
ViesTIM-sovellusprojekti, toinen katselmointi

Aika: 14.5.2021 klo 8.00-9.52

Paikka: Zoom

Läsnäolijat

Hannamari Heiniluoma

Kristian Käyhty

Mika Lehtinen

Tomi Lundberg

Jukka-Pekka Santanen

Denis Zhidkikh

Tuuli Veini (sihteeri)

Muistio

Laadittu: 16.5.2021

Muokattu: 20.5.2021

1. GitLab-kommentit

Zhidkikh kävi katselmoinnissa läpi pääasiassa GitLabiin laittamiensa kommenttien [1] sisältöä. Ryhmän koodiin laittamia `TODO/VIESTIM`-merkintöjä ei juuri käsitelty. Ne kannattaa kuitenkin käydä läpi ja joko toteuttaa ne koodissa tai kirjata ne jatkokehitykseen.

Zhidkikh esitti kommentteista seuraavia tarkentavia huomioita:

- Kokonaisuutena koodin dokumentaatio on ihan hyvällä tasolla, mutta niistä kannattaa korjata kielioppi- ja kirjoitusvirheet. Lisäksi dokumentaatio tulee

täydentää puuttuviin kohtiin sekä tarkistaa, että se on ajan tasalla sellaisissa funktioissa, joissa esim. parametrien tyypit ovat muuttuneet.

- Mailman-clientin olemassaoloa tarkistetaan useassa kohdassa `if _client is None` -lauseella. Lauseita kannattaa yksinkertaistaa ja yhteinäistää luomalla jonkinlainen `verify_mailman`-funktio, jolla tarkistetaan clientin olemassaolo samaan tapaan kuin esim. käyttäjältä tarkistetaan TIMissä sivun `edit`-oikeus. Funktiossa on parempi heittää poikkeus kuin palata `return`-lauseella. [2]
- Zhidkikh kysyi, onko viestilistoille tarkoitus toteuttaa projektin aikana `unfreeze-list`-funktio. Asioita, joita projektissa ei toteuteta, kannattaa yleensä siivota koodista pois. Tässä tapauksessa funktion tynkä kannattaa kuitenkin jättää, koska jäädytys ei ole ihan triviaalia ja puolet siitä on jo tehty. Hyvä tapa käsitellä tynkäfunktio on heittää `NotImplemented`-poikkeus ja lisätä sen yhteyteen viesti, miksi se ei ole vielä toteutettu. [3]
- Zhidkikh suositteli `try-except`-lohkojen refaktorointia yksinkertaisemmaksi. [4] Samassa yhteydessä käsiteltiin `delete_email_list`-funktion virheiden käsittelyä ja tarkentamista. Mailmanin dokumentaatio suosittelee virheeksi `HTTPError` ia. Eri vaihtoehtoja virheen käsittelyyn pohdittiin ja lopuksi todettiin, että toistaiseksi `HTTPError` it voi heittää sellaisenaan ja niiden toimimista testataan kesällä. Jonkinlainen virheviesti olisi tärkeää saada myös ylläpitäjille, etteivät virhetilanteet jää vain käyttäjien tietoon. Lundberg selvittää, onko mahdollista määrittää milloin näytetään käyttäjälle virhe ja milloin lähetetään ylläpitäjille `wuff`.
- Zhidkikh kysyi Lehtiseltä, lokitetaanko `RouteException` - ja `NotExist` -poikkeukset. Lehtisen mukaan `RouteException` ia ei lokiteta. Jos virheet lokitetaan, koodiin ei tarvitse erikseen kirjoittaa auki mistä virhe syntyy. [5]
- `verify_mailman`-tarkistus kannattaa lisätä myös `get_email_list_by_name`-funktioon, ellei ole täysin varmaa, että sitä kutsutaan aina sellaisista paikoista, joissa tarkastus on jo tehty. [6]
- Sähköpostiosoitteen tarkistus päätettiin siirtää jatkokehitykseen. [7]
- Koodissa on joitain funktioita, kuten `get_email_list_member`, jotka ei tee muuta kuin kutsuu toista funktiota. Niiden kohdalla kannattaa miettiä, onko siitä mitään hyötyä. [8]
- Viestilistan nimen tarkistusfunktio `check_messagelist_name_requirements` kannattaa nimetä uudelleen. TIMin tyyli on, että tarkistuksia tekevät funktiot alkavat `verify`-sanalla; `check`iä käytetään `true/false`-tapauksissa. [9]
- `MessageTIMversalis`-luokan nimeksi ehdotettiin mm. `MessageContent` ja `BaseTimMessage`. Lisäksi nimeämistyylin mukainen muoto alkuperäiselle nimelle

olisi `MessageTIMVersalis`. [10]

- `messagelist_utils` -moduulissa mainittua vastaussyvyyden toteutusta tuskin ehditään tekemään projektin aikana. Siihen liittyy heuristiikkaa, joka on melko työläs koodata ja testata. [11]
- `tim-message-list-admin` -komponentti toimii nyt niin, että se selvittää missä dokumentissa ollaan tällä hetkellä ja sen perusteella selvittää, mitä viestilistaa hallitaan. Jotta viestilistaa voisi hallita muualtakin, komponentin valinnaiseksi parametriksi voisi laittaa hallintadokumentin `id:n`. Jos komponentti laitetaan sellaisenaan ilman parametria, voisi olettaa, että ollaan hallintadokumentissa. Kommentoituun koodin kohtaan ei ole kuitenkaan pakko tehdä muutoksia. [12]
- `check_group_owner_or_manage_right` on sama kuin `has_manage_access`. Kun viestilistojen luominen sallitaan muillekin kun admineille, oikeuden tarkistuksen yhteyteen täytyy dokumentoida selkeästi, että ketä tahansa ei saa lisätä sähköpostilistaan. [13]
- `get_tim_messages_as_list` -funktiossa voisi olla hyvä tarkistaa, onko käyttäjällä katseluoikeus sivulle. Mitä jos käyttäjä menee sivulle, jossa tällä ei ole katseluoikeutta, mutta siellä näytetään kaikille sivun katselijoille joku viesti? Koodiin täytyy vähintään kirjoittaa joku huomio, että tarkastus on jätetty tahallaan tekemättä, koska projektissa globaaleja viestejä ei ole toteutettu. [14]
- TIM-viestien luomisesta kannattaa poistaa turha tietokannan `flush`. `create_message_displays` -funktiossa ei tarvitse tietää tietokantaan tulevaa viestin `id:tä`, vaan siinä voi hyödyntää SQLAlchemyn suhteiden käsittelyä. `msg_id:n` sijaan voi antaa suoraan `InternalMessage` -objektin, jolloin SQLAlchemy asettaa oikeat `id:t`, kun suoritetaan `db.session.commit()`. Sama kannattaa tehdä muille vastaaville kohdille. Lisäksi ei tarvitse luoda ensin objektia, vaan attribuutit voi suoraan listata parametreina. [15], [16] ja [17]
- `send_message_or_reply` -funktiossa kannattaa miettiä, tarvitseeko tehdä muita oikeuksien tarkistuksia kuin sisään kirjautuminen. Nyt kuka tahansa periaatteessa voisi lähettää minne tahansa viestejä. Heiniluoma huomautti, että `url_check` tarkistaa, onko käyttäjällä oikeus annetulle sivulle `edit`-oikeus. Zhidkikh totesi, että viestin lähettämistä ei-sallituille sivuille kannattaa silti testata jollain tapaa. Ryhmä kommentoi kohtaa GitLabiin, kun toiminnallisuudelle on tehty yksikkötestejä tai sen testaamisesta on muuten päätetty. [18]
- `message-list-admin` -komponentin viestilistan tietojen lataaminen kannattaa toteuttaa asynkronisilla metodeilla. Siten varmistetaan, että koodissa ei edetä ennen kuin edellinen HTTP-kutsu on suoritettu loppuun. Nyt käytetty `timeout` on epävarma toteutustapa, koska se voi olla liian lyhyt aika hitaammille nettiyhteyksille.

Myös `ngOnInit` -funktioista voi tehdä suoraan asynkronisen. [19]

- `tim-message-view` -komponentissa listalle parempi syntaksi on `TimMessageData[] . undefined` -määrite tulee korvata kysymysmerkillä. [20].
Samoin `tim-message` -komponentissa `undefined in` sijaan tulisi käyttää kysymysmerkkiä. Tilanne, jossa viestiä ei ole määritelty, tulee käsitellä jotenkin. [21]

2. Muut kommentit ja kysymykset

Zhidkikh kommentoi ryhmän saaneen hyvin koodia aikaan ja koodaustyylin parantuneen TIMin suuntaiseksi edellisen katselmoinnin jälkeen. Koko ryhmä on päässyt kokeilemaan komponenttien kirjoittamista ja tietokannan käsittelyä. Koodia on syntynyt noin 5000 riviä, mikä on hyvän kokoinen projekti.

Lundberg kysyi tarkennusta Python-koodin kommentointiin. `None` -paluuarvoa ei tarvitse kommentoida. `raises` -poikkeukset voi kommentoida, jos selitys ei mene suoraan käyttäjälle tai ole muuten itsestäänselvä. Tässä projektissa niitä ei todennäköisesti tarvitse kommentoida.

Jos Mailmanin kanssa joutuu "kikkailemaan" koodissa, kannattaa dokumentoida hyvin miksi niin tehdään.

Santanen suositteli luokkadokumenttien ajamista, jolloin ryhmän jäsenet pystyvät arvioimaan toistensa kirjoittaman dokumentaation selkeyttä pelkkien kommenttien perusteella. Mahdollisia luokkadokumenttiohjelmia on ainakin `pydoc` ja `Doxygen`.

Veini kysyi tarkennusta koodin lopulliseen hyväksymiseen. Santanen vastasi, että ryhmä julkistaa koodin tarkastettavaksi siinä vaiheessa, kun siihen ei tarvitse tehdä muuta kuin aivan välttämättömiä korjauksia. Sovittiin, ettei Lehtinen osallistu koodin lopulliseen hyväksymiseen, vaan Zhidkikh hoitaa asian. Zhidkikhillä koodin tarkistamisessa voi mennä kesäkuun puolelle riippuen siitä milloin ryhmä julkistaa sen.

3. Päätökset ja sovitut toimenpiteet

- Ryhmä korjaa koodia annettujen kommenttien perusteella.
- Kun Zhidkikhin GitLab-komentissa esittämät korjausehdotukset on tehty ja mergetty `dev` -haaraan, ryhmä kommentoi GitLab-komenttia ja Zhidkikh merkitsee sen tilaksi *resolved*.
- Zhidkikh vastaa koodin lopullisesta hyväksymisestä.

Tarkastellut ja jaetut materiaalit

[1] ViesTIM projekti, Merge Request

[2-21] sisältyvät materiaaliin [1]