

М. О. СЛАБІНОГА

кандидат технічних наук,
доцент кафедри комп'ютерних систем і мереж
Івано-Франківський національний технічний університет нафти і газу
ORCID: 0000-0002-7296-0356

Т. В. БОЙЧУК

аспірант кафедри комп'ютерних систем і мереж
Івано-Франківський національний технічний університет нафти і газу
ORCID: 0009-0006-5058-4958

Д. О. ФЕЩАК

студент кафедри комп'ютерних систем і мереж
Івано-Франківський національний технічний університет нафти і газу
ORCID: 0009-0009-1536-4302

АВТОМАТИЗАЦІЯ ПРОЦЕСУ ПРИЙОМУ ТА ОПРАЦЮВАННЯ ЗАЯВОК ЗАСОБАМИ TELEGRAM-БОТА ТА ПРОГРАМНОГО ІНТЕРФЕЙСУ СИСТЕМИ OPENPROJECT

Стаття присвячена питанням автоматизації процесів технічної підтримки та впровадженню сучасних технологій для підвищення ефективності роботи таких систем. У дослідженні акцентується увага на важливості швидкого реагування на запити користувачів для забезпечення безперервної роботи IT-інфраструктури. Автори аналізують вплив сучасних рішень, таких як автоматизація процесів і застосування штучного інтелекту, на ефективність технічної підтримки та рівень задоволеності користувачів.

У статті розглядається система OpenProject, яка використовується для управління проектами та задачами в рамках технічної підтримки. В роботі детально описується API OpenProject, його функції та можливості інтеграції з іншими програмними рішеннями. API побудований на REST архітектурі та використовує стандартні HTTP методи, що дозволяє легко налаштовувати автоматизацію процесів, таких як створення, оновлення та управління проектами, завданнями, користувачами та файлами. Важлива увага приділяється безпеці, зокрема використанню API ключів та OAuth 2.0 для аутентифікації.

Однією з ключових тем статті є розробка телеграм-бота для автоматизації подачі заявок до відділу цифровізації та дистанційного навчання. Бот, створений на основі Python та бібліотек aiogram і ruopenproject, забезпечує інтеграцію з OpenProject для автоматичного створення та обробки заявок. У статті детально описується архітектура бота, його компоненти, зокрема інтерфейс користувача, бізнес-логіка, база даних та взаємодія з API Telegram.

Автори також наводять приклади різних типів заявок, які обробляються відділом цифровізації Івано-Франківського національного технічного університету нафти і газу, включаючи адміністрування сайтів, управління інтернет-з'єднанням та технічною підтримкою обладнання. У статті описані функції бота, що дозволяють спростити ці процеси, зокрема реєстрацію заявок через Telegram і їх автоматичне додавання в систему OpenProject.

У статті висвітлюється практична цінність розробки, яка полягає в автоматизації рутинних процесів, зменшенні навантаження на працівників та підвищенні точності виконання заявок. Автори зазначають, що розробка є гнучкою та масштабованою, її можна адаптувати для інших підрозділів і організацій із подібними потребами в автоматизації технічної підтримки.

Ключові слова: технічна підтримка, система опрацювання заявок, програмний інтерфейс, чатбот, Telegram, OpenProject.

М. О. SLABINOHА

Candidate of Technical Sciences,
Associate Professor at the Department of Computer Systems and Networks
Ivano-Frankivsk National Technical University of Oil and Gas
ORCID: 0000-0002-7296-0356

T. V. BOICHUK

Postgraduate Student at the Department of Computer Systems and Networks
Ivano-Frankivsk National Technical University of Oil and Gas
ORCID: 0009-0006-5058-4958

D. O. FESHCHAK

Student at the Department of Computer Systems and Networks
Ivano-Frankivsk National Technical University of Oil and Gas
ORCID: 0009-0009-1536-4302

AUTOMATION OF TECHNICAL SUPPORT RECEIVING AND PROCESSING APPLICATION PROCESS USING TELEGRAM BOT AND OPENPROJECT API

The article is devoted to issues of automation of technical support processes and introduction of modern technologies to improve the efficiency of such systems. The study emphasizes the importance of quick response to user requests to ensure continuous operation of the IT infrastructure. The authors analyze the impact of modern solutions, such as process automation and the use of artificial intelligence, on the effectiveness of technical support and the level of user satisfaction.

The article discusses the OpenProject system, which is used to manage projects and tasks within the framework of technical support. The work describes in detail the OpenProject API, its functions and the possibilities of integration with other software solutions. The API is built on the REST architecture and uses standard HTTP methods, which allows you to easily configure the automation of processes such as creating, updating and managing projects, tasks, users and files. Important attention is paid to security, in particular the use of API keys and OAuth 2.0 for authentication.

One of the key topics of the article is the development of a Telegram bot to automate the submission of applications to the digitization and distance learning department. Built on Python and the aiogram and pyopenproject libraries, the bot provides integration with OpenProject to automatically create and process requests. The article describes in detail the architecture of the bot, its components, including the user interface, business logic, database and interaction with the Telegram API.

The authors also provide examples of different types of applications that are processed by the digitalization department of the Ivano-Frankivsk National Technical University of Oil and Gas, including site administration, internet connection management, and equipment technical support. The article describes the functions of the bot, which allow you to simplify these processes, in particular, the registration of applications through Telegram and their automatic addition to the OpenProject system.

The article highlights the practical value of the development, which consists in automating routine processes, reducing the burden on employees and increasing the accuracy of application execution. The authors note that the development is flexible and scalable, and can be adapted for other departments and organizations with similar technical support automation needs.

Key words: technical support, application processing system, software interface, chatbot, Telegram, OpenProject.

Постановка проблеми

Ефективне опрацювання заявок технічною підтримкою дозволяє не тільки усунути технічні несправності, але й попередити їх появу в майбутньому. Завдяки цьому можна забезпечити стабільну роботу ІТ-інфраструктури, що є основою для досягнення стратегічних цілей організації. Важливим аспектом є також швидкість реагування на запити користувачів, адже затримка у вирішенні проблем може призвести до значних втрат для бізнесу. Окрім того, опрацювання заявок користувачів технічною підтримкою сприяє накопиченню знань та досвіду, що дозволяє вдосконалювати процеси обслуговування. Використання сучасних методів та технологій, таких як автоматизація процесів, бази знань та штучний інтелект, дозволяє значно підвищити ефективність роботи технічної підтримки.

Цифровізація закладів вищої освіти в Україні активно триває, і дедалі більше цифрових інструментів впроваджуються в навчальний та управлінський процес. У зв'язку з цим, зростає кількість заявок, пов'язаних із технічною підтримкою використання таких систем. Тому важливою задачею є автоматизація процесу прийому заявок для максимальної зручності користувача і оператора.

Аналіз останніх досліджень і публікацій

Питанню автоматизації технічної підтримки присвячена значна кількість публікацій та технічних рішень і продуктів. Зокрема, в роботі [1] розглядається вплив COVID-19 на освітній процес, зокрема, на застосування інструментів онлайн-освіти та технічної підтримки, пов'язаної з ними. Робота [2] фокусується на розробці системи технічної підтримки з використанням бази знань. Автори роботи [3] звертають увагу на важливість задоволення технічною підтримкою клієнтів банківської сфери. Роботи [4-5] розглядають актуальне питання використання штучного інтелекту та генеративних мовних моделей у процесі автоматизації технічної підтримки.

Формулювання мети дослідження

Метою роботи є розробка телеграм-бота як комунікаційного каналу між користувачами цифрових продуктів та системою OpenProject, що використовується технічною підтримкою відділу цифровізації та дистанційного навчання Івано-Франківського національного технічного університету нафти і газу для опрацювання заявок.

Викладення основного матеріалу

OpenProject – це популярна система управління проектами з відкритим вихідним кодом, яка надає потужні інструменти для планування, відстеження прогресу та співпраці над проектами. Система широко використовується

в різних галузях для ефективного управління завданнями, ресурсами та термінами. Зокрема, вона забезпечує можливості для створення проєктів, керування завданнями, відстеження термінів виконання, а також інтеграцію з іншими інструментами та сервісами.

Однією з головних переваг OpenProject є її API (Application Programming Interface), який дозволяє інтегрувати систему з іншими програмними рішеннями та автоматизувати різноманітні процеси. API OpenProject [6] надає широкий набір функцій для взаємодії з даними системи, що дозволяє розробникам створювати кастомізовані рішення для своїх потреб. Використання API дає можливість розширювати функціональність системи, інтегрувати її з іншими інструментами та забезпечувати автоматизацію рутинних задач.

API OpenProject базується на REST (Representational State Transfer) архітектурі, що робить його зручним та інтуїтивно зрозумілим для використання. REST API використовує стандартні HTTP методи (GET, POST, PUT, DELETE), що забезпечує легку інтеграцію з будь-якими веб-додатками та сервісами. Кожен ресурс в API має унікальний URL, за допомогою якого можна отримувати доступ до необхідних даних або виконувати різноманітні операції з ними.

Основні ресурси, доступні через API OpenProject, включають:

- проєкти (Projects) – створення, отримання, оновлення та видалення проєктів. API дозволяє керувати усіма аспектами проєкту, включаючи його назву, опис, терміни виконання та відповідальних осіб;
- завдання (Work Packages) – створення, отримання, оновлення та видалення завдань, а також управління їхніми атрибутами (пріоритет, статус, відповідальний, терміни виконання тощо). Це дозволяє ефективно керувати роботою команди та забезпечувати виконання завдань у встановлені терміни;
- користувачі (Users) – управління користувачами системи, включаючи створення нових користувачів, оновлення інформації про них та призначення ролей. API забезпечує можливості для адміністрування користувачів та контролю їх доступу до різних ресурсів системи;
- коментарі (Comments) – додавання та отримання коментарів до завдань, що забезпечує ефективну комунікацію всередині команди. Це дозволяє користувачам обмінюватися інформацією, коментарями та рекомендаціями безпосередньо у системі;
- вкладення (Attachments) – управління файлами, прикріпленими до завдань або проєктів. API дозволяє завантажувати та отримувати файли, що додаються до завдань, забезпечуючи зручність у роботі з документацією та іншими ресурсами.

Для доступу до API необхідно використовувати аутентифікацію, яка може бути реалізована за допомогою API ключів або OAuth 2.0. Це забезпечує безпеку даних та обмежує доступ до них лише для авторизованих користувачів та додатків. Аутентифікація дозволяє контролювати, які дії можуть виконувати користувачі та додатки, що забезпечує захист конфіденційної інформації та запобігання несанкціонованому доступу.

API OpenProject також забезпечує можливість інтеграції з інструментами для комунікації та співпраці, такими як Slack, Microsoft Teams, Zoom тощо. Це дозволяє забезпечити ефективну комунікацію всередині команди та оперативний обмін інформацією про стан виконання завдань. Наприклад, за допомогою телеграм-бота можна автоматично надсилати повідомлення про нові завдання або зміни їхнього статусу до відповідних каналів в Slack або Microsoft Teams, що забезпечує своєчасне інформування команди та підвищує ефективність комунікації.

Одним із найпоширеніших способів автоматизованої комунікації є використання чатботів у різноманітних месенджерах. У зв'язку з тим, що за даними опитування здобувачів вищої освіти та науково-педагогічних працівників ІФНТУНГ, найбільш поширеним месенджером для комунікації є Telegram, було прийняте рішення розробити бота для подачі заявок до відділу цифровізації та дистанційного навчання щодо здійснення технічної підтримки.

Відділ цифровізації та дистанційного навчання включає в себе декілька підрозділів, а саме:

- центр дистанційного навчання;
- інформаційно-обчислювальний центр;
- сектор взаємодії з ЄДЕБО.

До основних заявок, що надходять від користувачів до відділу цифровізації, відносяться:

- технічне адміністрування сайту університету;
- технічне адміністрування сайтів наукових журналів університету;
- призначення відповідальних адміністраторів окремих розділів сайту;
- опрацювання вхідної кореспонденції, пов'язаної з цифровою інфраструктурою університету;
- виконання стратегії цифровізації університету.

До основних заявок, що надходять від користувачів до центру дистанційного навчання, відносяться:

- заявки на отримання доступу до сайту;
- заявки на створення електронних курсів;
- заявки на сертифікацію електронних курсів;
- заявки на встановлення спеціалізованих плагінів.

До основних заявок, що надходять від користувачів до інформаційно-обчислювального центру, відносяться:

- заявки на підключення інтернету;
- заявки на проблеми з інтернетом;
- заявки на ремонт комп'ютерної техніки;
- заявки на придбання комп'ютерної техніки;
- заявки на встановлення програмного забезпечення;
- заявки на встановлення іншого обладнання (принтери, проектори, тощо);

До основних заявок, що надходять від користувачів до сектору взаємодії з ЄДЕБО відносяться:

- заявки на додавання нового користувача ЄДЕБО;
- заявки на зміну відповідального користувача ЄДЕБО.

Основними компонентами для реалізації чат-бота [7-8] були вибрані мова програмування Python, бібліотеки aiogram та ruoreproject, Docker для контейнеризації, а також база даних PostgreSQL для збереження даних.

Telegram-бот включає наступні компоненти:

- інтерфейс користувача (UI) – включає в себе інтерактивні повідомлення, які бот відправляє текстові повідомлення користувачам з інформацією про товари, статус замовлень тощо. Кнопки для швидкого доступу до основних функцій, які полегшують взаємодію з ботом, наприклад, кнопки «Поломки», «Інша проблема», «Допомога»;
- бізнес-логіка – включає в себе обробку запитів користувачів, які бот приймає та обробляє текстові команди і повідомлення від користувачів, такі як /start, /help та основний список проблем. Логіка обробки заявок, який включає функції для подачі заявок, збереження їх у базі даних і обробку статусів заявок;
- база даних (БД) – Зберігання інформації про заявки та їхній статус;
- інтеграція з Telegram API – включає в себе взаємодія з серверами Telegram для обробки повідомлень та команд, під час яких бот використовує Telegram API для отримання повідомлень від користувачів, відправки відповідей і керування взаємодією з користувачами Telegram.

Архітектура Telegram-бота подана на рисунку 1.

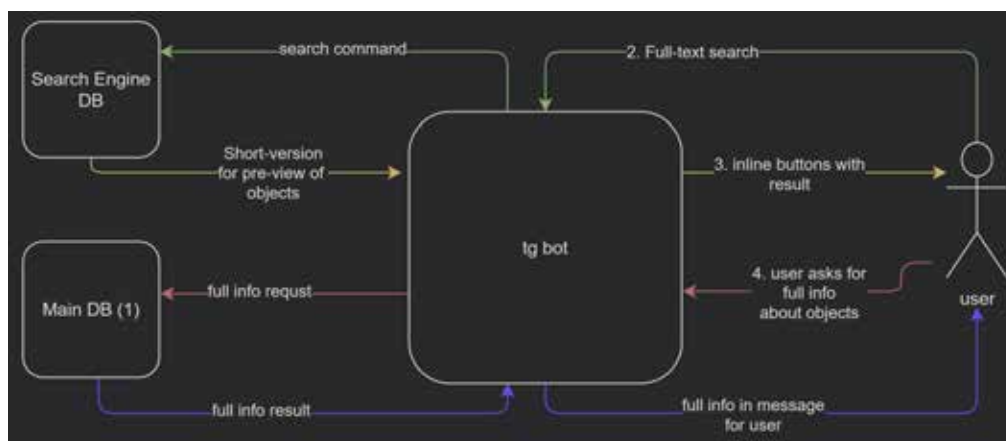


Рис. 1. Архітектура телеграм-бота

Файлова структура, представлена на рисунку 1, складається з декількох директорій та файлів, кожен з яких має свою функціональну роль у проєкті:

- директорія «database»: Відповідає за взаємодію з базою даних. Тут можуть зберігатися файли з моделями, скрипти для створення і оновлення таблиць, а також інші утиліти, пов'язані з базою даних;
- директорія «filters»: Містить файли, які визначають різні фільтри для повідомлень. Файл main.py ймовірно містить основну логіку для обробки фільтрів, а __init__.py робить можливим імпорт цього модуля;
- директорія «handlers» містить піддиректорії admin і user, які включають обробники для відповідних типів користувачів. Наприклад, main.py та other.py можуть містити функції для обробки команд та повідомлень від користувачів;
- Директорія «keyboards» включає файли для створення та управління клавіатурами бота. Наприклад, inline.py для інлайн-клавіатур, reply.py для клавіатур з відповідями;
- Директорія «misc»: Зберігає різні допоміжні скрипти і конфігураційні файли, такі як env.py для налаштувань середовища, util.py для утиліт;
- файл «entrypoint.sh»: Скрипт для запуску додатка в середовищі Docker.
- файл «run.py»: Основний файл для запуску бота.
- файл «.env.dev»: Файл для зберігання конфігурацій середовища під час розробки.

- docker-compose.yml: Файл конфігурації Docker Compose для визначення сервісів, мереж і томів для Docker-контейнерів.
 - Dockerfile: Файл конфігурації для створення Docker-образу.
 - requirements.txt: Файл, що містить список залежностей Python, необхідних для проекту.
- Приклад діалогового вікна розробленого телеграм-бота подано на рисунку 2.

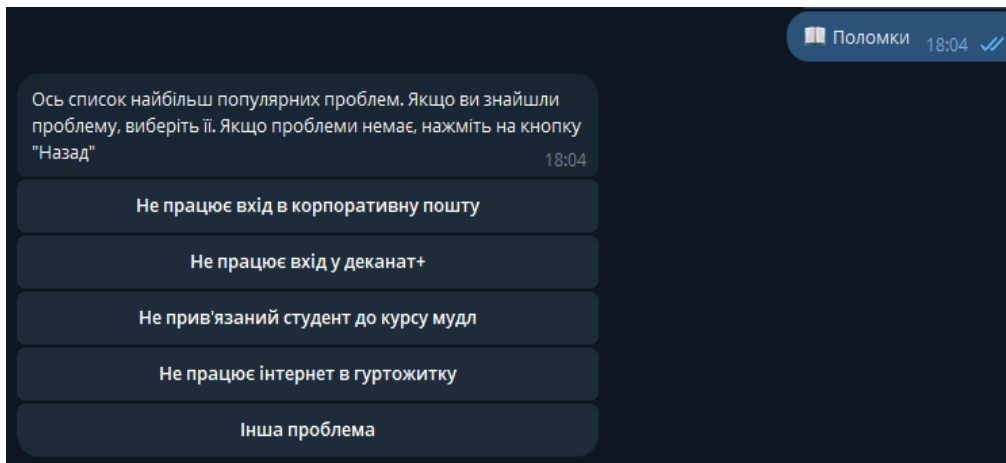


Рис. 2. Приклад взаємодії із телеграм-ботом

При виборі одного з пунктів, на екран виводиться текст про реєстрацію проблеми, а заявка з відповідним Telegram ID користувача передається у систему OpenProject. Приклад створеної заявки в OpenProject подано на рисунку 3.

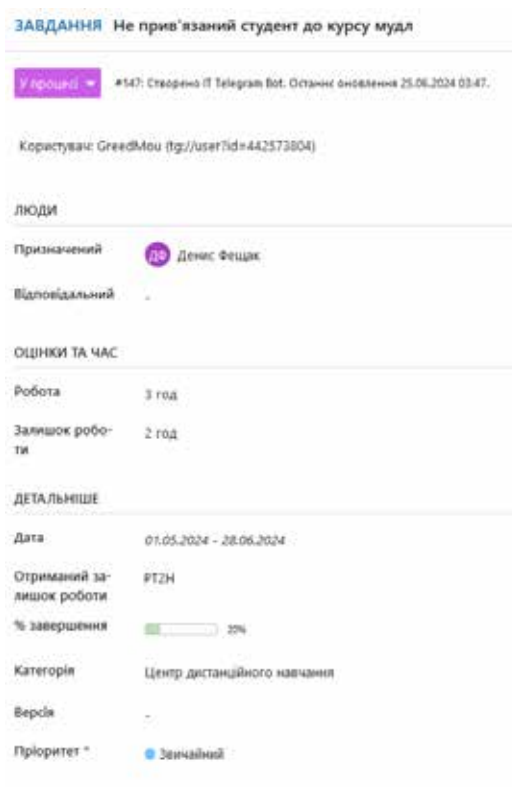


Рис. 3. Приклад взаємодії із телеграм-ботом

Таким чином, заявка потрапляє в систему технічної підтримки для подальшого опрацювання працівниками відділу цифровізації та дистанційного навчання.

Висновки

Практична цінність даної розробки полягає в декількох аспектах. По-перше, розробка телеграм-бота дозволяє значно підвищити ефективність роботи технічної підтримки, зменшуючи час на опрацювання заявок. По-друге, автоматизація процесів сприяє зниженню навантаження на персонал та зменшенню кількості помилок. По-третє, це покращує рівень задоволеності користувачів, що позитивно впливає на імідж організації.

Розроблений телеграм-бот може бути масштабований на інші підрозділи, які мають схожі потреби в автоматизації процесів технічної підтримки. Гнучкість і масштабованість рішення дозволяють легко адаптувати його до конкретних вимог будь-якої організації.

Подяка. Автори хочуть подякувати Збройним Силам України та всім захисникам України за можливість продовжувати наукову та технічну діяльність під час війни.

Список використаної літератури

1. Bao, Wei. (2020). COVID-19 and online teaching in higher education: A case study of Peking University. *Human Behavior and Emerging Technologies*. 2. 10.1002/hbe2.191.
2. Wicaksono, F., & Baswara, O. S. (2020). Design and Implementation of Web-Based Helpdesk Information Systems Using Extreme Programming Methods. *ITEJ (Information Technology Engineering Journals)*, 5(2), 88–96. <https://doi.org/10.24235/itej.v5i2.44>
3. Li, F., Lu, H., Hou, M., Cui, K., & Darbandi, M. (2021). Customer satisfaction with bank services: The role of cloud services, security, e-learning and service quality. In *Technology in Society* (Vol. 64, p. 101487). Elsevier BV. <https://doi.org/10.1016/j.techsoc.2020.101487>
4. Al-Hawari, F., & Barham, H. (2021). A machine learning based help desk system for IT service management. In *Journal of King Saud University – Computer and Information Sciences* (Vol. 33, Issue 6, pp. 702–718). Elsevier BV. <https://doi.org/10.1016/j.jksuci.2019.04.001>
5. Haleem, A., Javaid, M., & Singh, R. P. (2022). An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges. In *BenchCouncil Transactions on Benchmarks, Standards and Evaluations* (Vol. 2, Issue 4, p. 100089). Elsevier BV. <https://doi.org/10.1016/j.tbench.2023.100089>
6. OpenProject API // [Електронний ресурс]. URL:<https://www.openproject.org/docs/api/> (дата звернення: 01.05.2024)
7. Офіційний сайт Telegram API // [Електронний ресурс]: URL: <https://core.telegram.org/api>. (Дата звернення: 01.05.2024)
8. Medium // [Електронний ресурс]: URL: <https://servat.medium.com/creating-a-telegram-bot-with-python-4a3b4906c101> (Дата звернення: 02.05.2024)

References

1. Bao, Wei. (2020). COVID-19 and online teaching in higher education: A case study of Peking University. *Human Behavior and Emerging Technologies*. 2. 10.1002/hbe2.191.
2. Wicaksono, F., & Baswara, O. S. (2020). Design and Implementation of Web-Based Helpdesk Information Systems Using Extreme Programming Methods. *ITEJ (Information Technology Engineering Journals)*, 5(2), 88–96. <https://doi.org/10.24235/itej.v5i2.44>
3. Li, F., Lu, H., Hou, M., Cui, K., & Darbandi, M. (2021). Customer satisfaction with bank services: The role of cloud services, security, e-learning and service quality. In *Technology in Society* (Vol. 64, p. 101487). Elsevier BV. <https://doi.org/10.1016/j.techsoc.2020.101487>
4. Al-Hawari, F., & Barham, H. (2021). A machine learning based help desk system for IT service management. In *Journal of King Saud University – Computer and Information Sciences* (Vol. 33, Issue 6, pp. 702–718). Elsevier BV. <https://doi.org/10.1016/j.jksuci.2019.04.001>
5. Haleem, A., Javaid, M., & Singh, R. P. (2022). An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges. In *BenchCouncil Transactions on Benchmarks, Standards and Evaluations* (Vol. 2, Issue 4, p. 100089). Elsevier BV. <https://doi.org/10.1016/j.tbench.2023.100089>
6. *API documentation*. OpenProject.org. (01.05.2024). <https://www.openproject.org/docs/api/>
7. *Telegram apis*. Telegram APIs. (01.05.2024). <https://core.telegram.org/api>
8. Servat, A. (2024, February 18). *Creating a telegram bot with python*. Medium. <https://servat.medium.com/creating-a-telegram-bot-with-python-4a3b4906c101>