

# GitHub Repository Setup Guide

This guide explains how to set up the GitHub repositories for DATA 322.

## Step 1: Create GitHub Repositories

### 1. Create the private repository:

- Go to GitHub and create a new repository named `data322-private`
- Make it private
- Do NOT initialize with README, .gitignore, or license
- Copy the repository URL

### 2. Create the public repository:

- Go to GitHub and create a new repository named `data322-public`
- Make it public
- Do NOT initialize with README, .gitignore, or license
- Copy the repository URL

## Step 2: Initialize Local Repositories

### 1. Initialize `data322-private`:

```
cd data322-private
git init
git add .
git commit -m "Initial commit: DATA 322 course structure"
git branch -M main
git remote add origin https://github.com/roverhol/data322-private.git
git push -u origin main
```

### 2. Initialize `data322-public`:

```
cd ../data322-public
git init
git commit --allow-empty -m "Initial commit: Public website"
git branch -M main
git remote add origin https://github.com/roverhol/data322-public.git
git push -u origin main
```

## Step 3: Configure GitHub Actions

The GitHub Actions workflow in `.github/workflows/quarto-publish.yml` needs to be updated with your repository information:

### 1. Update the workflow file:

- Open `.github/workflows/quarto-publish.yml`
- Replace `[OWNER]` with your GitHub username
- The workflow will use `GITHUB_TOKEN` automatically (no secrets needed)

### 2. Alternative: Use Personal Access Token (if needed):

- If you need more permissions, create a Personal Access Token (PAT)
- Go to GitHub Settings > Developer settings > Personal access tokens > Tokens (classic)
- Create a token with `repo` scope
- Add it as a secret named `GH_PAT` in the private repository settings
- Update the workflow to use `${{ secrets.GH_PAT }}` instead of `${{ secrets.GITHUB_TOKEN }}`

## Step 4: Enable GitHub Pages

1. Go to the `stat109-public` repository settings
2. Navigate to Pages
3. Source: Deploy from a branch
4. Branch: `main / / (root)`
5. Save

## Step 5: Test Deployment

1. Make a small change to any `.qmd` file in `stat109-private`
2. Commit and push:

```
git add .
git commit -m "Test deployment"
git push
```

3. Check the Actions tab in GitHub to see the workflow run
4. After a few minutes, check the public repository - it should have the rendered files
5. Visit the GitHub Pages URL to see the live site

## Troubleshooting

- **Workflow fails:** Check the Actions tab for error messages
- **Pages not updating:** Make sure GitHub Pages is enabled and pointing to the main branch
- **Permission errors:** Ensure the workflow has write permissions (check repository settings > Actions > General > Workflow permissions)