**Fetch**

Dalam JavaScript kita bisa mengirimkan *network request* dan juga bisa mengambil informasi data terbaru dari *server* jika dibutuhkan.

Contoh *network request* yang biasa kita lakukan:

* Mengirimkan data dari sebuah *form*.
* Mengambil data untuk ditampilkan dalam *list/table*.
* Mendapatkan notifikasi.

Dalam melakukan *network request*, JavaScript memiliki metode bernamafetch().

Proses melakukan fetch() adalah salah satu proses *asynchronous* di JavaScript sehingga kita perlu menggunakan salah satu diantara *promise* atau *async/await*.

**Fetch dengan Promise**

Berikut ini contoh *request* data dengan fetch() menggunakan *promise*:

fetch("https://jsonplaceholder.typicode.com/posts")

.then(function (response) {

return response.json();

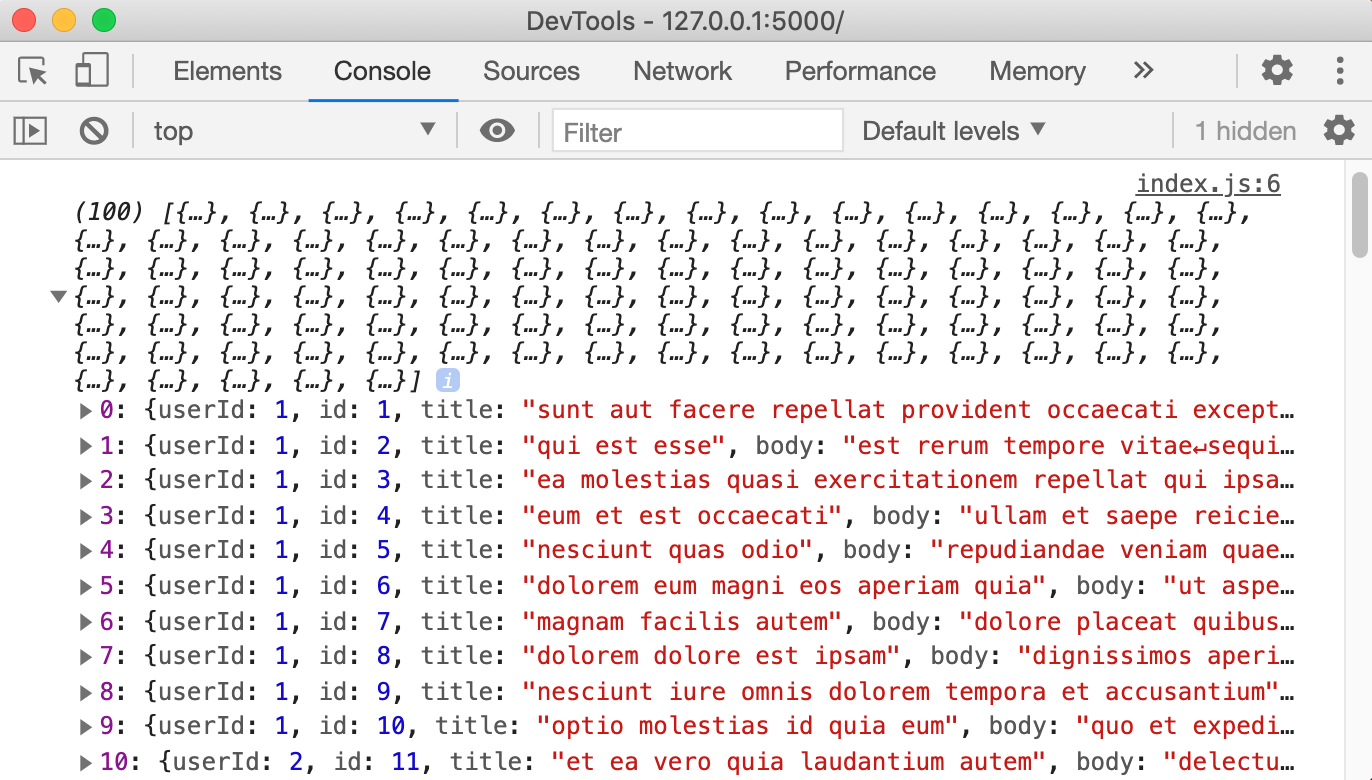
})

.then(function (post) {

console.log(post);

});

Kode di atas mengambil data *end-point* dari API [JSONPlaceholder](https://jsonplaceholder.typicode.com/). Berikut ini tampilan dari console.log() untuk data yang kita panggil:



Gambar di atas adalah data hasil *request* dari fetch() menggunakan *promise* yang kita lakukan.

**Fetch dengan async/await**

Berikut contoh *request* data dengan fetch() menggunakan async/await:

const tesFetchAsync = async () => {

let response = await fetch("https://jsonplaceholder.typicode.com/posts");

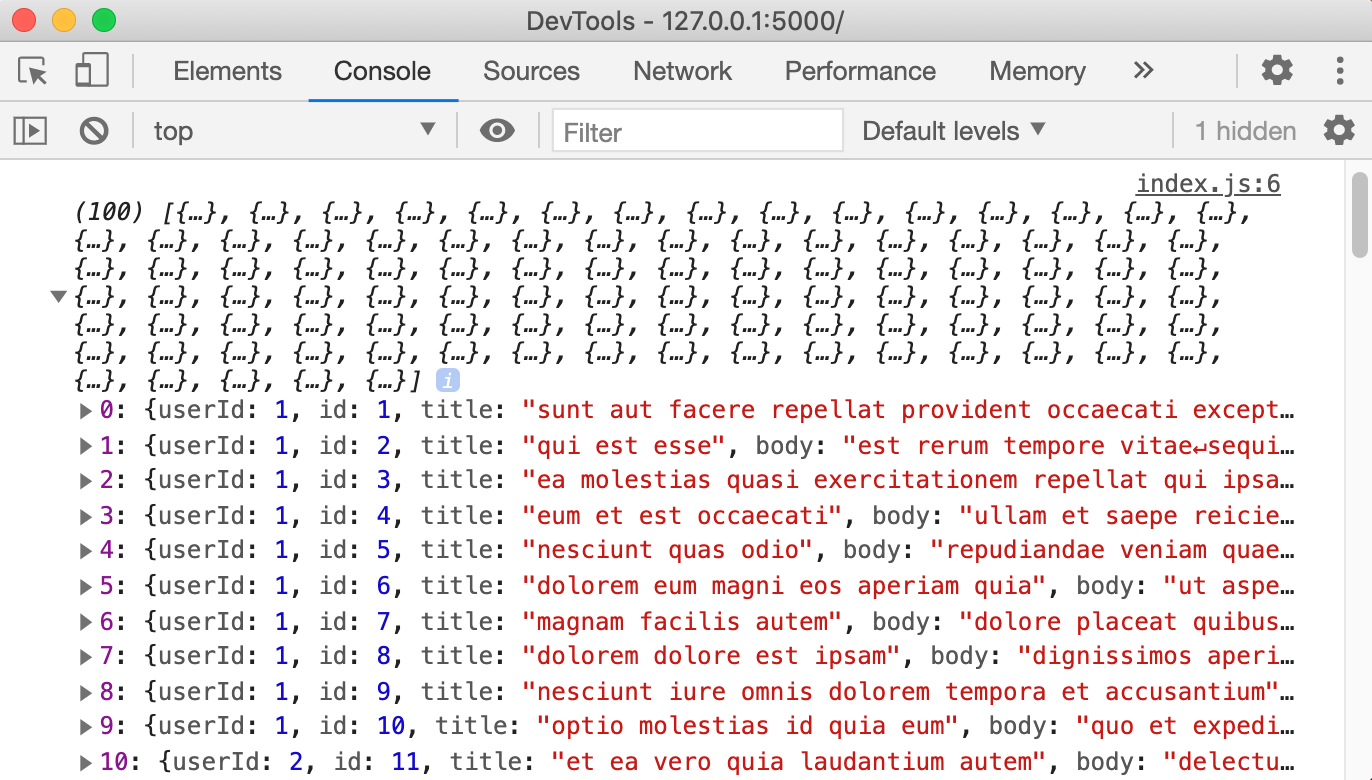
response = await response.json();

console.log(response);

};

tesFetchAsync();

Kita masih mengambil data dari sumber *end-point* yang sama dengan fetch() sebelumnya yang menggunakan *promise* sehingga hasilnya pun masih sama persis seperti sebelumnya.



Dalam penggunaan di dunia kerja dan aplikasi berskala besar kita bisa memilih menggunakan *promise* ataupun *async/await* tetapi kita lihat jika menggunakan *async/await*, kode kita terlihat lebih *clean* dan mudah dibaca.