

An aerial photograph showing a dark asphalt road winding through a dense forest. The forest consists of various trees, including conifers and deciduous ones, with some showing autumnal colors. The road has white dashed lines and is surrounded by green and brown vegetation.

API

Hva er det?

An aerial photograph showing a dark asphalt road winding through a dense forest. The forest consists of various trees, including conifers and deciduous ones, with some showing autumnal colors. The road has white dashed lines and is surrounded by green and brown vegetation.

API

Hva er det?

Frontend

API

Frontend

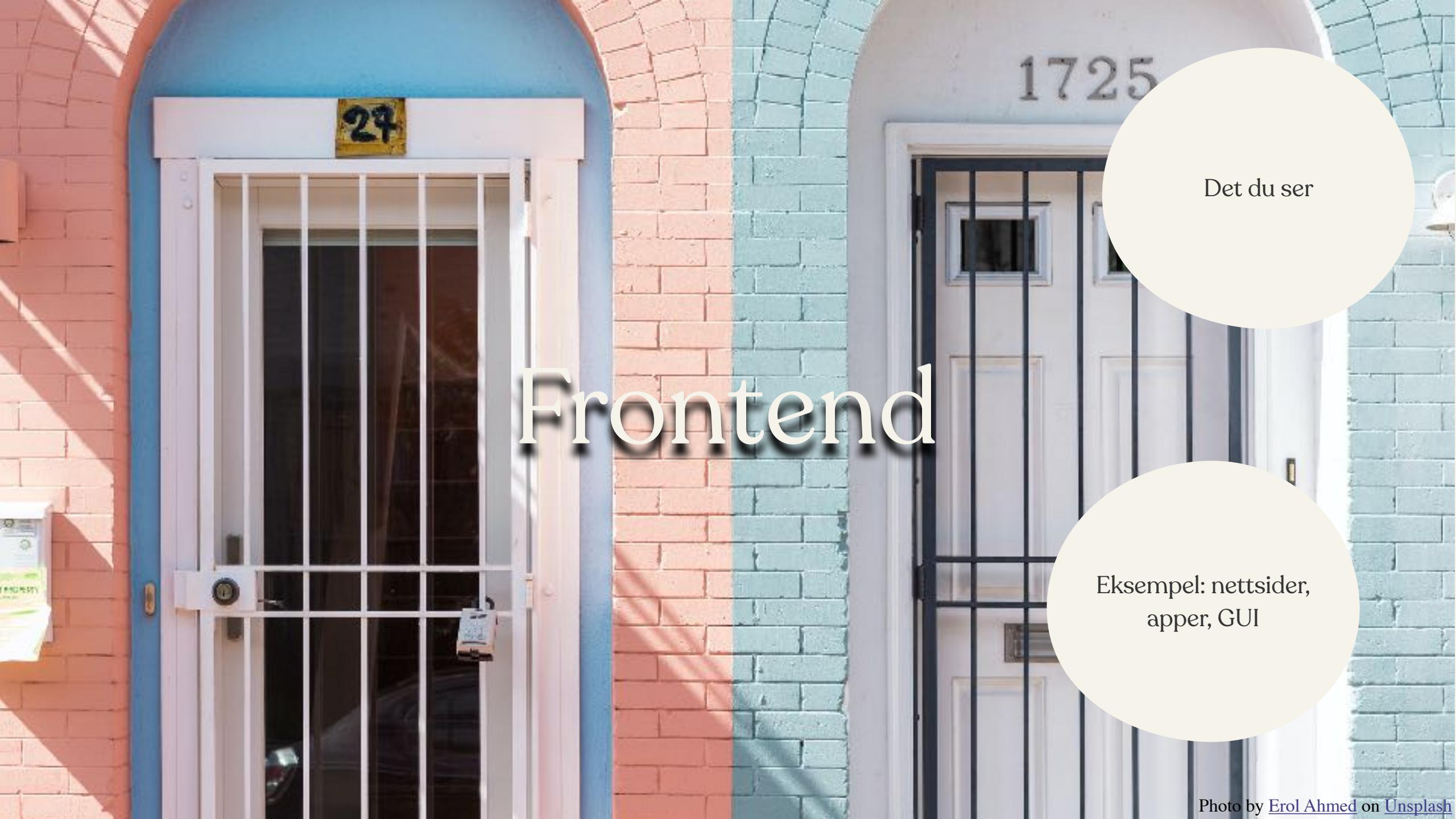
API

API

API

API

Backend



Frontend

Det du ser

Eksempel: nettsider,
apper, GUI

Backend

Det du ikke ser

Eksempel: server,
databaser, program



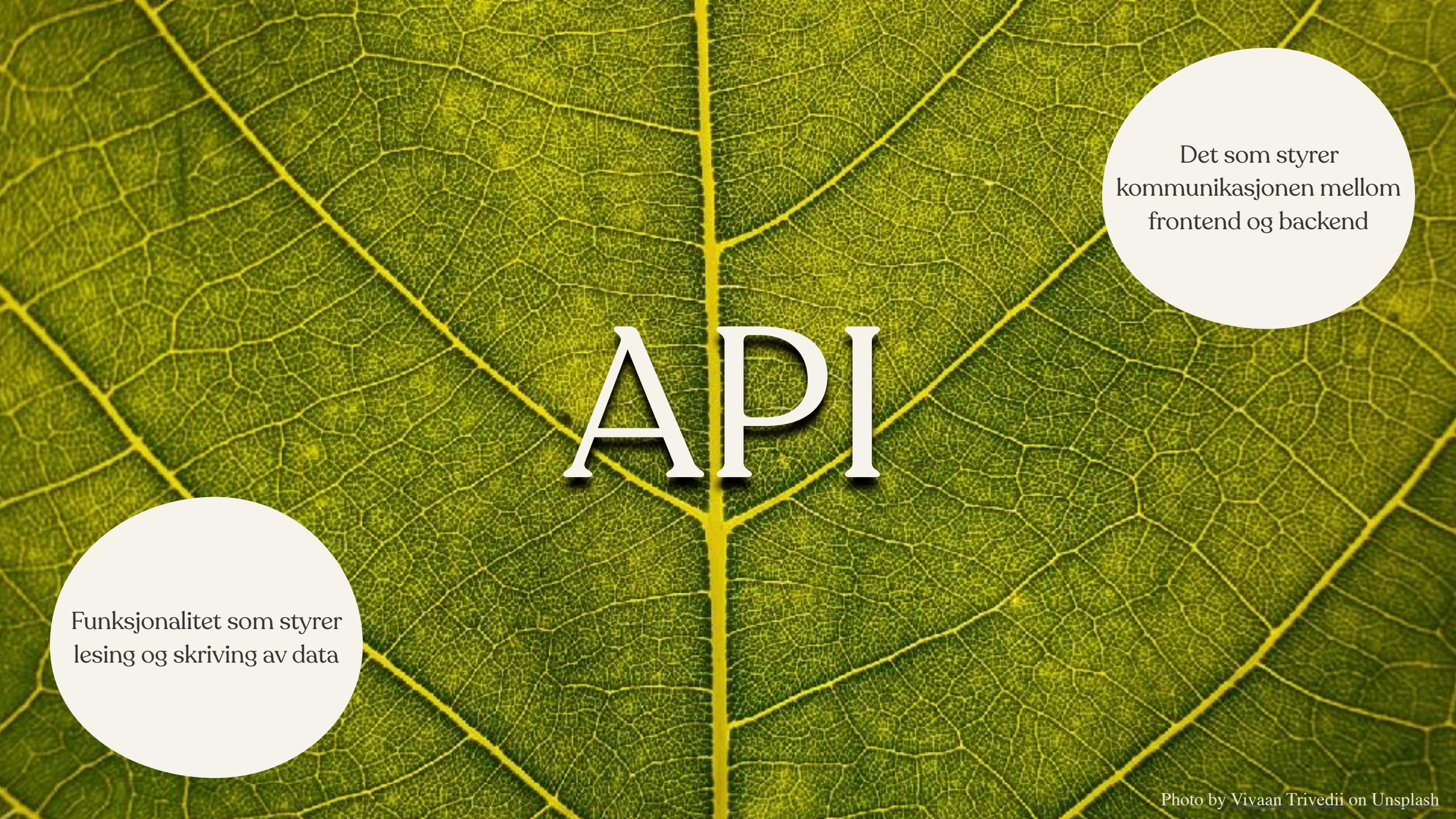
API

Application Programming Interface

Photo by [Erol Ahmed](#) on [Unsplash](#)



Photo by [William Bayreuther](#) on [Unsplash](#)



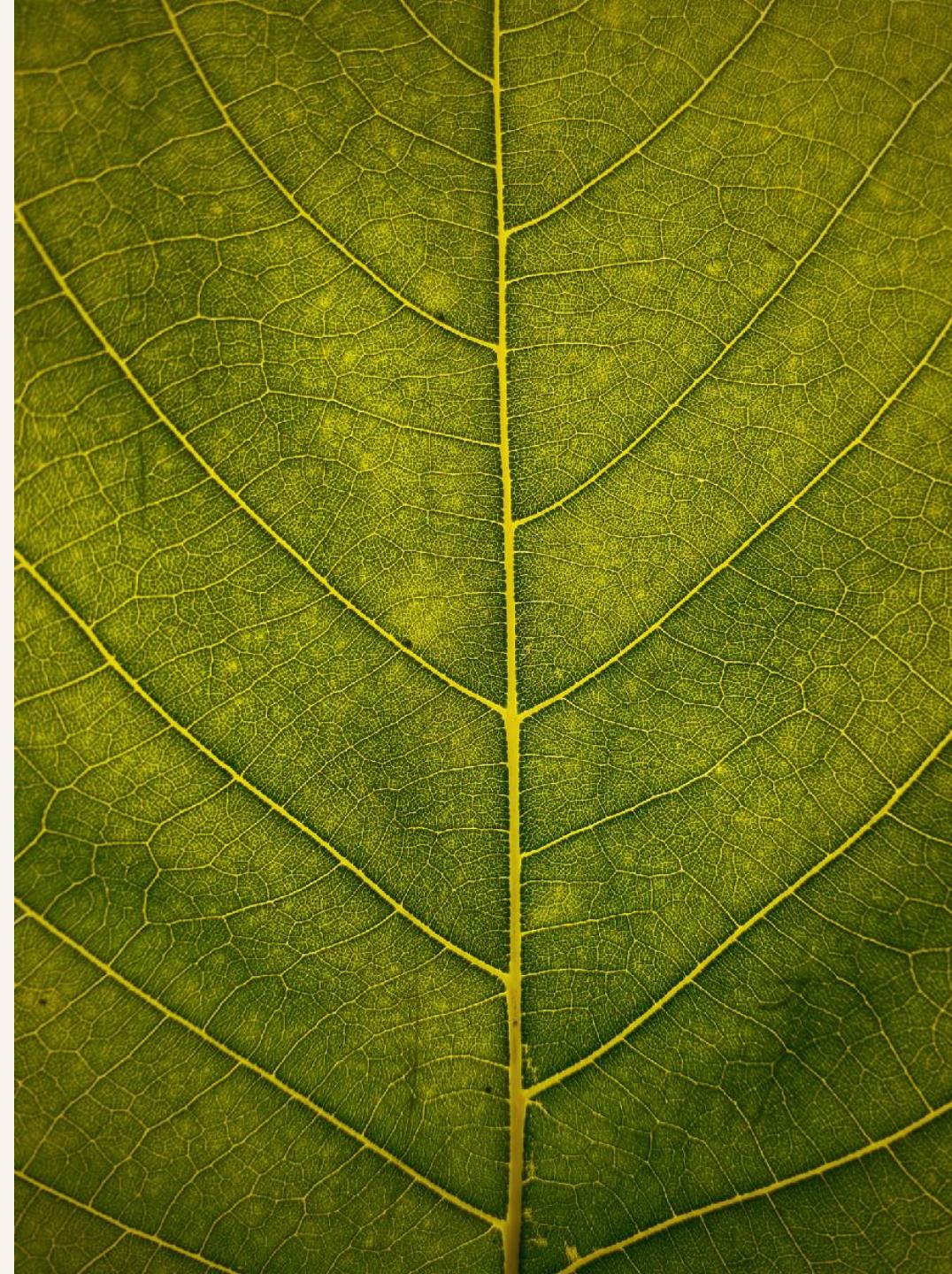
API

Funksjonalitet som styrer
lesing og skriving av data

Det som styrer
kommunikasjonen mellom
frontend og backend

API

- Et mellomlag mellom frontend-backend eller backend-backend
- Vanlig å ha et API per «tema»
F.eks.: Spotify har API for brukere, artister, låter osv
- Et API kan ha mange endepunkt

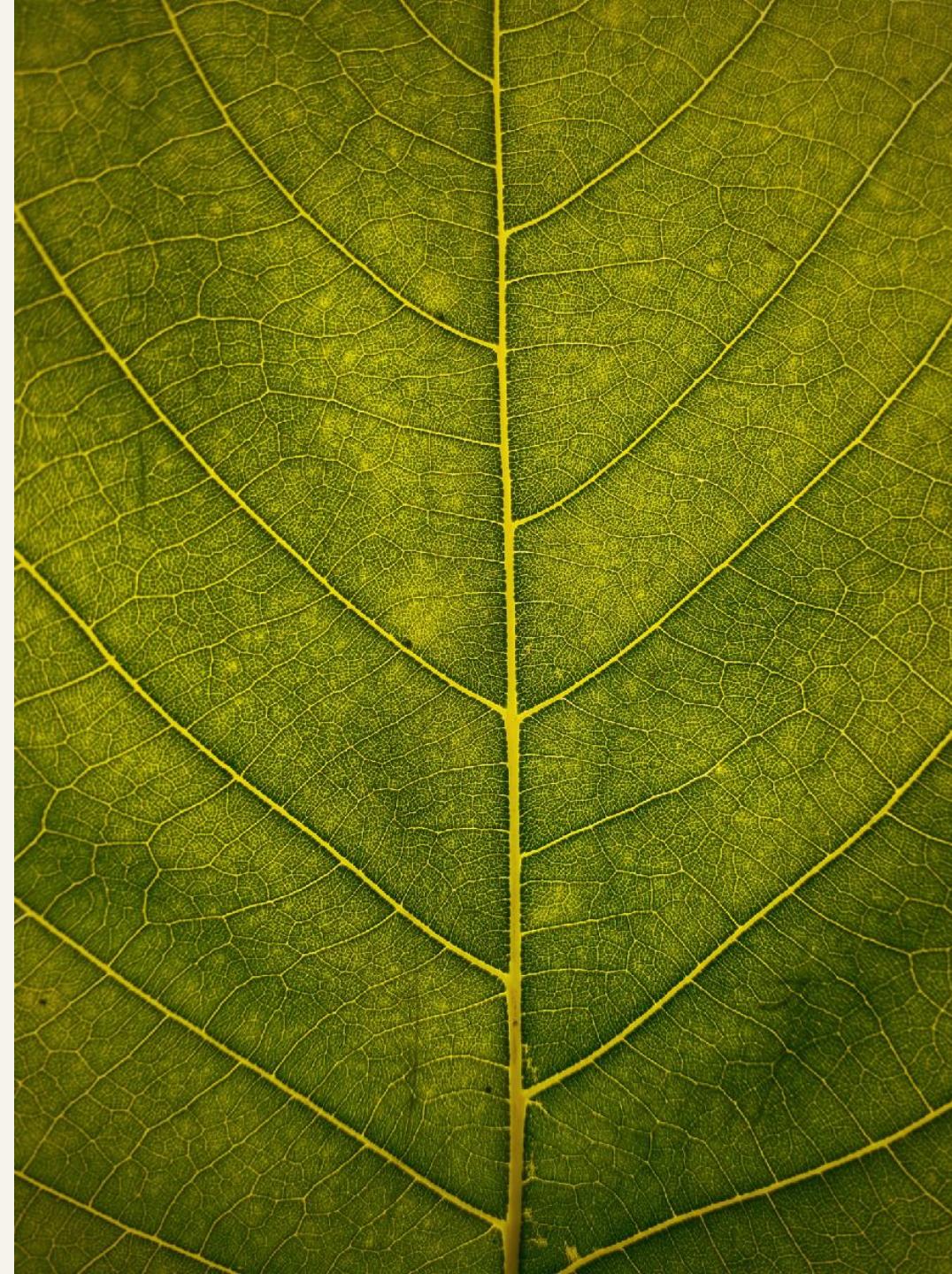


API

- Et API har flere metoder for å lese/skrive data til/fra server: GET, POST, PUT, DELETE

- Hver metode har et request-response-par.

Du sender en request også får du en response tilbake.



Eksempler

Ved bruk av Spotify API

Endpunkt

REFERENCE

[Reference Index](#)

[Search API](#)

[Browse API](#)

[Follow API](#)

[Playlists API](#)

[Library API](#)

[Artists API](#)

[Player API](#)

[Markets API](#)

[Personalization API](#)

[User Profile API](#)

[Albums API](#)

[Tracks API](#)

[Episodes API](#)

[Shows API](#)

[Objects Index](#)

Albums API

- Get Multiple Albums
- Get an Album
- Get an Album's Tracks

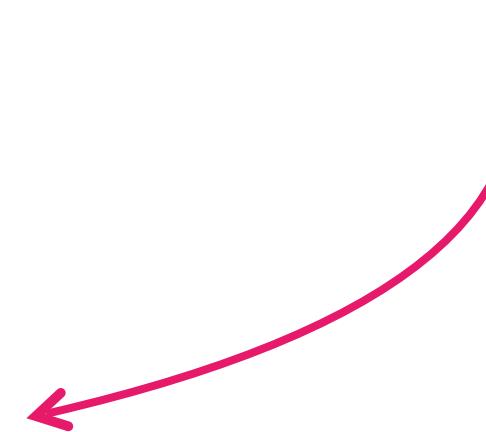
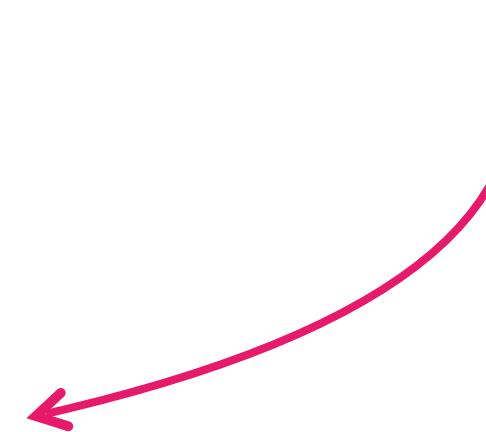
Artists API

- Get Multiple Artists
- Get an Artist
- Get an Artist's Top Tracks
- Get an Artist's Related Artists
- Get an Artist's Albums

Browse API

- Get All New Releases
- Get All Featured Playlists
- Get All Categories
- Get a Category
- Get a Category's Playlists
- Get Recommendations
- Get Recommendation Genres

APIer



Request-response

Eksempel på hvordan en request og en response kan se ut

Request

A screenshot of a web browser showing the Spotify Developer API console. The URL in the address bar is `developer.spotify.com/console/get-track/?id=3n3Ppam7vgaVaiiaRUc9Lp`. The page title is "Get a Track".

The left sidebar lists various API endpoints: CONSOLE, Albums, Artists, Browse, Episodes, Follow, Library, Markets, Personalization, Player, Playlists, Search, Tracks, Shows, and Users Profile.

The main content area shows the "Get a Track" endpoint details:

- API Reference: [Get a Track](#)
- Endpoint: `https://api.spotify.com/v1/tracks/{id}`
- HTTP Method: GET
- OAuth: Required

Form fields for the request parameters:

- id*: `3n3Ppam7vgaVaiiaRUc9Lp`
- market: `ES`

An OAuth Token field contains the value: `BQD3PPUfayJyN+KMG_Fl0a4ja8rm7OnZeJu...`. A green button labeled "GET TOKEN" is visible next to the token input field.

The right side of the screen displays the generated curl command:

```
curl -X "GET" "https://api.spotify.com/v1/tracks/3n3Ppam7vgaVaiiaRUc9Lp" -H "Accept: application/json" -H "Content-Type: application/json" -H "Authorization: Bearer BQD3PPUfayJyN+KMG_Fl0a4ja8rm7OnZeJuCGr6k_hhDPuNBz7NYtzY9eo7RtbNnPg0njCCry0cTZG0lHM1vJC6Zm...yZgNUPz3rrTN050kdIbi60sPvLB-Y5GQC6dJnxHnIWE7iXDz_I"
```

Headers

Preview

Response

Initiator

Timing

Request

General

Request URL: <https://api.spotify.com/v1/tracks/3n3Ppam7vgaVa1iaRUC9Lp>

Request Method: GET

Status Code: ● 200

Response

x Headers Preview Response Initiator Timing

▼ {album: {album_type: "album", ...}, ...}

▶ album: {album_type: "album", ...}

▶ artists: [{external_urls: {spotify: "https://open.spotify.com/track/3n3Ppam7vgaVa1iaRUC9Lp"}, name: "The Killers"}, {external_urls: {spotify: "https://open.spotify.com/artist/3n3Ppam7vgaVa1iaRUC9Lp"}, name: "The Killers"}]

▶ available_markets: ["AD", "AE", "AG", "AL", "AM", "AO", "AR", "AT", "AU", "BE", "BG", "CL", "CY", "DE", "DK", "DO", "ES", "FI", "FR", "GB", "GR", "HK", "HU", "ID", "IL", "IN", "IS", "IT", "MX", "NL", "NO", "NZ", "PE", "PT", "RO", "SE", "SI", "TR", "US", "VE"]

disc_number: 1

duration_ms: 222200

explicit: false

▶ external_ids: {isrc: "GBFFP0300052"}

▶ external_urls: {spotify: "https://open.spotify.com/track/3n3Ppam7vgaVa1iaRUC9Lp", href: "https://api.spotify.com/v1/tracks/3n3Ppam7vgaVa1iaRUC9Lp", id: "3n3Ppam7vgaVa1iaRUC9Lp"}

is_local: false

name: "Mr. Brightside"



GET, POST, PUT, DELETE

Eksempel på hvordan disse metodene brukes

GET Hente data

The image shows a Spotify album page for 'Hot Fuss' by The Killers. The album cover features a city skyline at night with the text 'KILLERS HOT FUSS'. The title 'Hot Fuss' is prominently displayed in large white letters. Below the title, it says 'The Killers • 2004 • 12 låter, 49 min 52 sek'. On the left, there's a play button, a heart icon, and a more options icon. The track list table has columns for number (#), title (TITTEL), and duration (⌚). The tracks listed are:

#	TITTEL	⌚
1	Jenny Was A Friend Of Mine The Killers	4:03
2	Mr. Brightside The Killers	3:42
3	Smile Like You Mean It The Killers	3:54
4	Somebody Told Me	3:17

On the right, a browser developer tools Network tab is open, showing a request for a track's data. The request URL is `https://api.spotify.com/v1/tracks/003vvx7Niy0yvhvHt4a68B?market=from_token`. The request method is GET, and the status code is 200 (from service worker). The referrer policy is strict-origin-when-cross-origin.

GET

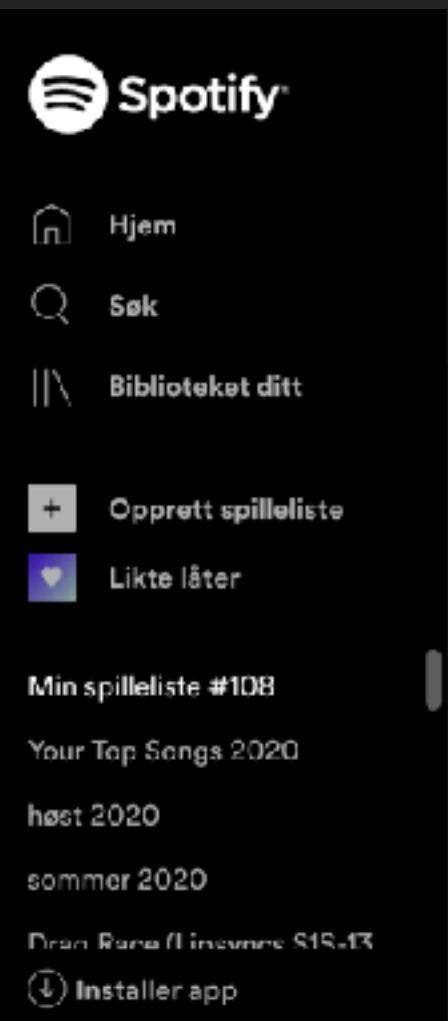


```
const fetchTrack = async (trackId: string) => {
  const response = await fetch(`https://api.spotify.com/v1/tracks/${trackId}`);
  if (response.status === 200) {
    return await response.json();
  }
  return "Could not find track";
}
```

▼ Request Payload

[view source](#)

```
▼ ops: [{kind: 6, updateListAttributes: {newAttributes: {values: {name: "Min spilleliste #108"}}}}]  
► ops: [{kind: 6, updateListAttributes: {newAttributes: {values: {name: "Min spilleliste #108"}}}}]
```



Request URL: <https://spclient.wg.spotify.com/playlist/v2/playlist>

Request Method: POST

Status Code: 200

Remote Address: 35.186.224.25:443

Referrer Policy: strict-origin-when-cross-origin

POST
Skrive data

POST



```
const addNewPlaylist = async (playlistData: Playlist) => {
  const response = await fetch("https://api.spotify.com/v1/playlist/", {
    method: "POST",
    body: JSON.stringify(playlistData)
  });
  if (response.status === 200) {
    return await response.json();
  }
  return "Could not create new playlist";
}
```

▼ Request Payload

[view source](#)

```
▼ {uris: ["spotify:track:2c3sKqWYVJ0NsE3elgRhKH"], position: null}  
  position: null  
  ▶ uris: ["spotify:track:2c3sKqWYVJ0NsE3elgRhKH"]
```



Screenshot of the Spotify web player interface showing a playlist with tracks by O-Zone and Jay Rock. The developer tools Network tab is open, showing a POST request to https://api.spotify.com/v1/playlists/73bxU2kSj7fjz4KzX70pmc/tracks. The request payload is {uris: ["spotify:track:2c3sKqWYVJ0NsE3elgRhKH"], position: null}, with the position field explicitly set to null.

Name	Value
Request URL	https://api.spotify.com/v1/playlists/73bxU2kSj7fjz4KzX70pmc/tracks
Request Method	POST
Status Code	201
Remote Address	35.186.224.25:443
Referrer Policy	strict-origin-when-cross-origin

POST
Skrive data

▼ Request Payload [view source](#)

```
▼ {range_start: 1, range_length: 1, insert_before: 0}  
  insert_before: 0  
  range_length: 1  
  range_start: 1
```

The screenshot shows a Spotify web player interface on the left and the browser's developer tools on the right. The developer tools are open to the Network tab, where a recent request is selected.

Request URL: `https://api.spotify.com/v1/playlists/73bxU2kSj7fjz4KzX70pmc/tracks`

Request Method: PUT

Status Code: 200

Remote Address: 35.186.224.25:443

Referrer Policy: strict-origin-when-cross-origin

The Spotify player shows a playlist with two tracks: "King's Dead (will.i.am vs Jay Rock, Ken...)" and "Dragostea Din ... O-Zone".

PUT

Oppdatere data

▼ Request Payload

[view source](#)

- ▼ {tracks: [{uri: "spotify:track:2c3sKqWYVJ0NsE3elgRhKH", positions: [1]}]}
- tracks: [{uri: "spotify:track:2c3sKqWYVJ0NsE3elgRhKH", positions: [1]}]

The screenshot shows the Spotify web player interface on the left and the Chrome DevTools Network tab on the right. The Spotify interface displays a play button and a list of songs in a playlist titled 'Anbefalt'. The DevTools Network tab lists several network requests, with one specific DELETE request highlighted. The request details are as follows:

- Name:** tr...
- Request URL:** https://api.spotify.com/v1/playlists/73bxU2kSj7fjz4KzX70pmc/tracks
- Request Method:** DELETE
- Status Code:** 200
- Remote Address:** 35.186.224.25:443
- Referrer Policy:** strict-origin-when-cross-origin

DELETE
Slette data

Frontend

API

Frontend

API

API

API

API

Backend