

SDK Implementation Experiment Brief

Peneliti : Alza Syauqi Billah Ghifari (23/528902/PTK/15592)
Program Studi : Magister Teknologi Informasi 2023 Genap
Departemen : Departemen Teknik Elektro dan Teknologi Informasi
Fakultas : Fakultas Teknik
Universitas : Universitas Gadjah Mada

Penelitian ini meneliti tentang bagaimana dampak penggunaan full-stack SDK terhadap produktivitas dan praktik pengembangan oleh developer, khususnya dalam mengembangkan code-base backend. Adapun kriteria responden yang dapat mengikuti eksperimen ini adalah:

1. Pernah/sedang bekerja ataupun magang sebagai developer setidaknya mempunyai pengalaman serta memahami basic tentang backend development
2. Belum pernah atau minim pengalaman dalam mengembangkan proyek menggunakan Python dan Django pada backend development

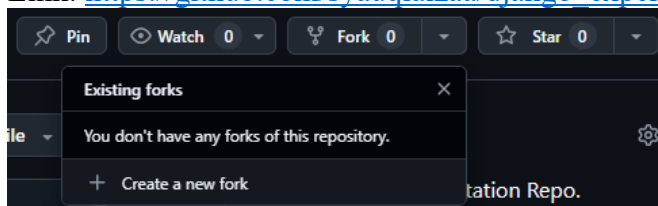
Responden akan diminta menyelesaikan 3 brief/instruksi dengan 2 kali implementasi, yaitu yaitu menggunakan vanilla Python (native) atau micro-framework dan Django sebagai full-stack SDK (bukan DRF (Django Rest Framework)), sehingga total coding yang dilakukan yakni sebanyak 6 kali.

1. Prosedur Eksperimen

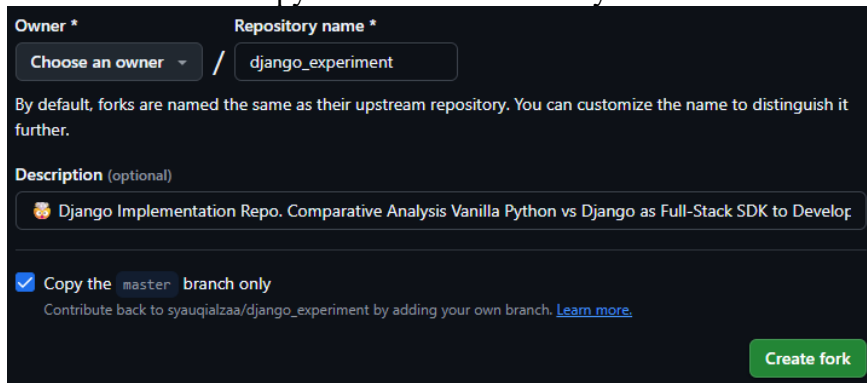
Dilakukan beberapa langkah atau prosedur untuk melakukan eksperimen ini. Pastikan Anda sudah memiliki akun GitHub. Berikut merupakan langkah-langkahnya:

1. Responden forking repo

Link: https://github.com/syauqialzaa/django_experiment.git

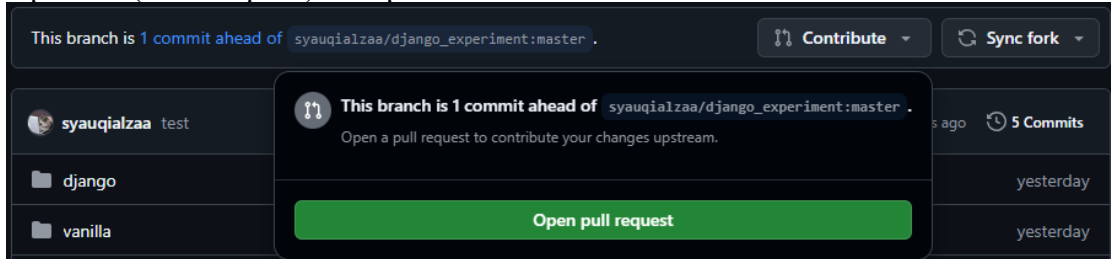


2. Ceklis checkbox 'Copy the master branch only' lalu create fork

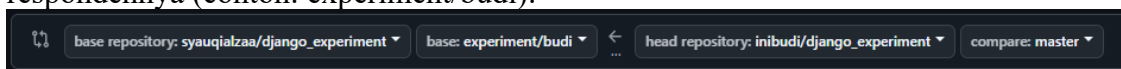


3. Clone forked repo yang sudah dilakukan

4. Install dependencies seperti Python v3.10^ (jika belum diinstal)
5. Buat virtual environment pada root directory
6. Install Django (bukan DRF atau Django Rest Framework) v5^
7. Lakukan coding sesuai brief Vanilla dan Django
8. Setelah selesai coding, commit dan push ke repo
9. Open PR (Pull Request) ke repo asli



10. Lakukan PR dengan target branch yang sudah disiapkan berdasarkan nama respondennya (contoh: experiment/budi):



11. Setelah sesi coding selesai, jawab beberapa pertanyaan pada google form
Link: <https://forms.gle/pCPaNHKh6HzDbdHu9>

Selama melakukan eksperimen, responden diperbolehkan melihat contoh implementasi kode, dokumentasi, dan browsing di internet.

2. Briefs

Aplikasi yang akan dibuat bernama HealthLinkr, yaitu aplikasi untuk manajemen appointment untuk health services. User bisa melakukan penjadwalan appointment kepada dokter pada fasilitas kesehatan setempat.

Brief 1 – Authentication & Authorization API (Vanilla Python dan Django)

Buatlah endpoint API untuk fitur Authentication & Authorization.

- GET /api/login/ untuk login user dengan username dan password.
- POST /api/signup/ untuk registrasi user (username, email, password, role)
- GET /api/user/{user_id}/ untuk mendapatkan data user tertentu (username, email, role).
- PUT /api/user/{user_id}/update untuk update data user
- DELETE /api/user/{user_id}/update untuk menghapus data user

User memiliki 3 roles, yaitu; patient, doctor, dan administrator

Brief 2 – Service Management API (Vanilla Python dan Django)

Buatlah endpoint API untuk mengelola Service di sebuah fasilitas kesehatan.

- GET /api/services/ untuk menampilkan daftar semua services beserta nama klinik/rumah sakit dan dokter yang tersedia
- POST /api/services/ untuk menambahkan services baru

- GET /api/services/{service_id}/ untuk menampilkan detail service
- PUT /api/services/{service_id}/ untuk update service
- DELETE /api/services/{service_id}/ untuk menghapus service

Brief 3 – Appointment Booking & Management API (Vanilla Python dan Django)

Buatlah endpoint API untuk pasien agar bisa melakukan booking appointment.

- POST /api/appointments/book/ untuk melakukan appointment dengan dengan jadwal yang tersedia
- POST /api/appointments/{appointment_id}/cancel/ untuk membatalkan appointment (contohnya: status berubah jadi cancelled dan slot menjadi available kembali).
- GET /api/appointments/ untuk menampilkan semua appointment

3. Notes

1. Models atau entitas yang dibutuhkan, relasi, serta properties-nya tidak dibatasi (bebas)
2. Responden diminta untuk fokus ke pengembangan API, apabila ingin mengembangkannya lebih dari itu (karena Django merupakan full-stack SDK), juga diperbolehkan
3. Database yang digunakan juga bebas (SQL, No-SQL, dll.)
4. Kapabilitas fitur dan aplikasi dapat ditambahkan sesuai keinginan responden jika ingin mengeksplorasi lebih lanjut (testing, data validation, dll)