# jsDelivr and GitHub Pages as CDN for Angular Assets

Contents

[jsDelivr and GitHub Pages as CDN for Angular Assets 1](#_Toc176858158)

[Introduction 2](#_Toc176858159)

[Option 1: jsDelivr 2](#_Toc176858160)

[Step 1: Prepare Your GitHub Repository 2](#_Toc176858161)

[Step 2: Organize Your Assets 2](#_Toc176858162)

[Step 3: Push Your Assets to GitHub 2](#_Toc176858163)

[Step 4: Get jsDelivr URLs 2](#_Toc176858164)

[Step 5: Update Your Angular Project 3](#_Toc176858165)

[Option 2: GitHub Pages 3](#_Toc176858166)

[Step 1: Follow the step 1-3 same as above 3](#_Toc176858167)

[Step 2: Now click on setting icon and click in pages 3](#_Toc176858168)

[Step 3: Update Asset References 4](#_Toc176858169)

[Conclusion 4](#_Toc176858170)

## Introduction

This guide provides detailed, step-by-step instructions for using two free CDN options with our Angular project: jsDelivr and GitHub Pages. Both options leverage GitHub repository to serve our assets, but they differ in their setup and use cases.

## Option 1: jsDelivr

<https://www.jsdelivr.com/?docs=gh>

https://www.jsdelivr.com/github

jsDelivr is a free, open-source CDN that works directly with GitHub repositories. It’s easy to set up and doesn’t require any account creation beyond your existing GitHub account.

### Step 1: Prepare Your GitHub Repository

1. Create a new public repository for your project:

### Step 2: Organize Your Assets

1. In your local project, create an assets:

* mkdir src/assets

1. Organize your assets (images, fonts, etc.) within this folder. For example:

* src/assets/  
  ├── images/  
  │ ├── logo.png  
  │ └── background.jpg  
  ├── fonts/  
  │ └── custom-font.woff2  
  └── data/  
   └── config.json

### Step 3: Push Your Assets to GitHub

1. Push your code to GitHub

### Step 4: Get jsDelivr URLs

jsDelivr uses a specific URL structure to serve files from your GitHub repository:

https://cdn.jsdelivr.net/gh/username/repo@version/file

For example, to serve the flash anime.jpg file from the main branch:

<https://github.com/Niket16/CDN-Test-Repo/tree/main>

https://cdn.jsdelivr.net/gh/Niket16/CDN-Test-Repo@main/assets/flash%20anime.jpg

### 

### Step 5: Update Your Angular Project

1. Open your Angular component files where you reference assets.
2. Update the asset URLs to use jsDelivr. For example:

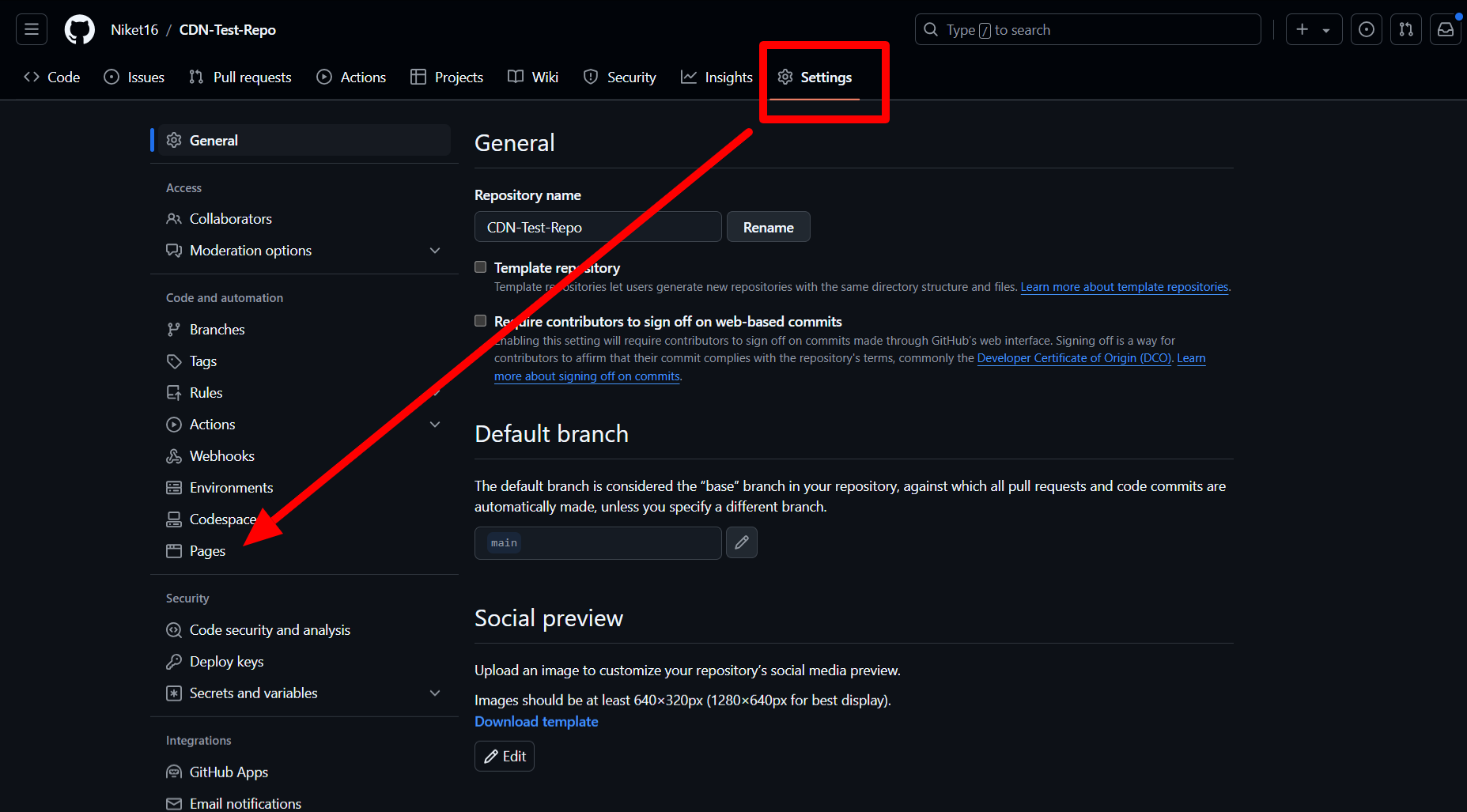
* // Before  
  logoUrl = 'assets/images/ flash anime.jpg';  
    
  // After  
  logoUrl = 'https://cdn.jsdelivr.net/gh/Niket16/CDN-Test-Repo@main/assets/flash%20anime.jpg';

## Option 2: GitHub Pages

GitHub Pages is a static site hosting service that takes files straight from a GitHub repository. While it’s primarily designed for hosting documentation and simple websites, it can be used to host Angular applications and serve as a CDN for our assets.

### Step 1: Follow the step 1-3 same as above

### Step 2: Now click on setting icon and click in pages



Now click on main branch and click on save.

A screenshot of a computer

Description automatically generated

After few moments you can see published URL same as below

A screenshot of a computer

Description automatically generated

### Step 3: Update Asset References

1. Open your Angular component files where you reference assets.
2. Update the asset URLs to use the GitHub Pages URL. For example:

* // Before  
  logoUrl = 'assets/images/flash.jpg';  
    
  // After  
  logoUrl = 'https://niket16.github.io/CDN-Test-Repo/assets/flash.jpg';

## Conclusion

**jsDelivr** is the better option if we are primarily looking for a **fast and easy-to-use CDN** with **global caching** and **versioning support**. It’s ideal for serving static assets like images or scripts directly from a GitHub repository without any complex setup.