

T1.1 Ykköstehtävä: Rajapinnat - yleistä

a) Mitä rajapinnat ovat? Miksi niitä tarvitaan? Mihin niitä käytetään? Mitä koet oppineesi tässä vaiheessa?

Rajapinnat (API, Application Programming Interface) ovat ohjelmointirajapintoja, jotka mahdollistavat eri ohjelmien ja järjestelmien kommunikoinnin keskenään. Rajapinta määrittelee säännöt ja sopimukset siitä, kuinka ohjelmat voivat pyytää toisiaan suorittamaan toimintoja tai vaihtamaan tietoja.

Miksi niitä tarvitaan:

- Rajapinnat mahdollistavat järjestelmien, sovellusten ja palveluiden integroimisen ja yhteistyön.
- Ne tarjoavat standardoidun tavan kommunikoida eri ohjelmien ja ohjelmistojen välillä ilman, että kaikki yksityiskohdat tarvitsee tietää.

Mihin niitä käytetään:

- **Web-palveluihin ja sovelluksiin:** Esimerkiksi sääpalveluiden, karttapalveluiden tai maksujärjestelmien rajapinnat tarjoavat tiedon tai toiminnallisuuksia, joita kolmannet osapuolet voivat käyttää.
- **Tietokantojen hallintaan:** Rajapinnat voivat tarjota ohjelmille pääsyn tietokantoihin, kuten API:t voivat mahdollistaa kyselyjen tekemisen tietokannasta.

Mitä koet oppineesi tässä vaiheessa:

- Rajapintojen merkitys ohjelmistokehityksessä on tullut selväksi, ja olen oppinut, kuinka rajapinnat voivat helpottaa tietojen ja toimintojen jakamista eri järjestelmien välillä.

b) Tutustu myös johonkin valitsemaasi rajapintaan syvemmin.

Kerro minkä rajapinnan valitsit? Miksi? Mihin sitä käytetään? Kuinka laaja rajapinta on, mitä kaikkea sillä voi tehdä? Pohdi voisiko tämä olla valitsemasi harjoitustyön aihe?

Valitsemani rajapinta: OpenWeatherMap API (säädata)

Miksi valitsin tämän rajapinnan?

- Valitsin OpenWeatherMap API:n, koska se tarjoaa helposti käytettävää sää tietoa, joka on yleisesti käytössä monissa sovelluksissa ja palveluissa.

Mihin sitä käytetään?

- OpenWeatherMap API tarjoaa sää tietoja, kuten lämpötilan, sademäärän, tuulen nopeuden ja muita sääolosuhteita eri paikkakunnilta. Tätä rajapintaa voidaan käyttää sovelluksissa, jotka tarjoavat sääennusteita.

Kuinka laaja rajapinta on?

- OpenWeatherMap API on laaja, sillä se tukee globaalisti säädataperusteista hakua ja tarjoaa useita eri päätepisteitä (endpoints) eri tarpeisiin, kuten historiatiedot, ennusteet ja ajankohtaiset tiedot.

Voisiko tämä olla valitsemasi harjoitustyön aihe?

- Tämä rajapinta voisi olla hyvä valinta harjoitustyöksi, koska se tarjoaa mielenkiintoista tietoa, jonka pohjalta voidaan rakentaa esimerkiksi sää tiedot näyttävä sovellus tai visuaalinen kartta, joka näyttää säilyneet tai ennustetut sääkohdat.

c) Ota selvää onko olemassa suomalaisia tietolähteitä, joihin on määritelty rajapinta? (Mahdollisia aiheita: Kartta/Aikataulut/Sää/Tilastot/...)

- Suomen Sää API (Meteorological Institute's) tarjoaa rajapinnan, josta voi saada sää tietoja eri paikkakunnilta Suomessa. Tämä rajapinta tarjoaa ajankohtaisia sääennusteita ja varoituksia.
- VR:n Aikataulut API tarjoaa tietoa junien aikatauluista ja reiteistä Suomessa. Tämä API olisi hyödyllinen sovelluksille, jotka tarjoavat reaaliaikaisia aikatauluja tai liikennetietoja.

T1.2 Käsitetehtävä (Määritelmiä/Apua/Ohjeita/Tietoa)

a) What is an API?

An **API (Application Programming Interface)** is a set of rules and protocols that allows one software application to interact with another. It defines the methods and data formats that applications use to request and exchange information. APIs are used to allow different

software systems to communicate with each other without exposing the internal workings of each system.

b) What is an API URL?

An API URL Path is an address that allows you to access an API and its various features. It is used to make requests to the server where the API resides. The URL typically includes the domain, the specific path to the endpoint, and sometimes query parameters for filtering or customizing the request.

c) What are parameters?

Parameters are values passed in the API request (usually in the URL or body) that define the specifics of the request, such as filtering data or setting limits. For example, when requesting weather data, a parameter might be the city name or country code to specify which location's weather to fetch.

d) What is an endpoint?

An endpoint is a specific URL path where an API can access resources or functionality. Each endpoint corresponds to a particular resource or service offered by the API. For example, the /weather endpoint might return weather data, while /forecast might return future weather predictions.

e) What is an API key/token?

An API key/token is a unique identifier used to authenticate a user or application to access an API. It ensures that the request is coming from a valid user and helps monitor API usage. API keys are commonly required to make requests to most public APIs.

f) What are headers?

API Headers are additional information sent with the HTTP request or response. They often include authentication information, content type, and other metadata like language preferences or cache settings. For example, a Content-Type header could indicate whether the request or response is in JSON or XML format.

g) What is a GET request?

A GET request is a type of HTTP request used to retrieve data from a server. It is typically used to fetch resources, such as a list of users or weather data, without modifying any server-side data.

h) What is a POST request?

A POST request is used to send data to a server to create or update resources. Unlike GET, which retrieves data, POST is used when you want to submit data, like creating a new user or submitting a form.

T1.3 JSON-tehtävä

Mikä on JSON?

JSON (JavaScript Object Notation) on kevyt tiedonsiirtomuoto, joka on helposti luettavissa ihmisille ja koneiden käsiteltävissä. JSON on tekstipohjainen ja sitä käytetään laajalti tietojen siirtämiseen verkkopalveluista ja sovelluksista. Se on suosittu erityisesti web-kehityksessä, ja se on virallinen internetin media tyyppi nimeltä application/json. JSON tiedoston päätteenä on .json.

Mihin sitä käytetään?

JSONia käytetään laajalti eri ohjelmointikielissä ja sovelluksissa, erityisesti verkkosovelluksissa ja API-rajapinnoissa, koska se on kevyempi ja helpompi käsitellä verrattuna muihin tiedonvaihtomuotoihin, kuten XML:ään. JSON mahdollistaa erilaisten tietojen siirron (esimerkiksi objekti- ja taulukkomuodossa) internetin yli palvelimien ja asiakasohjelmien välillä. Esimerkiksi verkkosivustot voivat käyttää JSON:ia saadakseen tietoja API-rajapinnoista, kuten uutisista, tai sosiaalisen median sisällöistä.

Miten C# toimii JSONin kanssa?

C# tukee JSON:ia suoraan System.Text.Json -kirjaston kautta, joka tarjoaa tehokkaita ja pieniresurssisia työkaluja JSON-datan käsittelyyn. Tämän kirjaston avulla voidaan muun muassa:

- Sarjallistaminen (serialization): Muuntaa C#-objekteja JSON-muotoon.
- Desarjallistaminen (deserialization): Muuntaa JSON-tekstiä takaisin C#-objekteiksi.

Esimerkki JSON:n käsittelystä C#:ssa:


```
0 references
static void Main(string[] args)
{
    // Example JSON string
    string jsonString = @"[{ "name": "Laiba Khan", "occupation": "Software Developer" },
    { "name": "Ali", "occupation": "driver" }]";

    // Deserialize JSON string into a dynamic object
    dynamic users = JsonConvert.DeserializeObject(jsonString);
    Console.WriteLine("Deserialized JSON:");
    foreach (var user in users)
    {
        Console.WriteLine($"Name: {user.name}, Occupation: {user.occupation}");
    }

    // now creating new object
    var newUser = new User
    {
        Name = "Komal Khan",
        Occupation = "Parlour",
        DateOfBirth = new MyDate { Year = 1995, Month = 4, Day = 13 }
    };

    // Serialize the object back to JSON
    string newJsonString = JsonConvert.SerializeObject(newUser, Formatting.Indented);

    Console.WriteLine("\nSerialized JSON:");
    Console.WriteLine(newJsonString);
}
```

 Microsoft Visual Studio Debug Console

```
Deserialized JSON:
Name: Laiba Khan, Occupation: Software Developer
Name: Ali, Occupation: driver

Serialized JSON:
{
  "Name": "Komal Khan",
  "Occupation": "Parlour",
  "DateOfBirth": {
    "Year": 1995,
    "Month": 4,
    "Day": 13
  }
}
```