

```
import { Octokit } from '@octokit/rest';

import { makeld } from '../random';

interface Options {

  name: string;

  code: string;

  ownerName: string;

  ownerEmail: string;

}

const githubAccessToken = process.env.SWARM_PUBLISHER_GITHUB_ACCESS_TOKEN || '';

const repoOwner = process.env.SWARM_REPO_OWNER || '';

const repo = process.env.SWARM_REPO_NAME || '';

const baseBranch = 'main';

const publishSwarmToGithub = async ({

  name,

  code,

  ownerName,

  ownerEmail,

}: Options) => {

  if (!githubAccessToken) {

    throw new Error('GitHub access token is missing');

  }

  const octokit = new Octokit({ auth: githubAccessToken });

  try {

    const newBranch = `swarm_${ownerName}_${name}_${makeld(5)}`;
```

// Step 1: Create a new branch

```
const createBranchResponse = await octokit.git.createRef({  
  owner: repoOwner,  
  repo,  
  ref: `refs/heads/${newBranch}`,  
  sha: (  
    await octokit.repos.getBranch({  
      owner: repoOwner,  
      repo,  
      branch: baseBranch,  
    })  
  ).data.commit.sha,  
});
```

// Step 2: Create a new directory with name

```
await octokit.repos.createOrUpdateFileContents({  
  owner: repoOwner,  
  repo,  
  path: `prebuilt_swarms/${name}/${ownerName}_${name}.py`,  
  message: `feat: add ${name} swarm , created by ${ownerName}`,  
  content: Buffer.from(code).toString('base64'),  
  branch: newBranch,  
});
```

// Step 4: Create a pull request

```
const res = await octokit.pulls.create({
```

```

    owner: repoOwner,

    repo,

    title: `Add ${name} swarm by ${ownerName}`,

    head: newBranch,

    base: baseBranch,

    body: `Add ${name} swarm created by ${ownerName} | ${ownerEmail}`,

  });

  if (res.status === 201) {

    return res.data;

  }

} catch (error) {

  // @ts-ignore

  throw new Error(`Failed to publish swarm to GitHub: ${error?.message}`);

}

return null;

};

const getSwarmPullRequestStatus = async (pull_number: string) => {

  const octokit = new Octokit({ auth: githubAccessToken });

  try {

    const status = await octokit.pulls.get({

      owner: repoOwner,

      repo,

      pull_number: Number(pull_number),

    });

    if (status.status === 200) {

      return status.data;

    }

  }

```

```
    }  
  
    return status.data;  
  
  } catch (error) {  
  
    return false;  
  
  }  
  
};  
  
export { publishSwarmToGithub, getSwarmPullRequestStatus };
```