

The background is a deep purple space scene. In the top left is a large planet with horizontal stripes. To its right is a smaller planet with a ring. In the bottom left, an astronaut floats with a coiled tether. In the bottom right is a large, cratered moon. The sky is filled with numerous small white stars and larger four-pointed starbursts. Abstract, wavy shapes in shades of purple and blue drift across the background.

# SEMAINE 02

## HTTP et REST

Services d'échange de données



# Plan de séances

- ★ Retour Semaine 01
- ★ Organisation de notre code
- ★ Protocole HTTP
- ★ Architecture REST
- ★ Début de Space Explorer API

# Retour Semaine 01

- ★ Installation de NVM et node.js
- ★ Langage javascript
- ★ Premier serveur
- ★ Des questions?

The background is a deep purple and blue space scene. In the top left, there's a large planet with horizontal stripes. Below it is a smaller planet with a ring. In the top right, an astronaut in a white suit is floating, holding a long, thin, looping rope. In the bottom right, there's a large, cratered moon. The background is filled with numerous small white stars and larger, four-pointed starburst shapes. There are also some soft, wavy, light blue and purple shapes that look like nebulae or gas clouds.

# Structure de notre code



# GitHub

- ★ Création de mon repository
- ★ Clone

# NPM

- ★ Node package manager
- ★ <https://www.npmjs.org/>
- ★ Ce que vous voulez existe peut-être (sûrement) déjà

# NPM



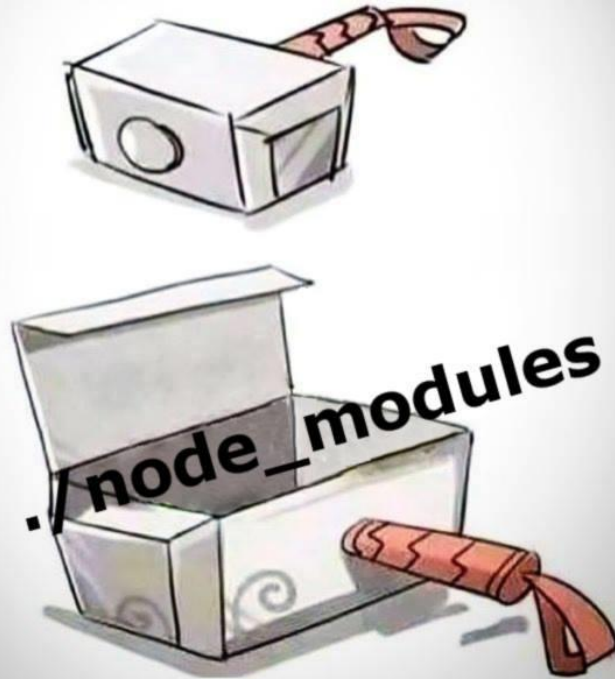
`npm install`

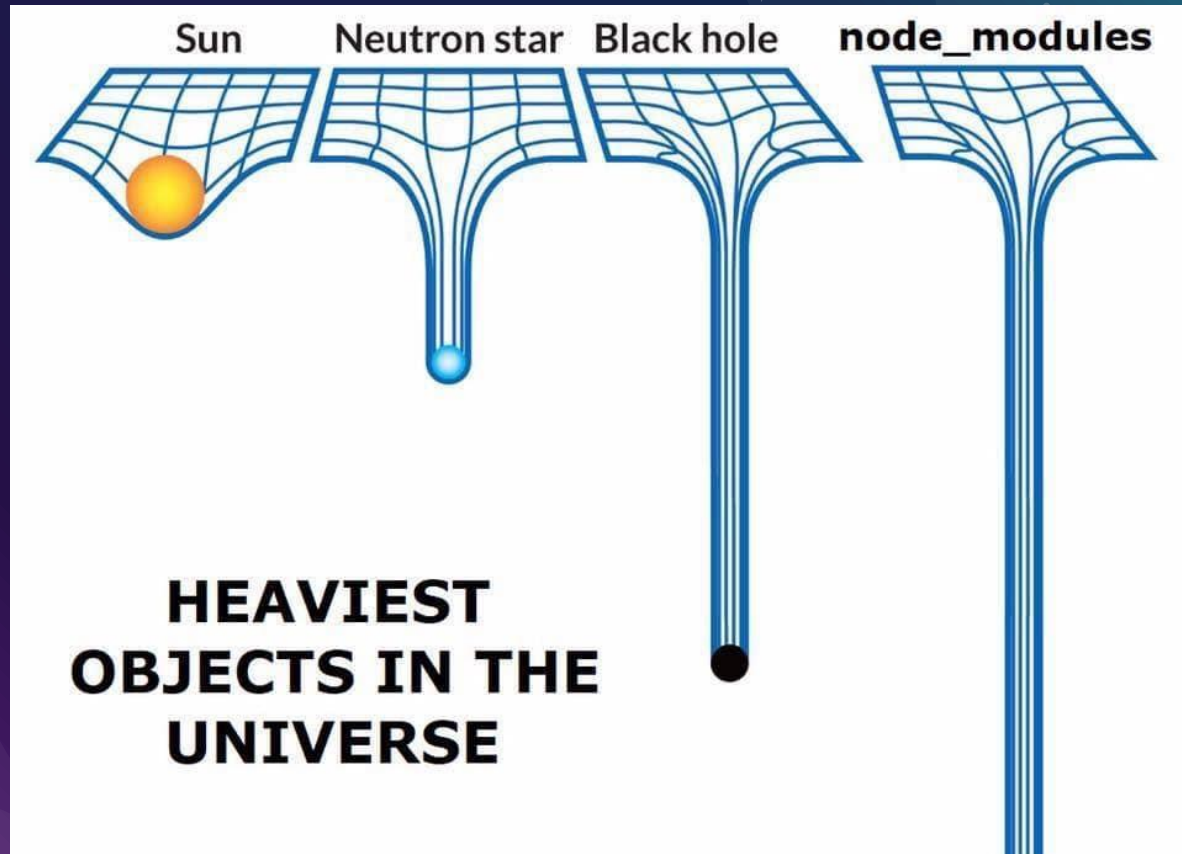
# package.json

- ★ `npm init`
- ★ Fichier de description du projet (package.json)
- ★ Comprend les dépendance du projet
- ★ `npm install [module]`



# The secret behind Thor's hammer





# Modules à installer

- ★ chalk
- ★ express

# Structure du projet

- ★ Basé sur
  - ★ 5 années d'expérience avec node.js
  - ★ recherches personnelles
- ★ Début de l'utilisation de l'orienté objet en JS
- ★ Si vous avez des idées, je suis preneur

# Fichiers et dossiers

- ★ `server.js`
- ★ `.gitignore`
- ★ `/src`
- ★ `/src/app.js`

# Routes et endpoint

- ★ Route: le chemin à suivre pour accéder à un endpoint.
- ★ Endpoint: Point terminal de la route début de notre code.

# Protocole HTTP

The background is a deep space scene. In the top left, there's a large planet with horizontal stripes. Below it is a smaller planet with a ring. In the top right, an astronaut in a white suit floats, holding a long, thin, looping rope. In the bottom right, there's a cratered moon. The sky is dark blue and purple, filled with numerous small white stars and larger, four-pointed starburst shapes. There are also some soft, wavy, translucent shapes in shades of purple and blue, suggesting nebulae or gas clouds.

# HTTP

- ★ Protocole de communication
- ★ Couche application du modèle OSI
- ★ Ports 80 ou 443
- ★ Concept de requête et de réponse



# URL

scheme:[//authority]path[?query][#fragment]

[userinfo@]host[:port]

scheme

http

host

//cstj.qc.ca

path

/etudiants

query

?programme=420

# Méthodes HTTP

- ★ Paramètre supplémentaire à fournir
  - ★ GET, PUT, POST, DELETE, PATCH
- ★ Représente une action à accomplir

# Request

[body]

method + http version  
host/path?query  
[headers]

GET HTTP/2  
[www.google.ca?q=test](http://www.google.ca?q=test)  
User-Agent: Firefox

# Request

| Method              | URL                  | HTTP Version |
|---------------------|----------------------|--------------|
| Headers             |                      |              |
| Body (optionnel)    |                      |              |
| GET                 | www.google.ca?q=test | HTTP/2       |
| User-Agent: Firefox |                      |              |
|                     |                      |              |

# Response

[body]

expéditeur  
version + status code  
headers

```
<html>  
<head>...</head>  
<body>...</body>  
</html>
```

HTTP/2 200 OK  
Content-Type: text/html  
Date: 2020-09-06

# Response

|             |              |
|-------------|--------------|
| Status Code | HTTP Version |
| Headers     |              |
| Body        |              |

|   |        |
|---|--------|
| 200 OK  | HTTP/2 |
| Content-Type: text/html<br>Date: 2020-09-06               |        |
| <html><br><head>...</head><br><body>...</body><br></html> |        |

# Code HTTP

- ★ 1xx: Information
- ★ 2xx: Succès
- ★ 3xx: Redirection
- ★ 4xx: Erreur du client
- ★ 5xx: Erreur du serveur

# Les plus utilisés

| Succès  | Erreur client  | Erreur serveur  |
|---|--|---|
| <b>200:</b> OK<br><b>201:</b> Created<br><b>204:</b> No Content | <b>400:</b> Bad Request<br><b>401:</b> <i>Unauthorized*</i><br><b>402:</b> Payment Required<br><b>403:</b> <i>Forbidden*</i><br><b>404:</b> Not Found<br><b>405:</b> Method Not Allowed<br><b>409:</b> Conflit<br><b>415:</b> Unsupported Media Type<br><b>422:</b> Unprocessable Entity | <b>500:</b> Internal Server Error<br><b>501:</b> Not Implemented<br><b>503:</b> Service Unavailable |





# Architecture REST

# Architecture REST

- ★ Representational State Transfer
- ★ Ce n'est pas une technologie
- ★ Ce n'est pas un standard
- ★ Ce n'est pas un cadre d'application
- ★ C'est un ensemble de bonnes pratiques

# Ressources

- ★ Le contenu offert par notre service d'échange
  - ★ Texte, vidéo, image, html, ...
- ★ Utiliser un nom pour identifier une ressource
- ★ URL pour accéder à la ressource

# Collection et Document

- ★ Collection
  - ★ Ensemble des ressources d'un même type
  - ★ []
- ★ Document
  - ★ Un élément de la collection
  - ★ {}

# Représentations

- ★ La manière dont les ressources peuvent être visualisées.
- ★ Format : JSON, XML, PNG, PDF, ...

# Représentations

```
<teacher>  
  <id>06476</id>  
  <firstname>Yannick</firstname>  
  <name>Charron</name>  
</teacher>
```

```
{  
  "id": "06476",  
  "firstname": "Yannick",  
  "name": "Charron",  
}
```



# Une bonne représentation

- ★ 3 critères
  - ★ Compréhensibilité
  - ★ Complétude
  - ★ Lien

# Concept de répertoire

```
etudiants
├── 0533563
│   └── cours
├── 1458962
│   └── cours
└── 1932112
    └── cours
```

```
[{
  "id": "0533563",
  "firstname": "Yannick",
  "name": "Charron",
  "cours": [{}, {}, {}]
},
{
  "id": "1458962",
  "firstname": "Robert",
  "name": "Turenne",
  "cours": [{}, {}, {}]
},
{
  "id": "1932112",
  "firstname": "Karine",
  "name": "Moreau",
  "cours": [{}, {}, {}]
}]
```



# REST et URL

- ★ Utiliser le pluriel pour nommer les ressources
- ★ Utiliser des minuscules
- ★ Si nécessaire, utiliser des tirets (-) dans les noms de ressources
- ★ Utilisation des méthodes HTTP, pour définir les opérations sur une ressource

# Lien entre HTTP et REST

Utilise les méthodes HTTP pour les actions

- ★ GET = Obtenir une ressource
- ★ PUT = Modifier une ressource
- ★ POST = Créer une ressource
- ★ DELETE = Supprimer une ressource
- ★ PATCH = Modifier partiellement une ressource



**Postman**



# Début de SpaceExplorer API

# Semaine prochaine

- ★ Suite de REST
- ★ Ajout d'une base de données NoSQL