



Application Programming Interface

David Schol / 2022

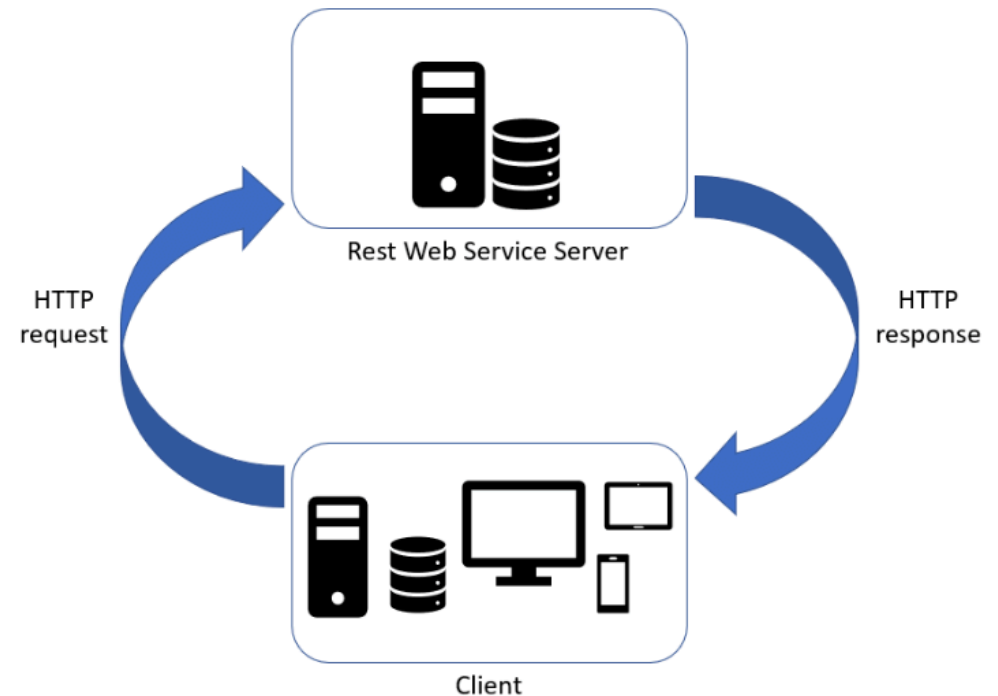
Wat is een API?

{ api }

API is het acroniem voor Application Programming Interface, een (software) intermediair waarmee twee toepassingen met elkaar kunnen praten. Telkens wanneer je een app als TikTok gebruikt, een Whatsapp bericht verstuurt of het weer op je telefoon bekijkt, gebruik je een API.

Wanneer je een app op je mobiele telefoon gebruikt, maakt die app verbinding met het internet en stuurt gegevens naar een server. De server haalt die gegevens vervolgens op, interpreteert ze, voert de nodige acties uit en stuurt ze terug naar jouw telefoon. De applicatie interpreteert die gegevens vervolgens en presenteert je de gewenste informatie op een leesbare manier.

Voorbeeld: Stel je voor dat je in een restaurant aan een tafel zit met een menu waaruit je kunt kiezen. De keuken is het deel van het "systeem" dat jouw bestelling klaarmaakt. Wat ontbreekt is de link om je bestelling door te geven aan de keuken en je eten terug naar je tafel te brengen. Daar komt de ober om de hoek kijken. De ober is de boodschapper - of API - die jouw verzoek of bestelling aanneemt en de keuken - het systeem - vertelt wat te doen. Vervolgens levert de ober je het antwoord terug; in dit geval is dat het eten.



Moderne API

{ api }

Een API biedt ook een beveiligingslaag

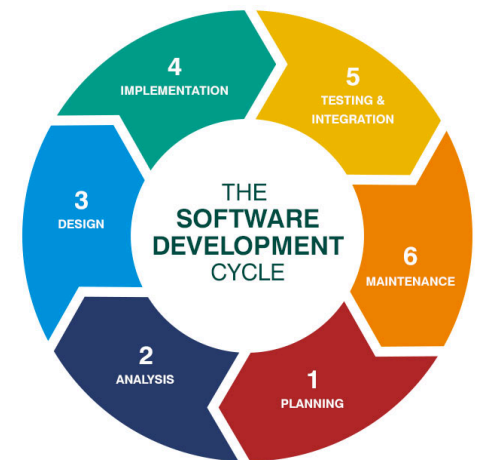
De gegevens van je telefoon zijn nooit volledig zichtbaar voor de server, en evenzo is de server nooit volledig zichtbaar voor je telefoon. In plaats daarvan communiceert ieder met kleine pakketjes gegevens, waarbij alleen de noodzakelijke gegevens worden gedeeld, zoals bij het bestellen van een afhaalmaaltijd. Je vertelt het restaurant wat je wilt eten, zij vertellen je wat ze daarvoor nodig hebben en uiteindelijk krijg je je maaltijd.

API's zijn zo waardevol geworden dat ze een groot deel van de inkomsten van veel bedrijven uitmaken. Grote bedrijven als Google, eBay en Amazon zijn slechts enkele van de bedrijven die geld verdienen aan hun API's.

De moderne API

In de loop der jaren heeft een "API" vaak een soort generieke interface voor een toepassing beschreven. Meer recentelijk echter heeft de moderne API een aantal kenmerken gekregen die hem buitengewoon waardevol en nuttig maken:

- Moderne API's houden zich aan normen (meestal HTTP en REST), die ontwikkelaar vriendelijk, gemakkelijk toegankelijk en begrijpelijk zijn
- Ze worden meer behandeld als producten dan als code. Ze zijn ontworpen voor gebruik door specifieke doelgroepen (bv. mobiele ontwikkelaars)
- De moderne API heeft zijn eigen levenscyclus voor softwareontwikkeling (SDLC) van ontwerpen, testen, bouwen, beheren en versiebeheer
- Ook zijn moderne API's goed gedocumenteerd voor gebruik en versiebeheer

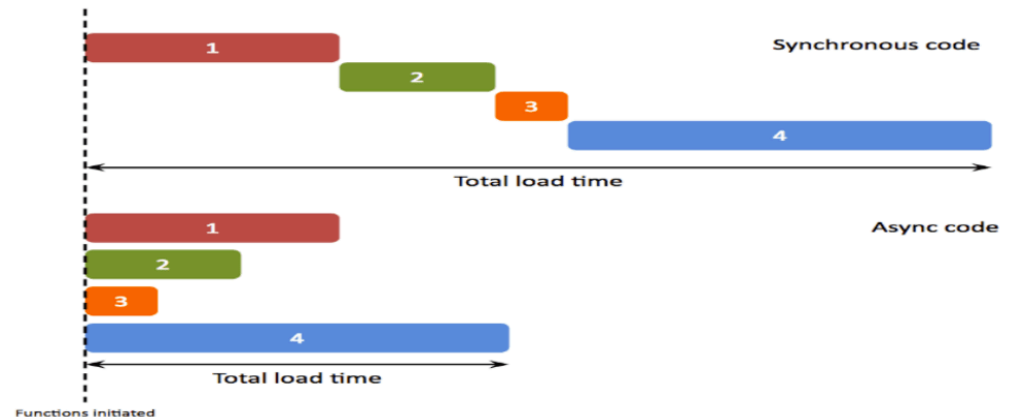


JS is een single threaded taal

{ api }

Aangezien JavaScript een single-threaded programmeertaal is, kan het slechts één statement tegelijk verwerken. Een actie als het opvragen van gegevens bij een API kan echter onbepaalde tijd in beslag nemen, afhankelijk van de omvang van de opgevraagde gegevens, de snelheid van de netwerkverbinding en andere factoren. Als API-aanroepen synchroon zouden worden uitgevoerd, zou de browser geen gebruikersinvoer kunnen verwerken, zoals scrollen of klikken op een knop, totdat die handeling is voltooid. Dit wordt blokkeren genoemd.

Om blokkerend gedrag te voorkomen heeft de browseromgeving veel web-API's waartoe JavaScript toegang heeft die asynchroon zijn, wat betekent dat ze parallel met andere bewerkingen kunnen worden uitgevoerd in plaats van achter elkaar. Dit is nuttig omdat de gebruiker dan de browser normaal kan blijven gebruiken terwijl de asynchrone bewerkingen worden verwerkt.

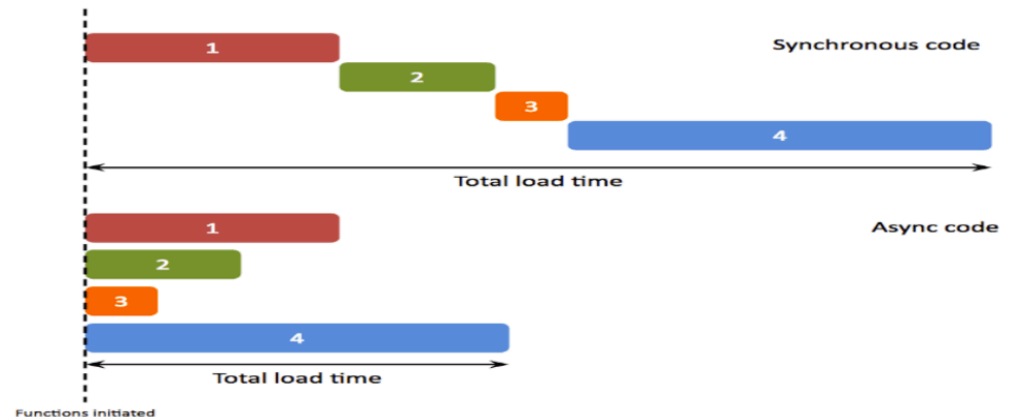


JS is een single threaded taal

{ api }

Aangezien JavaScript een single-threaded programmeertaal is, kan het slechts één statement tegelijk verwerken. Een actie als het opvragen van gegevens bij een API kan echter onbepaalde tijd in beslag nemen, afhankelijk van de omvang van de opgevraagde gegevens, de snelheid van de netwerkverbinding en andere factoren. Als API-aanroepen synchroon zouden worden uitgevoerd, zou de browser geen gebruikersinvoer kunnen verwerken, zoals scrollen of klikken op een knop, totdat die handeling is voltooid. Dit wordt blokkeren genoemd.

Om blokkerend gedrag te voorkomen heeft de browseromgeving veel web-API's waartoe JavaScript toegang heeft die asynchroon zijn, wat betekent dat ze parallel met andere bewerkingen kunnen worden uitgevoerd in plaats van achter elkaar. Dit is nuttig omdat de gebruiker dan de browser normaal kan blijven gebruiken terwijl de asynchrone bewerkingen worden verwerkt.

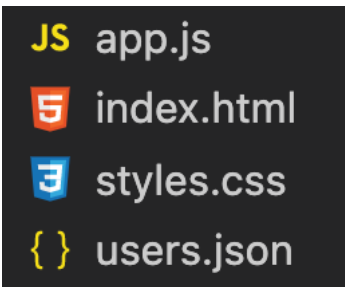


Basis API call

We maken in VSCode een nieuw project (naam: API call) aan, met de bestanden zoals ze in het onderstaande schermabeelding staan. We gaan nu één voor één de bestanden voorzien van de inhoud om de data uit het JSON bestand te halen en deze te tonen in de html pagina. Het css bestand is puur decoratief en is dus optioneel.

Zorg dat je de extensie 'live server' hebt geïnstalleerd in VSCode, om steeds te werking van de code te bekijken.

De uitvoer die we via een console.log() versturen, zie je terug in de JavaScript console. (View -> Developer -> JavaScript console of met de toetscombinatie Command+Option+J (Mac) or Control+Shift+J (Windows)).



```
[
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "city": "Gwenborough",
      "zipcode": "92998-3874"
    }
  },
  {
    "id": 2,
    "name": "Ervin Howell",
    "username": "Antonette",
    "email": "Shanna@melissa.tv",
    "address": {
      "street": "Victor Plains",
      "city": "Wisokyburgh",
      "zipcode": "90566-7771"
    }
  },
  {
    "id": 3,
    "name": "Clementine Bauch",
    "username": "Samantha",
    "email": "Nathan@yesenia.net",
    "address": {
      "street": "Douglas Extension",
      "city": "McKenziehaven",
      "zipcode": "59590-4157"
    }
  }
]
```



users.json

Basis API call

{ api }

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>API</title>
</head>

<body>
  <h1>Persons</h1>
  <ul id="persons-list"></ul>

  <script src="app.js"></script>
</body>
</html>
```

index.html

```
ul {
  margin: 0;
  padding: 0;
}

li {
  list-style: none;
  border: 1px solid #333;
  border-radius: 10px;
  padding: 5px 15px 10px 15px;
  margin-bottom: 10px;
  max-width: 260px;
}

li h2 {
  margin: 2px 0 5px 0;
}
```

styles.css

Basis API call

{ api }

```
const url = 'users.json';

const getUsers = async () => {

  const response = await fetch(url);
  if (!response.ok) {
    const message = `An error has occurred. Status: ${response.status}`;
    throw new Error(message);
  }

  const data = await response.json();
  return data;
};
```

```
getUsers()
  .then(data => console.log('Resolved: ', data))
  .catch(err => console.log('Rejected: ', err));
```

app.js

Het JavaScript bestand bestaat uit een functie en de aanroep van de functie.

We leggen met deze code de basis voor de API call. We halen de data op, maar tonen deze nog niet op de html pagina.

async /await zorgen ervoor dat de data binnen wordt, andere processen kunnen nu wel doorlopen.

Probeer het volgende uit:

1. Bekijk de uitvoer in de console
2. Pas de url aan (die is nu dus fout)
Bekijk de uitvoer in de console
3. Pas de json aan (die is nu dus fout)
Bekijk de uitvoer in de console

Basis API call

{ api }

```
const ul = document.querySelector('#persons-list');
```

```
getUsers()  
  .then(data => {  
    console.log('Resolved: ', data);  
  
    let users = data;  
  
    users.map(user => {  
      let li = document.createElement('li');  
      let name = document.createElement('h2');  
      let email = document.createElement('span');  
  
      name.textContent = `${user.name}`;  
      email.textContent = `${user.email}`;  
  
      li.appendChild(name);  
      li.appendChild(email);  
      ul.appendChild(li);  
    });  
  })  
  .catch(err => console.log('Rejected: ', err));
```

app.js

We gaan nu het element opzoeken in het html bestand waar we de data willen plaatsen. Voeg de regel toe, die het id met 'persons-list' voor ons opzoekt. In dit element gaan we onze data zetten. (Met eventuele opmaak).

Pas het .then element van de aanroep van de functie aan, zodat alle data opgehaald wordt en vormgegeven wordt door middel van html-tags.

1. Pas de url aan, zodat we nu geen lokale data ophalen, maar van een API end-point:

```
const url = 'https://  
jsonplaceholder.typicode.com/users';
```

2. Toon nu ook de adresgegevens en maak dit mooi op. (Gebruikt de console, om te zien of je het juiste ophaalt, alvorens je de data naar het html bestand schrijft.

Studio Ghibli Exercise

{ api }

Maak de volgende website na. De data wordt gehaald van het volgende end-point: <https://ghibliapi.herokuapp.com/films>

API Exercise

Menu comes here...



Studio Ghibli

Studio Ghibli Inc. is a Japanese animation studio headquartered in Koganei, Tokyo. It is best known for its animated feature films, and has also produced several short subjects, television commercials, and two television films. Its mascot and most recognizable symbol is a character named Totoro, a giant spirit inspired by raccoon dogs (tanuki) and cats from the 1988 anime film My Neighbor Totoro. Among the studio's highest-grossing films are Spirited Away (2001), Howl's Moving Castle (2004) and Ponyo (2008). The studio was founded on June 15, 1985, by directors Hayao Miyazaki and Isao Takahata and producer Toshio Suzuki, after the successful performance of Topcraft's Nausicaä of the Valley of the Wind (1984). It has also collaborated with video game studios on the visual development of several games.

Studio Ghibli Exercise

{ api }

Het gedeelte 'Movies' staat onder de vorige gedeelte (zie vorige slide). Alle 22 movies worden door de API geleverd.

Movies		
Castle in the Sky The orphan Sheeta inherited a mysterious crystal that links her to the mythical sky-kingdom of Laputa. With the help of resourceful Pazu and a rollicking band of sky pirates, she makes her way to the ruins of the once-great civilization. Sheeta and Pazu must outwit the evil Muska, who plans to use L...	Grave of the Fireflies In the latter part of World War II, a boy and his sister, orphaned when their mother is killed in the firebombing of Tokyo, are left to survive on their own in what remains of civilian life in Japan. The plot follows this boy and his sister as they do their best to survive in the Japanese countryside...	My Neighbor Totoro Two sisters move to the country with their father in order to be closer to their hospitalized mother, and discover the surrounding trees are inhabited by Totoros, magical spirits of the forest. When the youngest runs away from home, the older sister seeks help from the spirits to find her....
Kiki's Delivery Service A young witch, on her mandatory year of independent life, finds fitting into a new community difficult while she supports herself by running an air courier service....	Only Yesterday It's 1982, and Taeko is 27 years old, unmarried, and has lived her whole life in Tokyo. She decides to visit her family in the countryside, and as the train travels through the night, memories flood back of her younger years: the first immature stirrings of romance, the onset of puberty, and the fru...	Porco Rosso Porco Rosso, known in Japan as Crimson Pig (Kurenai no Buta) is the sixth animated film by Hayao Miyazaki and released in 1992. You're introduced to an Italian World War I fighter ace, now living as a freelance bounty hunter chasing 'air pirates' in the Adriatic Sea. He has been given a curse that c...