

# Ohjelmointirajapinta

Ohjelmointirajapinta, eli API (Application Programming Interface) voidaan määritellä joukoksi sääntöjä ja protokollia, joiden avulla tietokoneohjelmat voi kommunikoida keskenään. Rajapinnat voivat olla eritasoisia, mutta niiden perimmäinen tarkoitus on tarjota toiminnallisuutta ohjelmoijalle ilman, että tämän tarvitsee sukeltaa ohjelmiston syvimpään tekniseen rakenteeseen.

Ohjelmointirajapintoja käytetään, koska niiden avulla ohjelmoija voi käyttää valmiiksi tehtyjä toimintoja ilman, että hänen tarvitsee kehittää niitä alusta asti, mikä säästää aikaa ja resursseja. Rajapinnat mahdollistavat myös järjestelmien jakamisen pienempiin osiin, mikä tekee ohjelmien kehittämisestä ja ylläpidosta selkeämpää ja helpompaa. Eri ohjelmistot ja palvelut voivat integroitua toisiinsa API:n kautta. Esimerkiksi verkkosivusto voi käyttää rajapintaa noutaakseen ja näyttääkseen säätietoja toiselta palvelulta. Rajapinnat ovat myös tärkeitä turvallisuuden kannalta sillä ne antavat tavan kontrolloida, mitä tietoja ja toimintoja toisille ohjelmille tai palveluille annetaan käyttöön.

## Meme Generator -rajapinta

Meme Generator API on erityinen ohjelmointirajapinta, jota käytetään meemien luomiseen ja muokkaamiseen ohjelmallisesti. Se tarjoaa valmiita toimintoja, joiden avulla käyttäjät tai ohjelmat voivat valita valmiita meemipohjia, lisätä tekstiä ja luoda omia meemejä.

Valitsin Meme Generator API:n, koska se tekee työskentelystä hauskaa ja mahdollistaa luovien meemien luomisen ja muokkaamisen. Harjoitustyö voisi yhdistää koodauksen ja huumorin, mikä tekee projektista motivoivan ja viihdyttävän.

Meme Generator -rajapinta on suhteellisen yksinkertainen ja keskittyy pääasiassa meemeihin liittyviin toimintoihin. Se on tarkoitettu kevyisiin ja viihteellisiin sovelluksiin, kuten sosiaalisen median integraatioihin, mutta tarjoaa laajan joustavuuden meemien luomisen ja muokkaamisen osalta.

Meme Generator -rajapinnan avulla voit hakea suosituimpia meemipohjia palvelun tietokannasta. Sen kautta voidaan lisätä tekstiä valituille meemipohjille sekä määrittää tekstin paikka, fontti ja koko. Voit myös ladata omia kuvia meemin pohjaksi ja lisätä niihin tekstiä.

Meme Generator API olisi mielenkiintoinen harjoitustyön aihe, koska sen kanssa pääsee harjoittelemaan API:n käyttöönottoa, rajapinnan kautta datan hakua ja sen prosessointia. Lisäksi rajapinnan käyttöönotto on suhteellisen yksinkertaista, joten se sopii hyvin harjoitustyöhön riippumatta siitä, oletko vasta aloittamassa API:n käyttöä tai jos haluat syventää osaamistasi.

# Kysymykset:

## a) Mikä on API?

API on ohjelmointirajapinta, joka mahdollistaa erilaisten ohjelmien tai palveluiden vuorovaikutuksen toistensa kanssa. API:n avulla ohjelmat voivat pyytää ja vaihtaa tietoja tai käyttää toistensa toimintoja.

## b) Mikä on API URL?

API URL on osoite, joka ohjaa tiettyyn rajapintaan tai sen osaan. Tämä URL sisältää polun tiettyyn resurssiin tai toimintaan, jota kutsutaan rajapinnan kautta. Se on kuin osoite, jonka kautta lähetetään pyyntöjä ja vastaanotetaan tietoja.

## c) Mitä ovat parametrit?

Parametrit ovat muuttujia, joita lähetetään API-kutsun mukana, ja niiden avulla voidaan määrittää tai tarkentaa pyyntöä. Esimerkiksi hakuparametrit voivat määritellä, mitä tietoja haetaan ja millä kriteereillä.

## d) Mikä on päätepiste (endpoint)?

Endpoint eli päätepiste on API:n erityinen osoite, joka vastaa tiettyyn toimintoon tai resurssiin. Se on API URL osa, johon pyyntö lähetetään, ja se määrittelee, minkä tiedon tai toiminnon kanssa ollaan tekemisissä.

## e) Mikä on API avain/token?

API-avain tai -token on ainutlaatuinen tunnistus, jota käytetään API-kutsujen autentikointiin. Sen avulla API-palvelu tietää, kuka tekee pyynnön, ja voi tarkistaa, onko pyyntö oikeutettu. Se auttaa myös seuraamaan ja hallitsemaan käyttöoikeuksia.

## f) Mitä ovat ylätunnisteet?

Ylätunnisteet (headers) ovat lisätietoja, jotka lähetetään API-kutsun mukana ja jotka sisältävät esimerkiksi autentikointitietoja (kuten API-avain), tiedon pyyntömuodosta (esim. JSON tai XML) tai muita metatietoja, jotka auttavat palvelinta käsittelemään pyynnön oikein.

## g) Mikä on GET pyyntö?

GET-pyyntö on HTTP-metodi, jota käytetään API:n tietojen hakemiseen palvelimelta. GET-pyyntöissä lähetetään vain pyyntö, eikä siihen yleensä sisälly muuta tietoa kuin mahdolliset parametrit.

## h) Mikä on POST pyyntö?

POST-pyyntö on HTTP-metodi, jolla lähetetään tietoja palvelimelle API:n kautta. POST-pyyntöä käytetään yleensä silloin, kun halutaan luoda uusia tietoja tai lähettää dataa (esim. lomaketiedot tai joku tiedosto).

# JSON – JavaScript Object Notation

JSON (JavaScript Object Notation) on yksinkertainen ja kevyt tiedostomuoto, joka on helppolukuinen sekä ihmisille että koneille. Se perustuu JavaScriptin objektirakenteeseen, mutta on kieliriippumaton ja sitä käytetään laajasti eri ohjelmointikielissä tiedon siirtämiseen ja tallentamiseen. JSON on rakenteeltaan yksinkertainen, ja sen lukeminen ja kirjoittaminen on tehokasta.

JSON-tiedon perusrakenne muodostuu avain-arvo-pareista ja järjestetyistä arvojen listoista. JSON tukee muutamia yksinkertaisia tietotyyppejä:

- Merkkijono: Esitetään lainausmerkkien sisällä.
- Luku: Esitetään ilman lainausmerkkejä (kokonais- tai liukuluku).
- Boolean-arvo: tosi (true) tai epätosi (false).
- Taulukko: Järjestetty arvojen lista, joka esitetään hakasulkeiden [] sisällä.
- Objekti: Avain-arvo-parit, jotka esitetään aaltosulkeiden {} sisällä.
- Null: Arvo, joka osoittaa, että objekti ei sisällä mitään.

JSON:ia käytetään laajasti erilaisissa tietojenkäsittelysovelluksissa, erityisesti tiedon siirtämiseen ja säilyttämiseen. JSON on yleinen tiedonvaihtoformaatti web-sovelluksissa. Esimerkiksi selain ja palvelin voivat vaihtaa tietoa JSON-muodossa HTTP-pyyntöjen ja vastausten kautta. Eri ohjelmistot voivat myös vaihtaa tietoja JSON:in kautta riippumatta siitä, millä ohjelmointikielellä ne on kehitetty.

## C# ja JSON

C#-kielellä työskentelevät kehittäjät käyttävät usein JSON:ia tiedon serialisointiin ja deserialisointiin. Serialisointi tarkoittaa olion muuntamista JSON-muotoon, kun taas deserialisointi tarkoittaa JSON-datan muuntamista takaisin olioiksi. C# tarjoaa useita tapoja työskennellä JSON:in kanssa, ja suosituimpia ovat System.Text.Json ja Newtonsoft.Json (tunnetaan myös nimellä Json.NET).

System.Text.Json on Microsoftin virallinen JSON-kirjasto, joka on osa .NET Core -ekosysteemiä ja .NET 5.0 ja sitä uudempia versioita. Se on kevyt ja suorituskykyinen kirjasto JSON-datan käsittelyyn.

Newtonsoft.Json on yksi suosituimmista ja käytetyimmistä JSON-kirjastoista C#:ssa, erityisesti vanhemmissa .NET Framework -projekteissa. Se tarjoaa joustavimmat ominaisuudet ja enemmän lisäominaisuuksia kuin System.Text.Json, kuten tietojen validoinnin ja helpon muunnoksen komplekseihin olioihin.