

Kontribuutiot

Tämä osio dokumentoi projektin hakemistorakenteen ja siihen osallistuneiden henkilöiden vastualueet. Jos joku osa-alue on kokonaan yhden henkilön tekemä, sen kansioita/tiedostoja ei tässä erikseen listata.

[Eprice/Documents/](#) kansioista löytyy yksityiskohtainen dokumentaatio koko projektista.

```
| - projektinhallinta "juuren" tasolla, "full stack plus", (Paavo)
|
| - LICENSE
| - README.md
| - App
|   | - README.md
|   | - compose.yaml - konttien hallinta (Paavo)
|   | - project.env - projektin ja konttien konfiguraatiot (Paavo)
|   |
|   | - backend-tests - testauksen alustus (Paavo)
|   |   | - Dockerfile
|   |   | - README.md
|   |   | - tests
|   |     | - test_auth_controller.py - (Juho & Paavo)
|   |     | - test_data_controller.py - (Juho)
|   |
|   | - chat-engine - (Developer chat, Paavo)
|   |
|   | - client - käyttöliittymä (Paavo)
|   |
|   | - data-preparation - datan esikäsittely ja lataus (Paavo)
|   |
|   | - database-migrations - tietokantamuutokset (Paavo)
|   |   | - V1-V12__*.sql - eri tietokantaversiot ja skeemamuutokset
|   |   |   Huom. V7 puuttuu versionointivirheen takia
|   |
|   | - e2e-tests - kokonaisuuden testauksen alustus (Paavo, mutta en ole
|   |   ehtinyt vielä kirjoittaa juurikaan testejä)
|   |
|   | - python-server (Juho & Paavo)
|   |   | - Dockerfile
|   |   | - README.md
|   |   | - main.py
|   |   | - requirements.txt
|   |   |
|   |   | - config - konfiguraatiot ja salaisuudet
|   |   |   | - __init__.py
|   |   |   | - secrets.py
|   |   |
|   |   | - controllers - autentikointi- ja datakontrollerit
|   |   |   | - auth_controller.py (Juho & Paavo)
|   |   |   | - data_controller.py (Juho)
|   |   |
|   |   | - ext_apis - ulkoiset rajapintakutsut
```

```

├── ext_apis.py (Juho)
├── models - tietomallit ja poikkeukset
│   ├── custom_exception.py (Juho)
│   ├── data_model.py (Juho)
│   └── user_model.py (Paavo)
├── repositories - tietokantakerros
│   ├── porssisahko_repository.py (Paavo)
│   └── user_repository.py (Paavo)
├── scheduled_tasks - ajoitetut tehtävät (esim. sähkönhintadatan
ajantasaisuus)
│   └── porssisahko_scheduler.py (Paavo)
├── services - logiikka reitityksen (controllers) ja tietokannan
(repositories) / ulkoisten apien välillä
│   ├── auth_service.py (Paavo)
│   └── data_service.py (Juho)
├── utils - apufunktioita yms.
│   ├── email_tools.py (Paavo)
│   ├── porssisahko_service_tools.py (Juho)
│   └── porssisahko_tools.py (Paavo)
├── user-chat - käyttäjä-chat (alustus, Paavo, Juho hioo systeemi
promptia sikäli kuin ehtii)
└── Documents - projektin dokumentaatio
    ├── README.md
    ├── backend_design.md
    ├── openapi_endpoint_descriptions.md
    ├── project_description.md
    ├── project_directory_structure.txt
    ├── api_definitions
    │   ├── openapi.json
    │   └── README.md
    └── diagrams
        └── sources - PlantUML-muotoiset kaaviot
            ├── authentication_call_sequence_diagram.wsd
            ├── authentication_class_diagram.wsd
            ├── authentication_use_case.wsd
            ├── data_access_call_sequence_diagram.wsd
            ├── llm_retrieval.wsd
            ├── services_diagram.wsd
            ├── tool_calling.wsd
            └── use_case.wsd

```

Notebooks hakemisto (tavallaan extraa)

Developer chatin retrieval toiminto vaatii tietyn määrän esikäsittelyä (parserointi / siivous / lataus), joka tehdään offline-tilassa skripteillä ja tiedostolla [Eprice/Notebooks/document_loading.ipynb](#). Nämä perustuvat osittain Paavon (heh, siis minun) aiempiin ja rinnakkaisiin projekteihin. Kaikki ei ole vielä mukana tässä repossa, mutta tullaan julkaisemaan myöhemmin osoitteessa <https://github.com/PaavoReinikka>.

Projektin dokumentoinnin avuksi on koodattu myös UML-kaavioiden generointityökalu, joka käyttää PlantUML-syntaksia ja on toteutettu Pythonilla (lisätietoja Notebooks README:ssä).

Käyttäjächat / User-Chat

User-Chat perustuu Ollaman tarjoamaan avoimeen Llama 3.2 -malliin. Chat toimii lähinnä esimerkkinä siitä, että avoimen lähdekoodin kielimalleja voidaan käyttää myös kontitetyssä sovelluksessa. Hyvän suorituskyvyn saavuttaminen edellyttää kuitenkin hienosäätöä ja/tai suurempaa mallia. CUDA-yhteensopiva GPU ei ole välttämätön, ellei mallia kouluteta — pienemmät mallit toimivat hyvin myös moderneilla prosessoreilla.

Projektin aikarajoitteiden (n. 4–6 viikkoa) vuoksi mallin hienosäätö ei ollut tällä kertaa mahdollista. Myös datan kerääminen ja valmisteleva työ vie merkittävästi aikaa.

Kehittäjächat / Developer-Chat

[Eprice/App/chat-engine](#) kontissa ajettava developer chat käyttää OpenAI:n gpt-4o-mini -mallia, joka on käytännössä ilmainen tutkimus- ja kehityskäyttöön. Projektiin liittyy myös kokeiluja useiden avoimen lähdekoodin generatiivisten mallien kanssa (sekä chatti- että completion-malleilla), jotka julkaistaan myöhemmin osoitteessa <https://github.com/PaavoReinikka>.

Embeddaukseen käytetty malli on avoin malli HuggingFacesta. Sieltä löytyy useita laadukkaita vaihtoehtoja, ja hyvä aloituspiste on MTEB Leaderboard: <https://huggingface.co/spaces/mteb/leaderboard>.

Dokumentaatio

Dokumentaatio on pääosin Paavo Reinikan tekemää. Juho Ahopelto dokumentoi serverin endpointit (openapi endpoints).