Uploading Gmail AI Agent to GitHub

This guide provides step-by-step instructions for uploading the Gmail AI Agent project to GitHub.

Prerequisites

- Git installed on your system
- A GitHub account
- The Gmail AI Agent project files on your local machine

Setting Up Git

1. Create a GitHub Repository

1. Log in to your GitHub account
2. Click the "+" icon in the top-right corner and select "New repository"
3. Enter a repository name (e.g., "gmail-ai-agent")
4. Add an optional description
5. Choose visibility (public or private)
6. Do NOT initialize the repository with README, .gitignore, or license
7. Click "Create repository"

2. Configure Git Locally

Open a terminal/command prompt and navigate to your project folder:

```
cd path/to/gmail-ai-agent
```

Initialize Git in your project folder:

```
git init
```

Creating a .gitignore File

Create a `.gitignore` file to exclude unnecessary files:

```
# Dependencies
node_modules/
.pnp/
.pnp.js

# Environment variables
.env
.env.local
.env.development.local
.env.test.local
.env.production.local

# Build artifacts
dist/
build/
release/
out/

# Debug logs
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# IDE/Editor folders
.vscode/
.idea/
*.sublime-project
*.sublime-workspace

# OS files
.DS_Store
Thumbs.db

# Electron development
electron-builder.env
```

Save this file as `.gitignore` in your project root.

Adding and Committing Files

1. Stage Files

Add all project files to Git's staging area:

```
git add .
```

2. Verify Staged Files

Check which files are staged:

```
git status
```

Review this list to ensure no sensitive or unnecessary files are included.

3. Initial Commit

Commit your files with a descriptive message:

```
git commit -m "Initial commit: Gmail AI Agent"
```

Connecting to GitHub and Pushing

1. Link to GitHub Repository

Connect your local repository to the GitHub repository:

```
git remote add origin https://github.com/yourusername/gmail-ai-agent.git
```

Replace `yourusername` with your GitHub username and `gmail-ai-agent` with your repository name.

2. Push to GitHub

Push your code to GitHub:

```
git push -u origin main
```

Note: If you're using an older version of Git, the default branch might be called `master` instead of `main` :

```
git push -u origin master
```

Verifying the Upload

1. Go to your GitHub repository page

2. Refresh the page
3. Your files should now be visible

Additional Tips

Setting Up GitHub Authentication

If you haven't set up authentication with GitHub:

1. For HTTPS access: You'll be prompted for your GitHub credentials
2. For SSH access (recommended):
   - Generate an SSH key: `ssh-keygen -t ed25519 -C "your_email@example.com"`
   - Add it to GitHub: https://github.com/settings/keys
   - Use SSH URL: `git remote add origin git@github.com:yourusername/gmail-ai-agent.git`

Large Files Strategy

If you have large files in your project:

- Consider using Git LFS for large binary files
- Break up large commits into smaller ones
- If you have large binary files in your history that you want to remove, consider using BFG Repo-Cleaner

Branch Strategy

Consider creating a branch structure:

```
git branch development
git checkout development
# Make changes here first, then merge to main when stable
```

GitHub Actions Setup

Consider setting up GitHub Actions for CI/CD:

1. Create a `.github/workflows` directory
2. Add a YAML file (e.g., `build.yml`) with your workflow configuration
3. Example workflow for testing:

```yaml
name: Build and Test

on:
  push:
    branches: [ main, development ]
  pull_request:
    branches: [ main ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: Use Node.js
      uses: actions/setup-node@v2
      with:
        node-version: '16'
    - run: npm ci
    - run: npm run build --if-present
    - run: npm test --if-present
```

Troubleshooting

Authentication Issues

If you encounter authentication issues:

- Ensure you have the correct permissions for the repository
- Try using a Personal Access Token instead of password
- For SSH: verify your key is added to your GitHub account

File Size Limits

If you encounter file size issues:

- GitHub has a file size limit of 100 MB
- Use Git LFS for larger files
- Consider removing large files from your repository history if they were added accidentally

Push Rejection

If your push is rejected due to non-fast-forward errors:

- Pull changes from remote: `git pull --rebase origin main`
- Resolve any merge conflicts
- Try pushing again

Updating Your Repository

After initial upload, use these commands to update your repository:

```
# Pull latest changes
git pull

# Add new/modified files
git add .

# Commit changes
git commit -m "Description of changes"

# Push to GitHub
git push
```

Additional Resources

- GitHub Documentation
- Git Documentation
- GitHub Desktop (GUI alternative)