, the 'Git, Import, Export' menu is open, showing options for 'Read' and 'Write' access, a URL input field with the text 'https://api.glitch.com/rb-s', and buttons for 'Import from GitHub' and 'Export to GitHub'.