KL (DEEMED TO BE UNIVERSITY) | HTE HONORS THROUGH EXPERIENTIAL LEARNING

EXPERIENTIAL LEARNING & GLOBAL ENGAGEMENT

OPEN SOURCE ENGINEERING

Student ID: 2400030033

Semester: Odd

Academic Year: 2025-2026

Course Code: 24CS02EF

Under the guidance of

Dr. Arunkumar bala

1 Understanding the Core Ubuntu Linux Distribution

1.0.1 1. Overview and Philosophy

Ubuntu is a powerful, free, and open-source operating system built upon the stable foundation of Debian Linux. It stands as the world's most popular Linux distribution for desktop use, successfully blending cutting-edge features with unparalleled user-friendliness. Developed and maintained by Canonical Ltd., Ubuntu's guiding principle is "Linux for human beings." This philosophy drives its commitment to accessibility, stability, and providing an intuitive computing experience for everyone, from novice users to seasoned developers.

1.0.2 2. The Desktop Experience (GNOME)

The standard Ubuntu desktop utilizes the **GNOME** desktop environment, which presents a modern, clean, and highly efficient graphical interface. The key design elements include a permanent dock (launcher) on the left side for quick access to essential applications, and the **Activities Overview**. This view, easily accessed by pressing the Super (Windows) key, provides a centralized hub for managing all open windows, workspaces, and system-wide searching. This streamlined workflow makes Ubuntu feel contemporary and ensures high productivity. Furthermore, Ubuntu is recognized for its strong, out-of-the-box hardware detection and compatibility, simplifying the setup process for most users.

1.0.3 3. Software Management and Packaging

Ubuntu employs a robust dual-system for software management. The traditional and reliable **Advanced Packaging Tool (APT)** manages **DEB** packages, handling core system utilities and standard applications sourced from official repositories. Complementing this is the use of **Snaps**, a modern, containerized package format pioneered by Canonical. Snaps bundle an application with all its required dependencies, guaranteeing consistent performance across different Ubuntu versions. Crucially, Snaps run in a **sandboxed** environment, isolating them from the rest of the operating system to significantly enhance overall application security. This flexibility ensures users have access to a vast, up-to-date, and secure software library.

2 Encryption and GPG

2.1 Types of Encryption in Ubuntu

Ubuntu offers two primary approaches to encryption: **Full Disk Encryption (FDE)** and **File/Directory Encryption**.

2.1.1 1. Full Disk Encryption (FDE)

- **What it is:** FDE encrypts the entire hard drive or a large partition, including the operating system files, swap space, and user directories.
- **How it works:** Ubuntu uses **LUKS** (Linux Unified Key Setup) for FDE. When the system boots, you are prompted for a **passphrase**. If correct, LUKS decrypts the entire drive, and the decryption process runs transparently in the background while the system is in use.
- **Purpose:** The primary defense against data loss due to **theft** or **physical access** to the computer when it is turned off. If someone steals the hard drive, the data is useless without the LUKS passphrase.
- **Implementation:** FDE is typically enabled during the Ubuntu installation process by selecting the ”Encrypt the new Ubuntu installation” option. It's much more difficult to enable after installation.

2.1.2 2. File and Directory Encryption

- **What it is:** This method encrypts specific files, directories, or messages, offering granular control over which data is protected.
- **Tools:**
  - **GPG (GNU Privacy Guard):** The standard, used for encrypting individual files and especially for secure communication using **public-key cryptography**.
  - **eCryptfs (older):** Previously used for encrypting the user's Home directory, but has been largely phased out for FDE.

2.2 GPG (GNU Privacy Guard) Explained

GPG is the GNU implementation of the **OpenPGP** standard (originally Pretty Good Privacy - PGP). It is essential for protecting individual files and ensuring secure, authenticated communication.

2.2.1 1. Core GPG Concepts

GPG relies on **asymmetric cryptography**, which uses a pair of mathematically linked keys:

- **Public Key:** This key is shared with everyone. It can be used to **encrypt** a message that only you can read, or to **verify** a signature you created.
- **Private (Secret) Key:** This key is kept **secret** and is protected by a strong passphrase. It is used to **decrypt** messages sent to you, or to **digitally sign** files to prove they came from you.

2.2.2 2. Basic GPG Command-Line Usage

GPG is usually pre-installed on Ubuntu and is primarily used through the command line (Terminal).

A. Generating a Key Pair The first step is to create your public and private key pair:
Bash

```
gpg --full-generate-key
```

You will be prompted to select the key type (RSA and RSA is common), keysize (4096 is recommended), expiration date, and your Real Name, Email, and a strong **passphrase** to protect your private key.

B. Encrypting a File for Yourself (Symmetric Encryption) To quickly encrypt a file using a single passphrase (like a standard password), use symmetric encryption:
Bash

```
gpg -c myfile.txt
```

This command will prompt you for a passphrase and create an encrypted file named `myfile.txt.gpg`.

C. Encrypting a File for Someone Else (Asymmetric Encryption) To securely send a file, you must use the recipient's **Public Key** (which you must have previously imported into your keyring with `gpg --import`):
Bash

```
gpg --encrypt --recipient "recipient@example.com" mysecretfile.doc
```

This creates `mysecretfile.doc.gpg`. Only the recipient, who holds the corresponding Private Key, can decrypt it.

D. Decrypting a File To decrypt a file that was encrypted for you:
Bash

```
gpg --decrypt mysecretfile.doc.gpg
```

You will be prompted for the passphrase that protects your Private Key. You can use the `--output` option to specify the decrypted

3 Sending Encrypted Email

3.1 Prerequisite: Setting Up GPG

Before you can send or receive encrypted mail, both you and your recipient must have GPG keys set up and exchanged:

1. **Generate Keys:** Both parties must have generated a public/private key pair using GPG (as discussed previously, using `gpg --full-generate-key`).
2. **Exchange Public Keys:** You need the recipient's **Public Key**, and they need your Public Key. You can exchange these by:
   - **Exporting** the key: `gpg --armor --export 'Recipient Name' > recipient_key.asc` and sending the `.asc` file.
   - **Uploading** the key to a public key server.
3. **Import Key:** You must import the recipient's key into your GPG keyring: `gpg --import recipient_key.asc`.

3.2 Sending the Encrypted Email

The most common and user-friendly way to send GPG-encrypted emails on Ubuntu is by using **Mozilla Thunderbird** with the **Enigmail** add-on (or its built-in equivalent in modern versions of Thunderbird).

3.2.1 1. Compose the Message

- **Open Thunderbird** and start composing a new email.
- Write your message as usual.

3.2.2 2. Encryption and Signing

You will use the GPG function built into the mail client to perform two critical steps:

1. **Encryption:** You must encrypt the email using the **recipient's Public Key**. Only their corresponding **Private Key** can decrypt it. If you have multiple recipients, you must encrypt the message using the Public Key of *every single recipient*.

2. **Digital Signature:** You **sign** the email using **your Private Key**. This allows the recipient to verify that the email truly came from you and has not been tampered with in transit.

In Thunderbird, this is typically done by clicking a dedicated **OpenPGP or Security** menu or button within the compose window and ensuring both the **"Encrypt"** and **"Sign"** options are checked.

3.2.3 3. Verification and Sending

- The client will check that you have the required **Public Key** for the recipient(s). If a key is missing, it will warn you.

- When you click **Send**, Thunderbird uses GPG to encrypt the message body and attach your digital signature before transmitting the scrambled data.

3.2.4 4. Recipient's Experience (Decryption)

1. The recipient receives the scrambled email.

2. Their email client automatically uses their **Private Key** (protected by their passphrase) to decrypt the message contents, revealing the original text.

3. Their client simultaneously uses your **Public Key** to verify the digital signature, confirming the email's authenticity.

4 Privacy Tools From Prism Break

4.0.1 1. Tor Browser (Web Browsers / Anonymizing Networks)

- **What it is:** A web browser built on Firefox that routes your internet traffic through the Tor network, a volunteer-operated network of relays.

- **Privacy Focus:** Provides **strong anonymity** by obscuring your IP address and location from the websites you visit. It also includes anti-fingerprinting measures.

- **PRISM Break Note:** PRISM Break strongly recommends using Tor Browser for all web surfing when maximum anonymity is required.

4.0.2 2. Debian (Operating Systems)

- **What it is:** A popular and highly ethical GNU/Linux distribution known for its strict adherence to Free Software principles and ethical manifesto.
- **Privacy Focus:** Unlike proprietary operating systems like Windows and macOS (which PRISM Break generally avoids), Debian is fully open-source, allowing for audits. It has a long tradition of software freedom and transparency.
- **PRISM Break Note:** It's recommended as a top GNU/Linux choice for users transitioning from proprietary systems, highlighting its commitment to free software and its stable nature.

4.0.3 3. Thunderbird (Email Clients)

- **What it is:** A free, open-source, and cross-platform email client developed by Mozilla.
- **Privacy Focus:** Thunderbird is the top choice for desktop email due to its open-source nature and its long-standing **native support for OpenPGP** (GPG) encryption and digital signatures. This allows users to easily encrypt and authenticate their emails end-to-end.
- **PRISM Break Note:** It is highly recommended for securely managing email with built-in PGP features.

4.0.4 4. KeePassXC (Password Managers)

- **What it is:** A free, open-source, and cross-platform password manager.
- **Privacy Focus:** It stores all your passwords in a single, highly encrypted database file that is stored **locally** on your device, giving you total control over your sensitive data. It does not rely on a cloud service.
- **PRISM Break Note:** It is preferred for its strong encryption, open-source license, and local-only storage, minimizing exposure to third-party services.

4.0.5 5. Firefox (Web Browsers)

- **What it is:** A fast, flexible, and secure web browser developed by the non-profit Mozilla Foundation.

- **Privacy Focus:** Firefox is open-source and provides extensive privacy controls, including enhanced tracking protection (ETP), container technology, and a robust add-on ecosystem for further hardening security (like uBlock Origin).
- **PRISM Break Note:** While Tor Browser is for anonymity, Firefox is the recommended alternative for general web use when a site doesn't work well with Tor, provided the user configures its settings and replaces the default search engine with a privacy-focused one.

5 Open Source License

Certainly. Here is the information about the **MIT License** organized into clear, descriptive headings, strictly maintaining a paragraph-only format within each section.

5.1 The Core Purpose and Classification

The MIT License is renowned as one of the most permissive and concise open-source licenses currently in use. Originating from the Massachusetts Institute of Technology, its primary goal is to encourage maximum adoption and reuse of software with minimal legal friction. It is formally classified as a **permissive license**, meaning it grants users broad rights to use, modify, and distribute the software without imposing the reciprocal sharing obligations seen in copyleft licenses, such as the GNU General Public License (GPL). This makes the MIT License highly favorable for both commercial enterprises and proprietary software development.

5.2 Granted Rights and Permissions

The license grants blanket permission to any individual or entity obtaining a copy of the software and its associated documentation to deal with the Software without restriction. Specifically, users are granted explicit rights to **use, copy, modify, merge, publish, distribute, sublicense, and/or sell** copies of the software. This expansive grant allows developers to incorporate MIT-licensed code into projects that may ultimately be closed-source and sold commercially, provided they meet the few mandated conditions.

5.3 The Only Two Conditions for Distribution

Unlike licenses that enforce reciprocal sharing, the MIT License has only two critical requirements that must be met when the software is distributed or included in a larger work. The first condition is the mandatory inclusion of the original **Copyright Notice** (e.g., `Copyright <YEAR> <COPYRIGHT HOLDER>`).

The second is the mandatory inclusion of the full **License Text** itself. If these two simple requirements are satisfied, the user can otherwise treat the code as they wish, including releasing their modifications under a proprietary license.

5.4 Disclaimer of Warranty and Liability

A key component of the MIT License is its comprehensive liability disclaimer, which serves to protect the original authors. The license emphatically states that the software is provided **"AS IS,"** meaning it comes without any guarantee or warranty of any kind, whether express or implied, including warranties of merchantability or fitness for a particular purpose. Furthermore, the license explicitly protects the authors and copyright holders, asserting they **shall not be held liable** for any claim, damages, or other liability arising from the use or other dealings in the software. This places the entire risk associated with the software onto the end-user.

6 Self Hosted Server

6.1 Introduction to Kutt.it

Kutt.it is an open-source URL-shortening service designed to be fast, modern, and easy to manage. It allows users to create short links, track statistics, and use custom domains. Because it is self-hostable, users have full control over their data, security, and branding.

6.2 Key Features

Kutt offers features such as custom URLs, password protection, expiration times, and private analytics. It provides an admin dashboard to manage users and links, supports multiple databases, and allows disabling registrations or anonymous link creation. It is designed to work efficiently in self-hosted environments.

6.3 Installation Process

Installing Kutt requires Node.js and optionally a database like SQLite, Postgres, or MySQL. After downloading the project, users install dependencies with npm install, initialize the database using npm run migrate, and start the server with npm run dev or npm start. During first launch, the app prompts for creating an admin account, after which the system becomes ready to use.

6.4 Docker Setup

Kutt can also be installed using Docker for a simpler and automated setup. With Docker installed, users run the docker compose up command to start the application using the provided docker-compose files. Different configurations support SQLite, Postgres, MariaDB, and Redis, making deployment fast and consistent across environments.

6.5 Configuration

Kutt relies on environment variables stored in a .env file to control the app's behavior. These variables configure database settings, site information, email options, rate limiting, and the JWT secret used for authentication. By adjusting these values, users can tailor the server for development or production use.

6.6 Themes and Customization

Kutt supports theme customization through its custom folder, where users can place their own CSS, images, or HTML view templates. This allows replacing the default styles, adding new branding, or applying complete visual themes. Customizations load automatically and can change the appearance of the interface without modifying core files.

6.7 API Support

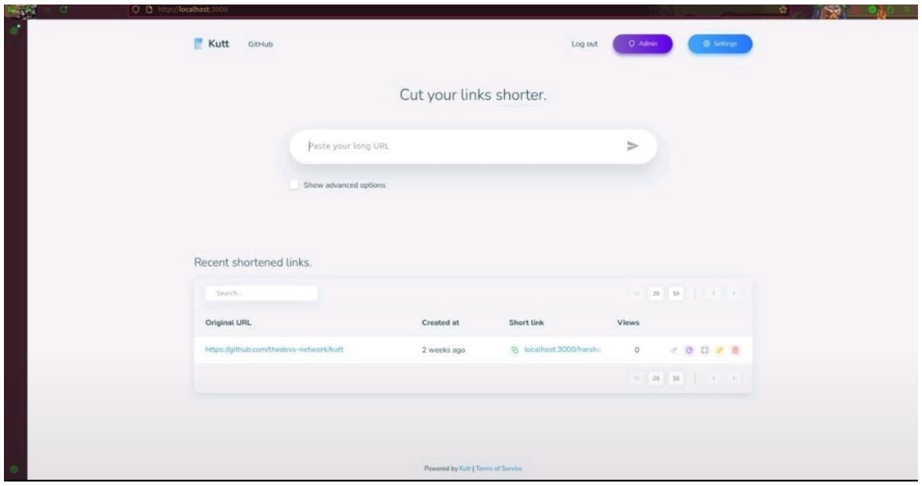
The platform provides a RESTful API that allows developers to automate link creation and management. Applications, scripts, and browser tools can interact with the Kutt server using API calls, enabling advanced integrations for automated shortening and analytics retrieval.

6.8 Browser Extensions and Integrations

Kutt offers official extensions for Chrome and Firefox, allowing quick URL shortening directly from the browser. It also integrates with tools like ShareX, Alfred workflows, and iOS shortcuts. Third-party libraries for languages such as Python, JavaScript, Ruby, Rust, and Go help developers easily use Kutt in their projects.

Kutt Resources

Translated Document



Original URL	Created at	Short link	Views
https://github.com/thedevs-network/kutt	2 weeks ago	localhost:3000/harsh	0

7 Open Source Contribution

7.1 PR 1 : First Contribution

7.1.1 Goal

The project's objective is to simplify the standard open-source contribution workflow, allowing beginners to easily add their name to the project's `Contributors.md` file.

7.1.2 The Contribution Workflow

The tutorial details the standard **fork - clone - edit - pull request** sequence, essential for collaborative coding.

7.1.3 1. Setup

- **Fork:** Create a copy of the repository in your personal GitHub account.
- **Clone:** Download the forked repository to your local machine using the `git clone` command and the SSH URL.
- **Prerequisites:** Ensure **Git** is installed; alternatives for users uncomfortable with the command line (GUI tools) are provided.

7.1.4 2. Making Changes

- **Branch:** Create a new isolated branch for your changes using `git switch -c your-new-branch-name`.

- **Edit:** Add your name to the `Contributors.md` file using a text editor.
- **Commit:** Stage the changes with `git add Contributors.md` and save them locally with `git commit -m "Add your-name to Contributors list"`.

7.1.5 3. Submission

- **Push:** Upload your local branch to your GitHub fork using `git push -u origin your-branch-name`.
- **Pull Request (PR):** Go to your GitHub repository and submit a PR via the "Compare & pull request" button for review by the project maintainers.

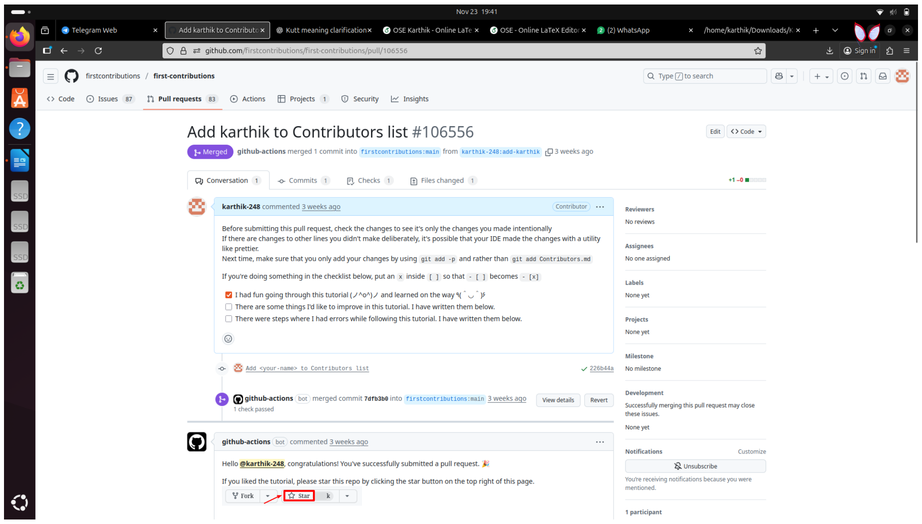
7.1.6 Difficulties and Solutions

The guide anticipates and solves two common beginner issues:

- **Old Git Version:** If the `git switch` command fails, use the older command: `git checkout -b your-new-branc`
- **Authentication Error:** If `git push` fails due to GitHub removing password support, the solution is to configure an **SSH key** or a **Personal Access Token** and ensure your remote URL is set to the **SSH protocol** (`git remote set-url origin git@github.com:...`).

7.1.7 Next Steps

Upon merging the PR, the user is encouraged to celebrate their first contribution and seek out other beginner-friendly issues on the project list.



Telegram Web
Add karthik to Contributo
Kutt meaning clarification
OSE Karthik - Online LaTe
OSE - Online LaTeX Editor
(2) WhatsApp
/home/karthik/Downloads/K
github.com/firstcontributions/first-contributions/pull/106556
firstcontributions / first-contributions
Type / to search
Code
Issues 87
Pull requests 83
Actions
Projects 1
Security
Insights
Edit
Code

Add karthik to Contributors list #106556

Merged github-actions merged 1 commit into `firstcontributions:main` from `karthik-248:add-karthik` 3 weeks ago

Conversation 1 · Commits 1 · Checks 1 · Files changed 1 · +1 −0

karthik-248 commented 3 weeks ago · Contributor

Before submitting this pull request, check the changes to see it's only the changes you made intentionally
If there are changes to other lines you didn't make deliberately, it's possible that your IDE made the changes with a utility like prettier.
Next time, make sure that you only add your changes by using `git add -p` and rather than `git add Contributors.md`

If you're doing something in the checklist below, put an `x` inside `[ ]` so that `- [ ]` becomes `- [x]`

- [x] I had fun going through this tutorial (ノ^o^)ノ and learned on the way ٩(ˆ ◡ ˆ)۶
- [ ] There are some things I'd like to improve in this tutorial. I have written them below.
- [ ] There were steps where I had errors while following this tutorial. I have written them below.

`Add <your-name> to Contributors list` ✓ 226b44a

github-actions bot merged commit `7dfb3b0` into `firstcontributions:main` 3 weeks ago
1 check passed · View details · Revert

github-actions bot commented 3 weeks ago

Hello @karthik-248, congratulations! You've successfully submitted a pull request. 🎉

If you liked the tutorial, please star this repo by clicking the star button on the top right of this page.

Fork · Star · k

Reviewers: No reviews
Assignees: No one assigned
Labels: None yet
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request may close these issues. None yet
Notifications: Customize · Unsubscribe · You're receiving notifications because you were mentioned.
1 participant

7.2 PR 2 : JSON Dataset

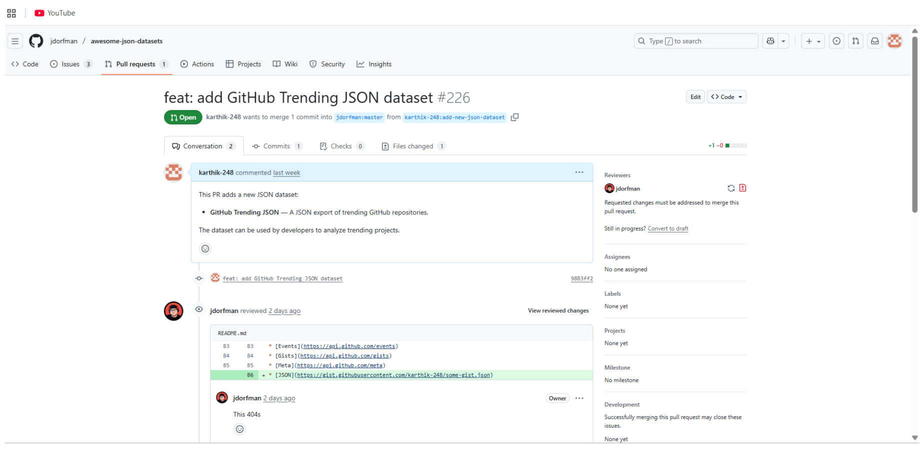
This pull request introduces a new JSON dataset designed to provide a structured export of trending GitHub repositories. The dataset, titled GitHub Trending JSON, enables developers, researchers, and analysts to conveniently access up-to-date trending project information in a unified, machine-readable format. The addition enhances the repository's overall dataset collection by contributing a practical resource for building dashboards, analytics pipelines, developer tooling, or trend-monitoring applications. The dataset is contributed as part of an effort to expand the breadth of publicly available JSON datasets accessible to the open-source community.

7.2.1 Licensing and Self-Hosting Options

The GitHub Trending JSON dataset is provided under the repository's open-source license, allowing developers to freely use, modify, and integrate it into their projects. The dataset is typically hosted through GitHub Gist or raw GitHub content, ensuring easy public access. For users who prefer self-hosting, the JSON file can be served via static hosting platforms such as GitHub Pages, Netlify, Vercel, or any Nginx/Apache server. Developers can also run it locally using lightweight servers like Node.js or Python's built-in HTTP module. Docker-based deployment is supported for creating isolated, reproducible environments. Self-hosting provides greater reliability, enables custom update schedules, and ensures long-term accessibility of the dataset.

7.2.2 Community and Support

The GitHub Trending JSON dataset is maintained as part of a collaborative open-source ecosystem, where community involvement plays a key role in improving data quality and accessibility. Contributors and users can engage through the repository's issue tracker to report bugs, suggest enhancements, or request new dataset features. Discussions and clarifications typically take place through GitHub comments, ensuring transparent and traceable communication. Developers are encouraged to participate by reviewing pull requests, validating dataset links, and contributing additional resources. For technical concerns or dataset accessibility issues, users can open detailed issue reports to receive support from maintainers. The project values open participation, making it easy for newcomers and experienced contributors alike to get involved and help strengthen the dataset collection.



feat: add GitHub Trending JSON dataset #226

karthik-248 wants to merge 1 commit into jdorfman:master from karthik-248:add-new-json-dataset

karthik-248 commented last week

This PR adds a new JSON dataset:

- **GitHub Trending JSON** — A JSON export of trending GitHub repositories.

The dataset can be used by developers to analyze trending projects.

feat: add GitHub Trending JSON dataset 9883ff2

jdorfman reviewed 2 days ago — View reviewed changes

```
README.md
83  83    * [Events](https://api.github.com/events)
84  84    * [Gists](https://api.github.com/gists)
85  85    * [Meta](https://api.github.com/meta)
    86  + * [JSON](https://gist.githubusercontent.com/karthik-248/some-gist.json)
```

jdorfman 2 days ago — Owner

This 404s

Reviewers: jdorfman

Requested changes must be addressed to merge this pull request.

Still in progress? Convert to draft

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues.

None yet

7.3 PR 3 : Y24 Open Source Engineering

7.3.1 Introduction and Purpose

This pull request introduces a self-hosted demo designed to guide users in installing and using the Kutt URL-shortening platform. The primary purpose is to make the self-hosting process accessible to a wider audience by providing instructions in the local language. This helps users understand how to run Kutt independently on their own systems without depending on cloud hosting services. By enabling localized documentation, the PR aims to improve usability, promote self-sufficiency, and encourage more contributors and learners to adopt open-source self-hosting practices.

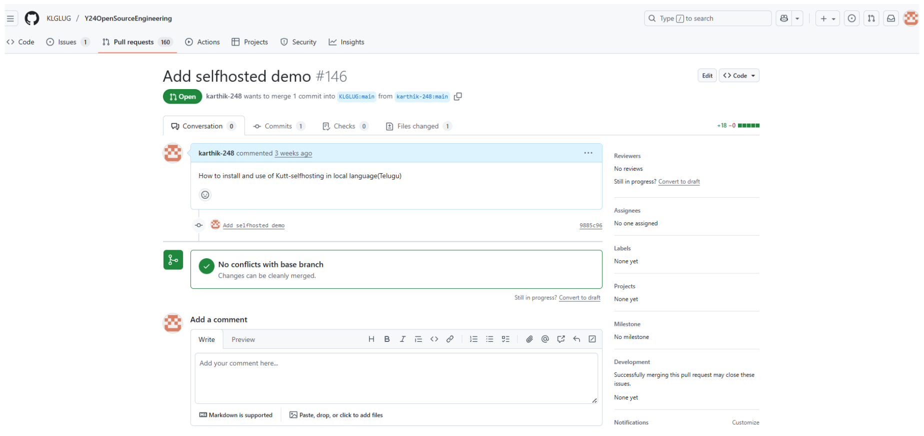
7.3.2 Technical Components

The self-hosted demo relies on a lightweight server environment capable of running the Kutt backend. Essential components include Node.js for backend execution, a database service such as PostgreSQL or SQLite for storing shortened links, and basic networking support to expose the service locally or publicly. Configuration files are used to set environment variables like API keys, database URLs, and domain settings. The setup can run on Linux, Windows, or macOS systems, and works effectively even on low-resource hardware, making it suitable for educational or small-scale deployments.

7.3.3 Operation and Usage

Once installed, the Kutt self-hosted instance runs as a local web service that users can access through a browser. Administrators manage the service by configuring environment variables, creating API keys, and supervising the link-generation workflow. The setup allows users to generate custom short

URLs, track link statistics, and manage their own private URL-shortening environment. It can be hosted locally for personal use or deployed publicly using a domain name. The demo provides hands-on experience in running a complete web application and showcases the practical benefits of self-hosting over relying on third-party services.



Add selfhosted demo #146

karthik-248 wants to merge 1 commit into KLGLUG:main from karthik-248:main

karthik-248 commented 3 weeks ago

How to install and use of Kutt-selfhosting in local language(Telugu)

Add selfhosted demo 9885c96

No conflicts with base branch
Changes can be cleanly merged.

Still in progress? Convert to draft

Add your comment here...

No reviews
Still in progress? Convert to draft
No one assigned
None yet
None yet
No milestone
Successfully merging this pull request may close these issues.
None yet

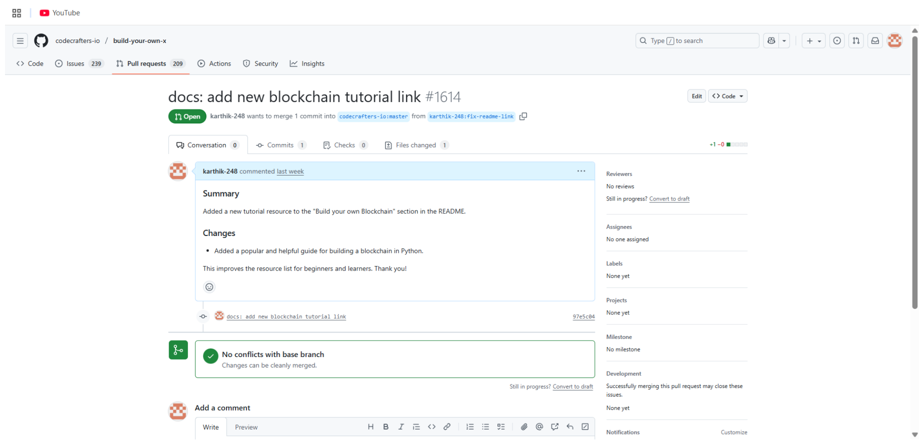
7.4 PR 4 : Fix/ Docs Formatting

7.4.1 The Issue (The Problem Being Fixed)

The main issue addressed by this Pull Request is the absence of a helpful learning resource within the "Build your own Blockchain" section of the documentation. The existing list lacked a beginner-friendly tutorial that clearly explains how to create a blockchain using Python, leaving a gap for learners looking for accessible guidance. This omission reduced the completeness of the resource list and limited the support available for newcomers exploring blockchain fundamentals. The PR is therefore categorized as a documentation enhancement rather than a functional modification, as it improves the quality and usefulness of the project's learning materials without affecting any code or breaking existing features.

7.4.2 The Solution (What Was Done)

The solution implemented in this Pull Request involves adding a new, widely used tutorial link to the blockchain section of the README. This tutorial provides a clear, practical introduction to building a blockchain using Python, making the section more valuable for beginners. The contributor updated the documentation by inserting the link in the appropriate section and ensuring it aligns with the project's formatting and contribution guidelines. Since the change only enhances the resource list and does not alter any executable code, the PR is safe, non-breaking, and ready for merging after review. The update ultimately improves the completeness and educational depth of the documentation.



docs: add new blockchain tutorial link #1614

karthik-248 wants to merge 1 commit into `codecrafters-io:master` from `karthik-248:fix-readme-link`

karthik-248 commented last week

Summary

Added a new tutorial resource to the "Build your own Blockchain" section in the README.

Changes

- Added a popular and helpful guide for building a blockchain in Python.

This improves the resource list for beginners and learners. Thank you!

`docs: add new blockchain tutorial link` 97e5c04

No conflicts with base branch
Changes can be cleanly merged.

Still in progress? Convert to draft

Add a comment

Write | Preview

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications | Customize

8 Linkedin Post Links

8.1 PR :

https://www.linkedin.com/posts/gnana-karthik-mada-b6984635b_over-the-past-few-weeks-ive-been-active
utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAFmlwwEB01dZRXd1Qosdlw-tcTW4nsUX7PY&
utm_campaign=copy_link

8.2 Journey Of Open Source :

https://www.linkedin.com/posts/gnana-karthik-mada-b6984635b_open-source-journey-activity-7398366504
utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAFmlwwEB01dZRXd1Qosdlw-tcTW4nsUX7PY&
utm_campaign=copy_link

8.3 Self Hosted Project :

https://www.linkedin.com/posts/harsha-sai-polnati-3281b6302_klu-projectexpo-selfhosted-ugcPost-7382
utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAFmlwwEB01dZRXd1Qosdlw-tcTW4nsUX7PY&
utm_campaign=copy_link