Here are the **easiest step-by-step instructions** to connect your **Eclipse IDE** to **GitHub**:

**✅ Prerequisites**

* **Eclipse IDE** (preferably Eclipse IDE for Enterprise Java and Web Developers)
* **GitHub account**
* Project opened in Eclipse

**🔧 1. Install Git Plugin (if not already installed)**

Most modern Eclipse versions have **EGit (Git integration)** built-in.

To check:

* Go to Help → Eclipse Marketplace
* Search for: **Git Integration**
* If it's **installed**, you're good. If not, install: **"EGit - Git Integration for Eclipse"**

**🧭 2. Create a GitHub Repository**

1. Go to <https://github.com>
2. Click + (top right) → New repository
3. Enter:
   * Repository name
   * Description (optional)
   * Keep it **public** or **private**
4. **Don't initialize with README**
5. Click **Create repository**

**🔀 3. Convert Your Eclipse Project to a Git Repository**

1. In Eclipse:
   * Right-click your project → Team → Share Project...
2. Choose **Git**
3. Click **Create Repository**
   * Leave the default location or choose your project root
4. Click **Finish**

✅ Your project is now a **local Git repo**.

**⬆️ 4. Commit Your Code**

1. Right-click your project → Team → Commit...
2. Add a commit message (e.g., Initial commit)
3. Select all files to commit
4. Click **Commit**

**🌐 5. Link to Remote GitHub Repository**

1. Go to:
   * Right-click project → Team → Remote → Push
2. In **URI** section:
   * **URI**: https://github.com/your-username/your-repo-name.git
   * **Host**: auto-filled
   * **Repository path**: auto-filled
   * **Protocol**: https
3. Enter your GitHub **username and personal access token** (not your GitHub password).  
   If you don’t have a token:

🔒 [Create a GitHub token here](https://github.com/settings/tokens) 🡪 profile Setting 🡪Developer Setting 🡪 Personal Access Token🡪 Select all the check box🡪 Generate Token and save it for future use

1. Click **Next**

**🧩 6. Set Up Push Configuration**

1. Choose:
   * Source ref: master or main
   * Destination ref: refs/heads/main or refs/heads/master
2. Click **Add Spec**
3. Click **Finish**

✅ Your project is now connected and **pushed to GitHub**!

**🔁 7. Future Commits and Pushes**

* Right-click project → Team → Commit to commit
* Then: Team → Push to Upstream to push to GitHub