Information Technology & Telecommunications

# Citywide GitHub Policy

**Final 1.0**
**2/3/2016**

City of New York
Department of Information Technology and Telecommunications
Application Development Management

# Citywide Policy for GitHub

## Table of Contents

# 1.0 Overview

## 1.1 Introduction

GitHub is a web-based application platform that enables users to manage and collaborate on various types of projects. Best known as a collaborative code development tool, the site augments the use of Git, a distributed revision control system (DVCS) used for source code management (SCM). Git users can work on the same project at the same time, merge different versions together, and revert back to earlier versions. GitHub streamlines these collaborative capabilities through its website.

It is important to note that GitHub's Terms of Service states that, "We claim no intellectual property rights over the material you provide to the Service. Your profile and materials uploaded remain yours." Therefore, any City material uploaded to the City's GitHub account remains the City's.

## 1.2 Audience

This policy is for City agencies interested in using GitHub. All City employees and vendors involved in a project that utilizes GitHub must be familiar with and adhere to this policy.

## 1.3 Purpose

The purpose of this policy is to outline permissible uses of GitHub as a collaborative code development tool between different target users: within City agencies, between different City agencies and agency consultants, and with the public.

By enabling the use of GitHub in New York City government, this policy also serves broader Citywide interests in promoting reusable software, pooling resources and mechanisms for collaboration, and expanding engagement with the public.

## 1.4 Definitions

### 1.4.1 Repository

A repository is the most basic element of GitHub, best explained as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

The difference between public and private repositories is that public repository contents are visible to anyone and private repository contents are visible only to designated project collaborators. Public repositories are free, and private repositories cost money. There are no other functional differences between public and private GitHub repositories.

### 1.4.2 Accounts

There are two types of GitHub accounts that are used to administer and maintain GitHub repositories: individual user accounts and organization accounts. Both account types can own zero or more public

and private repositories. Organization accounts are composed of user accounts which are then granted different permissions for working with the organization's repositories.

### 1.4.3 Permissions

Repositories owned by user accounts have two permission levels: owner (singular) and collaborator (multiple). Organization account repositories can have multiple owners and use the concept of teams to allow much more fine-grained control over permission levels. Individual user accounts are assigned to teams of one of the four following types: owners, administrators, write access, read access.

It is also important to note that neither public nor private GitHub repositories allow GitHub users who have not been explicitly designated as collaborators by the repository owner (and/or admin in the case of organizations) to modify repository content. Registered GitHub users with permission to view a repository can "fork" its contents and submit a pull (change) request back to the owner(s). If enabled by the repository owner(s), users can modify project Wiki content or use GitHub's built-in issues functionality to submit feature and report defects.

See also:

- GitHub Glossary
- GitHub Account Plans
- User vs. Organizational Accounts
- Permissions Levels: User Accounts
- Permissions Levels: Organizational Accounts

## 1.5 Scope

This policy outlines the acceptable use cases for City agencies to utilize GitHub:

1. **Internal use** of private GitHub repositories as a collaborative development tool within and between City agencies on City technology projects.
2. **Public posting** of City code as a public repository to an official City GitHub account to promote a more open and collaborative government.

# 2.0 Policy

## 2.1 Policy Statement

The City encourages the use of GitHub for internal collaboration and for public posting for purposes of more efficient collaboration as well as expanded engagement with the public. At the same time, the City must uphold the overarching responsibility to protect City data, ensure IT security of systems, and abide by relevant intellectual property rights. Therefore, all City employees and vendors involved in a project that utilizes GitHub must adhere to the requirements detailed in sections 2.2 to 2.7 regarding data protection, account set-up and management, usage, consultants, licensing, and roles and responsibilities.

## 2.2 Data Protection and Allowable Content

The following information cannot be uploaded to any City of New York GitHub repository, private or public, under any circumstances:

**Security and Privacy information:**

- Authentication credentials, such as usernames, passwords, private key files, etc.
- Information that provides direct or indirect information about or insight into non-public City virtual or physical infrastructure, such as private network configuration, firewall policies, internal IP addresses or hostnames, routing tables, etc.
- Sensitive, private, or confidential data as defined by the Data Classification Standard.

Development teams should keep such information locally and establish procedures for "plugging it in" after code checkout.

**Copyrighted or third-party material:**

- Code that was not created or is not owned by the City of New York.
- Code that the City was not granted explicit permission to redistribute.

It is the responsibility of the authorized personnel of each City agency using GitHub to enforce this policy. Examples of acceptable scenarios include open source software and software being developed under a contractual arrangement with external vendor/consultants.

The following table illustrates licensing and allowable data and content on City GitHub accounts.

| Repo type | Licensing | Data sensitivity | Allowable content | Visibility |
|---|---|---|---|---|
| **Public** | Apache 2.0 (see 2.5) | Public data only | 1. Original work authored by or for the City and owned by the City<br>2. 3rd party work whose redistribution has been explicitly granted to the City (e.g. open source software, explicit vendor contract) | Anyone |
| **Private** | N/A | | | Project members only |

### 2.2.1 Responding to Public Comments

- Any replies or interactions with public collaborators should adhere to the data protection guidelines detailed in this policy, namely that only data or information classified as public be discussed.
- City employees communicating on GitHub should conduct themselves appropriately as with any form of public communication, digital or otherwise. Any agency using GitHub can consider additional policies for employee conduct on GitHub that it deems appropriate.
- Further information about working with comments is available here.

## 2.3 Account Management

This section outlines three levels of account management:

- **City of New York organizational account**: Administered by DoITT, representing Citywide technology priorities and projects.
- **Individual City Agency organizational accounts**: Administered and set up by agencies.

- **City Employee Accounts**: Individual user accounts for City employees and authorized consultants.

### 2.3.1 City of New York's Organizational Account

In order to provide a secure and organized Citywide account and to ensure that user credentials are properly administered, DoITT administers the City of New York GitHub Citywide organizational account. As such, any public repository set up on this account as of the publishing of this policy must be administered by DoITT or a DoITT-designated administrator, which may include authorized personnel at other City agencies. Projects that predate this policy will be brought into compliance by March 2016. Projects that are determined to be Citywide priorities or involve multiple City agencies should strongly consider using a repository hosted on this Citywide organizational account in order to increase visibility and facilitate inter-agency collaboration.

#### 2.3.1.1 Requesting a repository on the NYC GitHub

City agency seeking to establish a repository on the Citywide account can do so by submitting a request to the Citywide Service Desk (MyDesk self-service portal or 212-NYC-HELP or NYCHelp@doitt.nyc.gov). In the request, please include:

1. Name of repo (should be all lower case)
2. Names of users who need access, their work email addresses, and their permission levels
    - Please refer to section 2.3.3 on City employee account setup. All team members should already have their GitHub accounts properly set up prior to submitting the request.
3. Confirmation that you've read and understand this policy

### 2.3.2 Individual City Agency Organizational Accounts

A City agency may set up its own agency organizational account on GitHub, provided it takes full responsibility for the management and expense of that account and it complies fully with this Citywide policy. Agencies hosting their own repositories must do so using this single, top-level organization account. In other words, all of an agency's GitHub projects should be in the agency's one, official account.

To provide clear accountability of what accounts are managed by what agencies across the City, DoITT will maintain a list of agency accounts. Therefore, when establishing an organizational account, the City agency must notify DoITT by submitting a ticket to the Citywide Service Desk (MyDesk, 212-NYC-HELP, NYCHelp@doitt.nyc.gov) and including a link to the account and the contact information for the account administrator in the email.

Official City GitHub repositories *must be owned by an organization account*. All agencies (including DoITT) are prohibited from publishing projects owned by individual user accounts. City employees are allowed to maintain their own forks of **public** City repositories; however, they must abide by the same policy restrictions that apply the City's use of GitHub. Employees cannot use any City private repository content in their own non-City-related GitHub repositories.

Interagency projects or projects determined to be Citywide priorities should use the City of New York GitHub Citywide organizational account as stipulated in section 2.2.1.

### 2.3.3 City Employee Account Setup and Management

Only City employees may be GitHub repository owners (admin-level rights) to New York City government organizational accounts. These agency administrators are responsible for setting up repositories, assigning team members, and permissions for GitHub projects.

City staff (including authorized consultants) must use their official City email address (@agency.nyc.gov) when working with official City repositories. Project members who do not have an existing GitHub account can simply provide their City email address when creating their new user account.

City staff with an existing GitHub user account must add their City email address to their account profile using GitHub's site before using GitHub in an official capacity for the City. Similarly, developers must configure their local, downstream City repositories to use their City email address prior to pushing changes back to GitHub. GitHub allows users to keep their email addresses private, and City staff can choose to configure their accounts as such.

City staff do not necessarily have to designate their work address as their primary email in their GitHub account profile. However, City staff must ensure that any work on GitHub on an official City project is associated with their City email.

See also:

- Adding an email to your GitHub Account
- Changing your primary email

It is the responsibility of each agency to ensure that all City staff granted permission to use GitHub in an official capacity for the City have read and understand GitHub's recommendations for preventing unauthorized access.

GitHub best practices recommend that that individuals create only a single GitHub account adding different secondary/additional email addresses per project as necessary.

## 2.4 Internal Use

The internal use of GitHub's private repositories may be used only within City government: within a City agency or between City agencies, including authorized consultants, collaborating on a technology project. The internal use of GitHub aims to leverage the collaborative benefits and workflow efficiencies that the platform offers.

It is the responsibility of the organizational account administrators to configure new repositories, teams, and individual user accounts to enforce this and any applicable City policies, including all IT Security policies.

As with any software development project, proper permissions must be configured to avoid un-authorized access within a private repository beyond the designated team.

### 2.4.1 Contractors/Consultants

City employees who manage consultants must ensure that consultants' use of GitHub complies with all relevant non-disclosure agreements (NDAs) and City contract terms, such as the stipulation that the City owns and maintains the rights to code developed by consultants. Authorized consultant users must use a City-issued email on their GitHub account for working on City projects as described in 2.2.3.

Consultants can only be assigned to projects as team members and cannot serve as repository owners or maintain any administration rights. Agency administrators are responsible for revoking access permissions when a consultant leaves a project and ensuring that the consultant removes their official City email (@agency.nyc.gov) from their individual account if the consultant leaves the agency.

## 2.5 IT Security Controls

Ensuring the integrity of the source code hosted on GitHub is essential both for project collaborators developing against local repository clones and the general population of developers wishing to build/fork or examine the City's repositories. The City does not own or operate the systems that host its GitHub repositories (GitHub does!), therefore, it is important that City GitHub projects provide information on how to insure downloaded content does not contain unauthorized modifications. Fortunately, because GitHub does not allow unencrypted access to repository files and due to the core design of Git itself, this verification process is straightforward.

GitHub repository content can **only** be accessed over encrypted channels (HTTPS, SSH), so a "man-in-the-middle" attack cannot occur without detection by a client's browser or local Git installation using the 'clone', 'fetch' or 'pull' commands. As such, project collaborators and users should not ignore or override invalid certificate warnings, encryption/decryption errors, etc. Furthermore, any local content left-over after failed transfers should be deleted immediately.

Git's core versioning and content retrieval functionality is implemented using cryptographic hashing of repository content and metadata. The algorithm that produces the identifiers for accessing repository content is computed using all previously calculated identifiers.

This design allows Git to detect content changes very quickly: for example, to determine if the contents of two directories are the same Git only needs to compare their 160-bit SHA-1-generated hashes instead of the potentially hundreds of subfolders and files. It also insures that any direct content modification to repository files or metadata will result in different SHA-1 identifiers. As a result, all of the Git commands for synchronizing a local repository with its upstream counterparts (GitHub in this case) will fail if any incoming content was changed in-transit.

To ensure the users of any public repository are aware of the built-in content-integrity checks provided by Git and GitHub, administrators of a public repository should include a README.md file in the top-level project directory which contains the following such that it is highly visible at the beginning of the text:

> "The City of New York recommends that users wishing to store City repository content on their local computers adhere to the following guidelines to insure the integrity of transferred information:
>
> 4. Never disable, circumvent or ignore SSL/TLS warnings or errors issued by the browser or local Git installation preventing the successful transfer of data from GitHub to the client machine. This includes the overriding of untrusted or self-signed certificates, configuration of Git to skip SSL verification, etc.
> 5. If the local Git installation fails to completely clone, fetch/pull, etc. from a GitHub (or any other) repository, discard any partially transferred files. This is most easily accomplished by creating a new local branch solely for the purpose of running the Git remote-based command and deleting it permanently if the command fails.

6. Developers wishing to collaborate or fork City GitHub repositories should insure their local Git installation is up-to-date and properly configured. See https://help.github.com/articles/set-up-git/ for details.
7. Members of the public downloading compressed repository content from the GitHub site should take the same security precautions as taken when accessing file downloads from any trusted Internet web site (e.g., ensure up-to-date and enabled antivirus software, etc.)"

## 2.6 License Agreement

NYC GitHub content terms and conditions of use, reproduction, and distribution are defined the Apache License Version 2.0, an industry standard permissive license agreement:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

## 2.7 Roles and Responsibilities

**City employees** are responsible for abiding by the policies set forth in this document.

**Agency organizational account administrators** are responsible for configuring all repositories, teams, and individual user accounts managed under the agency account and enforcing all relevant City policies, including this one.

**Authorized Consultants** are responsible for adhering to their contracting requirements and are subject to the oversight of City project manager, and use of GitHub is no exception to this. Any GitHub use is subject to this ongoing contractual responsibility.

**DoITT authorized personnel** are responsible for administering the City of New York GitHub Citywide organizational account and vetting all development projects intended to be posted on this account; they must also designate any additional administrators at other City agencies as appropriate.

# 3.0 Additional Resources

- Using distributed workflows
- Resources on Open Source for Government
- Guides on GitHub
- Guide on using the GitHub Issues feature

# 4.0 Authority

DoITT was established by Local Law 24 of 1995 as "New York City's information technology and telecommunications agency."

DoITT's Application Development Management division develops enterprise web applications, mobile applications, and enterprise solutions for NYC.gov, CityShare, agency sites and 311. ADM leverages its expertise to guide agencies on the use of application development tools and technologies, such as GitHub.

## 5.0 Ownership and Contact

This policy is owned by Don Sunderland, Deputy Commissioner of Application Development Management.

Per section 2.3.2, if an agency chooses to set up their own organizational account, please notify DoITT by submitting a ticket to the Citywide Service Desk (MyDesk, 212-NYC-HELP, NYCHelp@doitt.nyc.gov).

For assistance in establishing a repository under the NYC GitHub account, please submit a Service Desk request per section 2.3.1.1. The ticket will be routed to the DoITT Application Support.

## 6.0 Authorship and Change History

**Contributors**

| | |
|---|---|
| Kim Truong, Policy Analyst, Governance and External Affairs – DoITT<br>Daynan Crull, Sr. Policy Advisor , Governance and External Affairs – DoITT | Don Sunderland, Deputy Commissioner, Application Development Management – DoITT<br>Colin Reilly, Director of GIS, Application Development Management – DoITT<br>Matthew Lipper, Senior GIS Architect, Application Development Management |

**Change Details**

| Version | Change Highlights | Author(s) | Date |
|---|---|---|---|
| 1.0 | Initial publication | M. Lipper, K. Truong, D. Crull, et al | 2/3/2015 |

# 7.0 Appendix: Project Assessment Checklist for Use of GitHub

Please complete the checklist to determine if the project can be posted to GitHub.

If all answers are "yes," posting to GitHub is possible. Any "no" answers require resolution.

| CONTENT | Yes | No |
|---|---|---|
| **Privacy** | | |
| 1. Is the content free of personally identifiable information (e.g. name, phone number, SSN, ID numbers)? | | |
| | | |
| **IT Security** | | |
| 2. Is the content free of authentication credentials, such as usernames, passwords, passphrases and private key files? | | |
| 3. Is the content free of information that provides direct or indirect information about or insight into non-public City virtual or physical infrastructure (e.g. private network configuration, firewall policies, internal IP addresses or hostnames, routing tables, etc.)? | | |
| 4. Is the content free of sensitive, private, or confidential data as defined by the Data Classification Standard (i.e. contains only public data)? | | |
| | | |
| **Copyright** | | |
| 5. Can you confirm that the content only contains the following:<br>– Original work authored by or for the City and owned by the City; and<br>– 3rd party work whose redistribution has been explicitly granted to the City (e.g. open source software, explicit vendor contract) | | |
| 6. Can you confirm that there are no terms of use or any exclusive licenses that would prohibit New York City government from licensing the content on GitHub? | | |
| | | |
| **Legal** | | |
| 7. Is the public release and use of the content permitted under law, contract, and policy (i.e. there are no legal, contractual, or policy restrictions or limitations)? | | |
| 8. If "no" to question 8, have the legal, contractual or policy restrictions or limitations been resolved? | | |
| | | |
| **ADMINISTRATION** | **Yes** | **No** |
| 9. Is the project in question a multi-agency collaboration or something that is determined to be a top City priority? If yes to either, this project should be placed in a repository on the Citywide organizational account. | | |
| 10. Does your agency have its own organizational account? If yes (and no to question #13) | | |

| | | |
|---|---|---|
| contact your agencies GitHub administrator to set up the repository for this project. If your agency does not have an account or wish to set one up, they may use the Citywide organizational account. | | |
| 11. Is there a designated repository administrator for the project, who will manage the repository team, permissions levels, and project activities? | | |
| 12. If the project involves consultants, have their contracts been reviewed to ensure that the City maintains ownership of their contributions? | | |
| 13. Have all contributors of the project configured their GitHub account to include their City email address and ensured that all official City work will be associated with this address? | | |