

T1.1

a)

- Rajapinnat ovat tiedon välittäjiä
- Rajapintoja tarvitaan, että järjestelmät toimisivat paremmin yhdessä
- Tässä vaiheessa mitä tajuan, tästä on se, että rajapinta helpottaa tiedon kulkua järjestelmien välillä, esim UI:n ja Backendin.

b)

- valitsin catfact rajapinnan, vaitsin tämän, koska se vaikutti typerältä
- catfact rajapinta tarjoaa satunnaisia kissafaktoja, mahdollistaa niiden jakamisen eri alustoilla, kuten tekstiviesteinä, Twitterissä ja muiden käyttäjien kanssa keskusteluina. Se tarjoaa myös keinoja hallita ja mukauttaa kissafaktan lähettämistä ja vastaanottamista.

c)

Suomi.fi siellä on viestit kohta, jossa voi vastaanottaa viranomaisilta tärkeitä viestejä, kuten verotuksen tai lupahakemusten tilapäivityksiä. Asiointi, jonka kautta voidaan tehdä viranomaisasiointia verkkosivustojen kautta, kuten hakea lupia, ilmoittaa tietoja tai katsella omia tietoja.

T1.2

a)

API on liittymä, joka auttaa ohjelmia kommunikoimaan tehokkaammin.

b)

Osoite, jonka avulla voit käyttää API:ta ja sen eri ominaisuuksia.

c)

API-parametrit ovat vaihtoehtoja, jotka voidaan välittää päätepisteen (endpoint) mukana vaikuttamaan vastaukseen.

d)

API-päätepiste (Endpoint) on API:n tietty "sisääntulopiste" ja yksi tärkeimmistä osista API:n dokumentaatiota.

e)

Yksilöllinen tunniste, jonka avulla palvelun käyttäjä (esimerkiksi kehittäjä tai sovellus) voi todistaa oikeutensa käyttää API:a. Se toimii kuin salasana, jonka avulla API tunnistaa ja autentikoi käyttäjän tai sovelluksen.

f)

API-otsakkeet (headers) ovat kuin lisätietolähde jokaiselle API-kutsulle, jonka teet.

g)

GET-pyyntö on yksinkertaisesti tapa, jolla voit hakea tietoja tietolähteestä internetin avulla.

h)

A POST request, in simple terms, is a way for you to send data to a destination with the help of the internet.

T1.3

Json on JavaScript Object Notation, sitä käytetään tietojen tallentamiseen ja siirtämiseen. Jsonia käytetään erityisesti verkkosovelluksissa ja API-yhteyksissä. Se on kielettömän ja ihmisten luettavissa olevan rakenteen ansiosta erittäin suosittu tietojen välittämiseen palvelimien ja asiakasohjelmien välillä.

JSON kehittyi JavaScriptin sisäisestä objektimuodosta ja tuli virallisesti suosituksi tiedonsiirtomuodoksi web-teknologioiden kehittyessä. Vaikka JSON on peräisin JavaScriptistä, sitä tukevat lähes kaikki ohjelmointikielet, kuten Python, Java, C#, Ruby jne.

C#:ssa JSON:n käsittelyyn käytetään yleisesti kirjastoja kuten **Newtonsoft.Json** tai **System.Text.Json** (uudempi). **Newtonsoft.Json** on erittäin suosittu ja tarjoaa laajan tuen JSON-objektien käsittelyyn.

