Introduction à Deno 1.30

Réalisez un backend javascript très facilement avec Deno

Jeudi 02 Février 2022
A202+Discord, 17h30

Présenté par Logan TANN

Pour OnePanthéon

ONE
efrei
PANTHÉON-ASSAS UNIVERSITÉ

Audience

- Pour les curieux du web en général 🤓 -

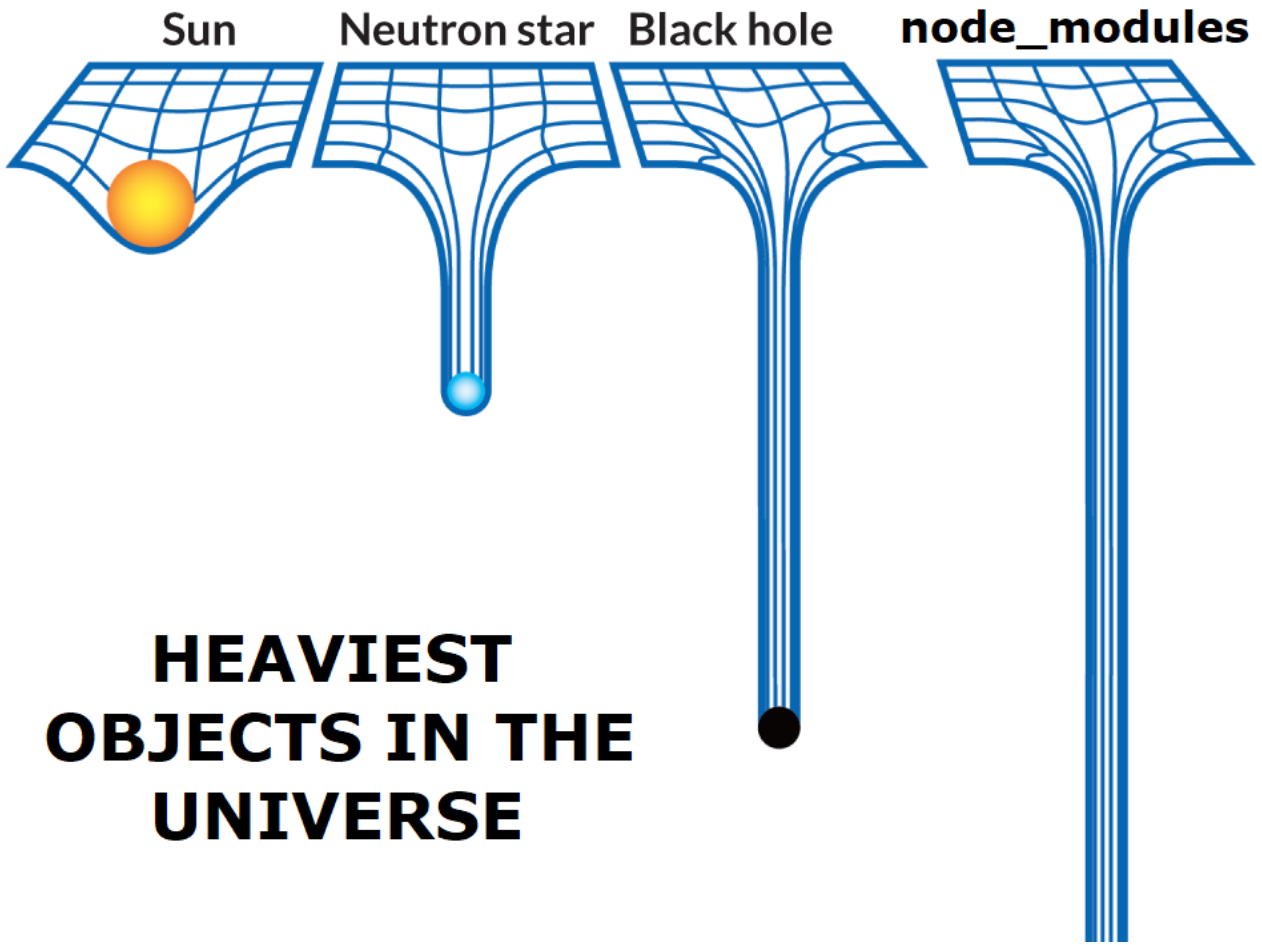
👉 En particulier : les Typescript Enthousiasts, ou pour ceux qui ont réalisé un projet avec NodeJs.

👓 Note : Pour comprendre les exemples, une expérience en Javascript moderne est recommandée.

ONE
efrei
PANTHÉON-ASSAS
UNIVERSITÉ

Un peu d'histoire

> De Node.JS à Deno



ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Les débuts de Node.JS

😳 Ryan Dahl est frustré des limitations d'Apache

- Difficultés d'implémenter des applis en temps réel
- Mécanismes de prog évènementielle et concurrente pas assez efficace.

JSConf 2009 : Il présente un projet perso, qui deviendra Node.JS

👉 Bénéficie aujourd'hui d'une grande popularité

Un projet devenu trop populaire

10 ans plus tard, il présente à la JSConf 2019 :
« 10 Choses que je regrette à propos de Node.JS »

Il évoque par exemple :

- Node n'est pas assez proche des standards JS des navigateur
- NPM souffre de plein de défauts

Prise en main de Deno

Installation et découverte en exemples

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Installation

1 - Juste le binaire à installer depuis github

Installation

1 - Juste le binaire à installer depuis github

2 – Installeur automatique en ligne de commande

Shell (Linux & Mac) :

```
$ curl -fsSL https://deno.land/x/install/install.sh | sh
```

Windows :

```
$ irm https://deno.land/install.ps1 | iex
```

Docker :

```
$ docker run -it denoland/deno:1.28.1
```

Exemple 1 : Hello World

Consigne :

1. Faire un programme qui affiche "Hello World".
2. Faire un programme qui affiche le contenu du fichier "canard.txt"

```
~/p/A/0/solutions  main ↑1  $ deno run --allow-read .\01_HelloWorld.js  4ms
Hello World !

   _       _       _
 >(.)__  <(.)__  =(.)__
  (___/   (___/   (___/
```

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Exemple 1 : Hello World

Solution :

```
console.log("Hello World !");

// support du top level await
//  ...  et aussi un système de permissions vraiment bien foutu.
// avant ça faisait un throw, mais depuis la version 1.28, ça a changé
const texte = await Deno.readTextFile("canard.txt");

console.log(texte);
```

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Exemple 2 : Appel d'API

1. Voici une API très sympa : https://catfact.ninja/fact
2. Faire appel à cette api en utilisant Typescript et fetch
3. Ne rendre l'affichage du message que si l'utilisateur vous autorise

```
~/p/A/0/solutions main ↑1 $ deno run --allow-net .\02_ApiCall.ts
Voulez vous écouter un fait sur les chats ?  [y/N] y

Voici un fait sur les chats :
> 70% of your cat's life is spent asleep.
```

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Exemple 2 : Appel d'API

Solution :

```ts
// typescript et ES Module support out of the box
import {CAT_ENDPOINT, CatResponse} from "./catfact.ts";

// Les APIs web sont supportées :D
const request = await fetch(CAT_ENDPOINT);
const response: CatResponse = await request.json();
const message = response.fact;

const agreed: boolean = confirm("Voulez vous écouter un fait sur les chats ? ");
if (agreed) {
    console.log("\nVoici un fait sur les chats : ");
    console.log(">", message);
} else {
    console.log("\nBon temps pis ... ");
}
```

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Exemple 3 : Serveur web et NPM

1. Importer la librairie Aqua (https://deno.land/x/aqua@v1.3.5/mod.ts) et faire un hello world
2. Importer une librairie NPM (npm:qrcode@1.5.0) et afficher un QR Code
3. Utiliser un import map pour rendre l'importation des dépendances plus maintenables

Exemple 3 : Appel d'API (Solution)

```ts
import Aqua from "aqua";
import QRCode from "qrcode";

console.log("Démarrage du serveur ... ");

const aqua = new Aqua(3001);
aqua.get("/", async ()=> {
    const url = await QRCode.toDataURL('https://www.onepantheon.fr/');
    return `
        <H1>Hello world</H1>
        <p>Un super site : <br>
            <img src="${url}" />
        </p>
    `;
});

console.log("Serveur web démarré sur http://localhost:3001");
```

01 - Introduction > solutions > {} deno.json > ...

```json
{
    "imports": {
        "aqua": "https://deno.land/x/aqua@v1.3.5/mod.ts",
        "qrcode": "npm:qrcode@1.5.1"
    }
}
```

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Récap

😀 Avantages :

1. Expérience développeur très satisfaisante.
2. Un binaire standalone avec plein d'outils built-in
3. Veut faire changer les habitudes.

😢 Inconvénients :

1. Difficile de faire changer les habitudes
2. Librairies souvent peu documentées ou supportées
3. VsCode est le seul éditeur fiable

=> Génial pour des scripts d'automatisation ou des petits projets

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Pour aller plus loin...

- **Documentation** : https://deno.land/manual@v1.28.1/introduction
- **Nouveautés Deno** : https://deno.com/blog/
- **DenoDeploy** : plateforme de déploiement on the edge (gratuit)
- **Fresh** : framework web fullstack concurrent à React
- **Deno News** : Veille numérique officielle sur Deno

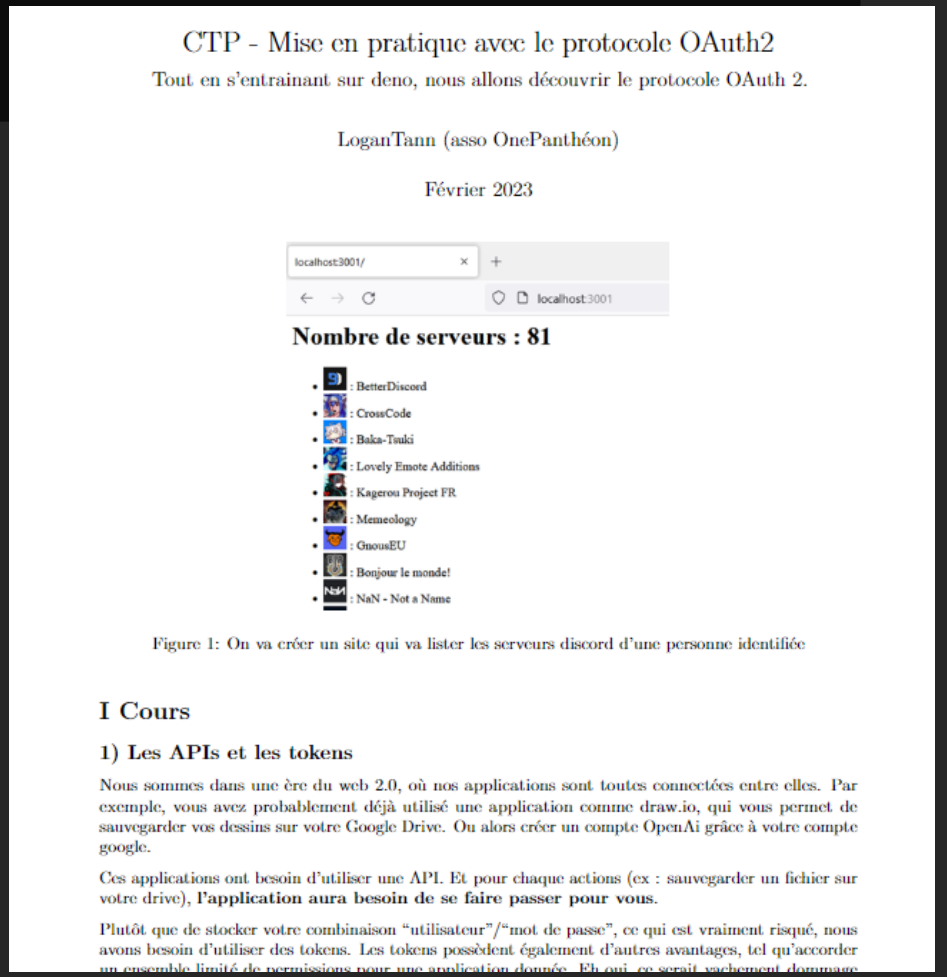
- **Discordeno** : Lib pour faire des bots discord
- **Caviar** : Game Engine basé sur le WebGPU
- **Awesome deno** sur github

TP

Pratique avec le protocole Oauth2

Sujet : Faire une application listant les différents serveurs discord d'une personne

Date de rendu: 02 mars (dans un mois)



CTP - Mise en pratique avec le protocole OAuth2

Tout en s'entrainant sur deno, nous allons découvrir le protocole OAuth 2.

LoganTann (asso OnePanthéon)

Février 2023

localhost:3001/

localhost:3001

Nombre de serveurs : 81

- : BetterDiscord
- : CrossCode
- : Baka-Tsuki
- : Lovely Emote Additions
- : Kagerou Project FR
- : Memeology
- : GnousEU
- : Bonjour le monde!
- : NaN - Not a Name

Figure 1: On va créer un site qui va lister les serveurs discord d'une personne identifiée

I Cours

1) Les APIs et les tokens

Nous sommes dans une ère du web 2.0, où nos applications sont toutes connectées entre elles. Par exemple, vous avez probablement déjà utilisé une application comme draw.io, qui vous permet de sauvegarder vos dessins sur votre Google Drive. Ou alors créer un compte OpenAi grâce à votre compte google.

Ces applications ont besoin d'utiliser une API. Et pour chaque actions (ex : sauvegarder un fichier sur votre drive), **l'application aura besoin de se faire passer pour vous**.

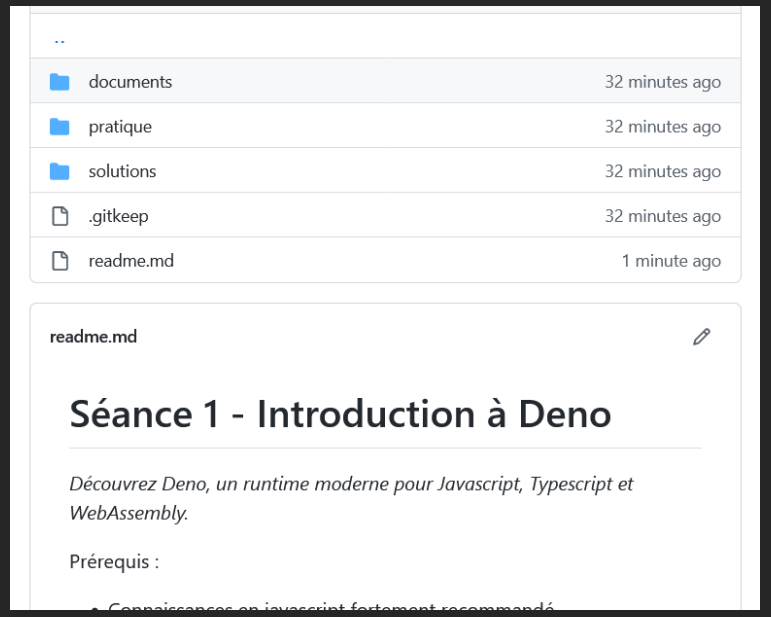
Plutôt que de stocker votre combinaison "utilisateur"/"mot de passe", ce qui est vraiment risqué, nous avons besoin d'utiliser des tokens. Les tokens possèdent également d'autres avantages, tel qu'accorder un ensemble limité de permissions pour une application donnée. Eh oui, ce serait vachement dommage

ONE efrei PANTHÉON-ASSAS UNIVERSITÉ

Des questions ?

Merci pour votre écoute !

Lien github vers les slides et le code :
https://github.com/OnePantheon/Ateliers-deno/tree/main/01 - Introduction



ONE efrei PANTHÉON-ASSAS UNIVERSITÉ