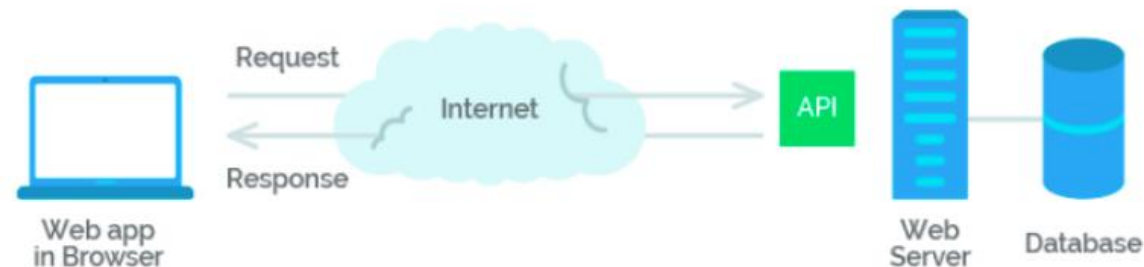


DE24- Programming 40 Yhp

API

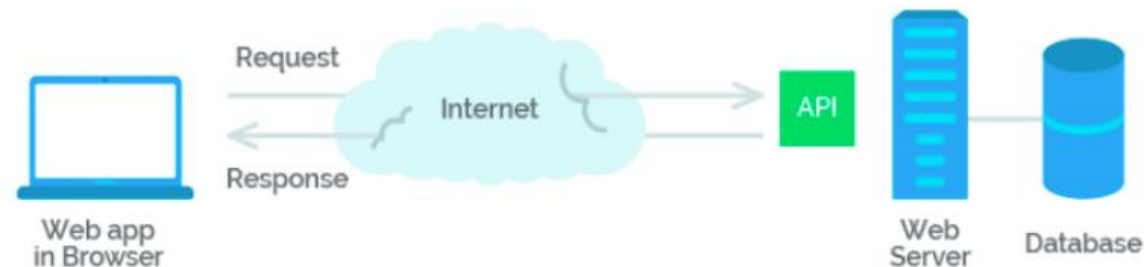
API

- Ett API, eller Application Programming Interface, är en uppsättning regler som låter ett mjukvaruprogram eller system interagera med ett annat.
- Det definierar metoder och datastrukturer som utvecklare kan använda för att interagera med mjukvarukomponenten, vare sig det är operativsystem, bibliotek eller olika tjänster. API:er används för att möjliggöra integration mellan olika mjukvarusystem.



Rest API

- REST bestämmer hur API:t ser ut. Det står för "Representational State Transfer". Det är en uppsättning regler som utvecklare följer när de skapar sitt API.
- Att anropa en URL är en förfrågan medan datan som skickas tillbaka till dig kallas ett response.



Anatomin av en förfrågan

- The endpoint
- The method
- The headers
- The data (or body)

The Endpoint

- Endpointen (eller vägen) är URL:en som du begär.

Den följer denna struktur:

```
root-endpoint/?
```

- Root-endpoint är startpunkten för det API som du begär data från.

Root-endpoint för GitHubs API är:

```
https://api.github.com
```

The Endpoint/The Path

- För att förstå vilka vägar som är tillgängliga i API:t behöver du gå igenom API-dokumentationen. Till exempel, om vi vill ha en lista över repositories från en viss användare via GitHubs API, visar dokumentationen att du ska använda följande väg:

```
https://api.github.com/users/:username/repos
```

- Kolon (:) i en väg markerar en variabel. Ersätt dessa värden med faktiska värden när du skickar din förfrågan. I detta fall med det faktiska användarnamnet (LexiconNE20). Det kan också vara {username} eller [username].

```
https://api.github.com/users/LexiconNDE20-2/repos
```

```
https://api.github.com/users/LexiconNDE20-2/repos?sort=pushed
```

The Endpoint/The Path

URL-Parametrar (Query Parameters)

- Det är vanligt att inkludera olika parametrar i förfrågan, till exempel för att begära ett specifikt format eller en begränsning av data i svaret.
- Strukturen bestäms av utvecklaren, men det är ofta ett "?" före den första parametern och "&" för följande parametrar i URL:en.

```
https://api.github.com/users/LexiconNDE20-2/repos
```

```
https://api.github.com/users/LexiconNDE20-2/repos?sort=pushed
```

```
https://api.github.com/users/LexiconNDE20-2/repos?sort=pushed&per_page=5
```

The Method

- **GET**
- **POST**
- **PUT**
- **PATCH**
- **DELETE**

Dessa används för att utföra fyra möjliga åtgärder:
C.R.U.D.

Kom ihåg att hur metoderna används faktiskt är upp till utvecklaren.

GET

- GET används för att hämta en resurs från en server. Om du gör en `GET`-förfrågan, letar servern efter den data du begärde och skickar tillbaka den till dig. Med andra ord, en `GET`-förfrågan utför en `READ`-operation.
- Det här är standardmetoden för förfrågningar och när du surfar på webbsidor används faktiskt denna metod.

POST

- Denna förfrågan används för att skapa en ny resurs på en server. Om du gör en `POST`-förfrågan, skapar servern en ny post i databasen och meddelar om skapandet var framgångsrikt.
- Med andra ord, en `POST`-förfrågan utför en `CREATE`-operation.

PUT & PATCH

- Dessa två förfrågningar används för att uppdatera en resurs på en server. Om du gör en `PUT`- eller `PATCH`-förfrågan, uppdaterar servern en post i databasen och meddelar om uppdateringen lyckades.
- Med andra ord, en `PUT`- eller `PATCH`-förfrågan utför en `UPDATE`-operation.

DELETE

- Den här förfrågan används för att ta bort en resurs från en server. Om du gör en `DELETE`-förfrågan, tar servern bort en post i databasen och meddelar om borttagningen lyckades.
- Med andra ord, en `DELETE`-förfrågan utför en `DELETE`-operation.

The Headers

- Headers används för att ge information till både klienten och servern.
- De kan användas för många ändamål, såsom autentisering och att ge information om body-innehållet.

HTTP Headers Reference

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>

The Headers

- Det här exemplet säger till servern att jag kommer att skicka JSON-innehåll.

`"Content-Type: application/json" https://api.github.com`

The Body (The Data) JSON

```
{
  "id": 284640588,
  "node_id": "MDEwOlJlcG9zaXRvcnkyODQ2NDA1ODg=",
  "name": "MyShop",
  "full_name": "LexiconNE20/MyShop",
  "private": false,
  "owner": {
    "login": "LexiconNE20",
    "id": 65213711,
    "node_id": "MDQ6VXNlcjY1MjEzNzEx",
    "avatar_url": "https://avatars0.githubusercontent.com/u/65213711?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/LexiconNE20",
    "html_url": "https://github.com/LexiconNE20",
    "followers_url": "https://api.github.com/users/LexiconNE20/followers",
    "following_url": "https://api.github.com/users/LexiconNE20/following{/other_user}",
    "gists_url": "https://api.github.com/users/LexiconNE20/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/LexiconNE20/starred{/owner}/{/repo}",
    "subscriptions_url": "https://api.github.com/users/LexiconNE20/subscriptions",
    "organizations_url": "https://api.github.com/users/LexiconNE20/orgs",
    "repos_url": "https://api.github.com/users/LexiconNE20/repos",
    "events_url": "https://api.github.com/users/LexiconNE20/events{/privacy}",
    "received_events_url": "https://api.github.com/users/LexiconNE20/received_events",
    "type": "User",
    "site_admin": false
  },
  ...
}
```

HTTP status Codes /Error messages

- **100+** står för någon form av information.
- **200+** betyder att förfrågan har lyckats.
- **300+** betyder att förfrågan omdirigeras till en annan URL.
- **400+** betyder ett fel som härstammar från klienten.
- **500+** betyder ett fel som härstammar från servern.