

Création d'une API REST à l'aide de NodeJS

Infos concernant cette activité

- Cette activité est-elle notée ?
 - Oui, la note de cette activité fait 50% de la note de la première partie de la ressource R301 (backend).
- Quand dois-je terminer cette activité ?
 - Après 4 semaines. La date limite sera le vendredi 28 octobre 2022 (avant de commencer la deuxième partie - front end - de cette ressource).
- Comment puis-je soumettre mon projet ?
 - Ouvrir Moodle
 - Rechercher le cours : BUT_INFO_S3 R3.01 dev. web - côté serveur (jazar)
 - Clé: webdevserver2022
 - Ouvrir "Activité : Création d'une API REST à l'aide de NodeJS"

Ressources supplémentaires:

- Projet de la semaine 1 pour lire/écrire dans un fichier json.
- Projet de la semaine 2 pour la création d'un serveur web avec le pattern Routers-Controllers-Services.
- Tutoriels sur EJS et Handlebars postés sur cours-info sous la semaine 4.
- Travailler avec des tableaux JSON :
 - <https://restfulapi.net/json-array/>
 - <https://www.javatpoint.com/es6-array-methods>

Dans ce projet, vous allez créer une API REST avec NodeJS pour servir les données d'une base de données JSON.

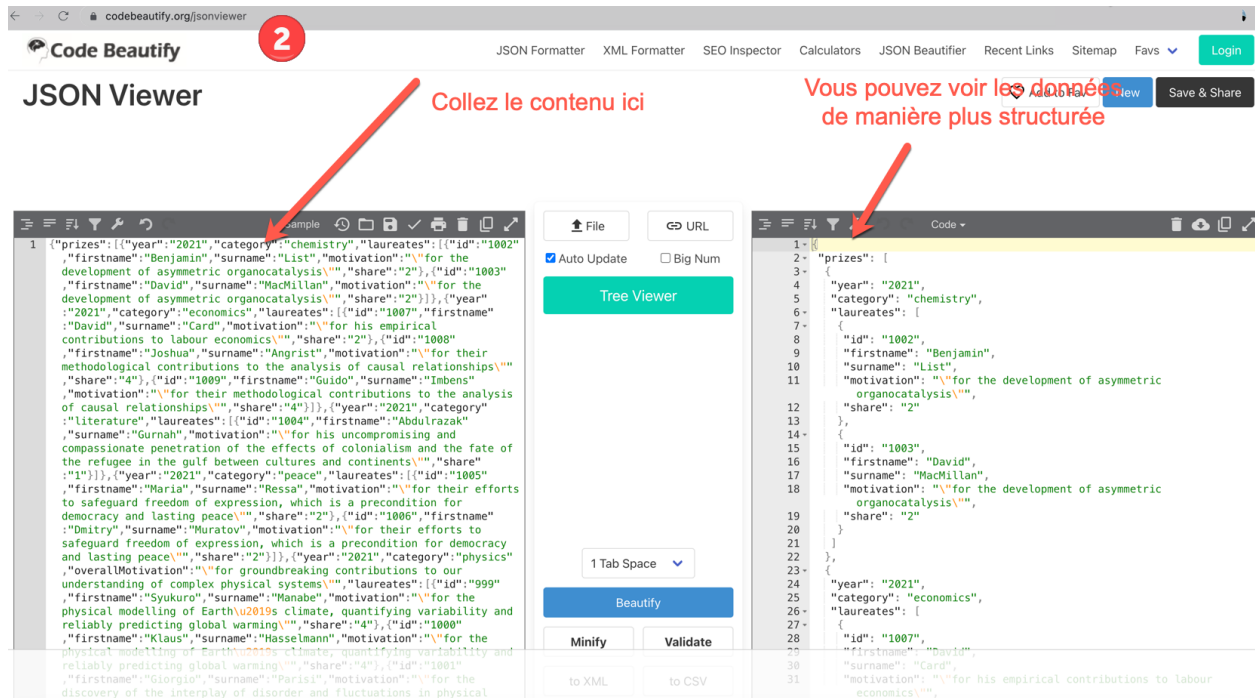
Base de données : **prize.json**

La base de données à utiliser dans ce projet est un fichier JSON qui contient des informations sur les prix Nobel et les lauréats du prix Nobel. Le nom du fichier est **prize.json**. Vous pouvez le télécharger sur cours-info sous la semaine 4.

Pour mieux comprendre ce que contient ce fichier, ouvrez-le dans un éditeur de texte, copiez le contenu, ouvrez un site Web Formateur JSON (JSON beautifier) tel que <https://codebeautify.org/jsonviewer> et collez les données JSON.



```
{
  "prizes": [
    {
      "year": "2021",
      "category": "chemistry",
      "laureates": [
        {
          "id": "1002",
          "firstname": "Benjamin",
          "surname": "List",
          "motivation": "\nfor the development of asymmetric organocatalysis",
          "share": "2"
        },
        {
          "id": "1003",
          "firstname": "David",
          "surname": "MacMillan",
          "motivation": "\nfor the development of asymmetric organocatalysis",
          "share": "2"
        }
      ]
    },
    {
      "year": "2021",
      "category": "economics",
      "laureates": [
        {
          "id": "1007",
          "firstname": "David",
          "surname": "Card",
          "motivation": "\nfor his empirical contributions to labour economics",
          "share": "2"
        },
        {
          "id": "1008",
          "firstname": "Joshua",
          "surname": "Angrist",
          "motivation": "\nfor their methodological contributions to the analysis of causal relationships",
          "share": "4"
        },
        {
          "id": "1009",
          "firstname": "Guido",
          "surname": "Imbens",
          "motivation": "\nfor their methodological contributions to the analysis of causal relationships",
          "share": "4"
        }
      ]
    },
    {
      "year": "2021",
      "category": "literature",
      "laureates": [
        {
          "id": "1004",
          "firstname": "Abdulrazak",
          "surname": "Gurnah",
          "motivation": "\nfor his uncompromising and compassionate penetration of the effects of colonialism and the fate of the refugee in the gulf between cultures and continents",
          "share": "1"
        }
      ]
    },
    {
      "year": "2021",
      "category": "peace",
      "laureates": [
        {
          "id": "1005",
          "firstname": "Maria",
          "surname": "Ressa",
          "motivation": "\nfor their efforts to safeguard freedom of expression, which is a precondition for democracy and lasting peace",
          "share": "2"
        },
        {
          "id": "1006",
          "firstname": "Dmitry",
          "surname": "Muratov",
          "motivation": "\nfor their efforts to safeguard freedom of expression, which is a precondition for democracy and lasting peace",
          "share": "2"
        }
      ]
    },
    {
      "year": "2021",
      "category": "physics",
      "overallMotivation": "\nfor groundbreaking contributions to our understanding of complex physical systems",
      "laureates": [
        {
          "id": "999",
          "firstname": "Syukuro",
          "surname": "Manabe",
          "motivation": "\nfor the physical modelling of Earth's climate, quantifying variability and reliably predicting global warming",
          "share": "4"
        },
        {
          "id": "1000",
          "firstname": "Klaus",
          "surname": "Hasselmann",
          "motivation": "\nfor the physical modelling of Earth's climate, quantifying variability and reliably predicting global warming",
          "share": "4"
        },
        {
          "id": "1001",
          "firstname": "Giorgio",
          "surname": "Parisi",
          "motivation": "\nfor the discovery of the interplay of disorder and fluctuations in physical systems from atomic to planetary scales",
          "share": "2"
        }
      ]
    },
    {
      "year": "2021",
      "category": "medicine",
      "laureates": [
        {
          "id": "997",
          "firstname": "David",
          "surname": "Julius",
          "motivation": "\nfor their discoveries of receptors for temperature and"
        }
      ]
    }
  ]
}
```



Ce fichier JSON contient un tableau "prizes" avec 658 objets. Chaque objet contient l'année, la catégorie du prix, et les gagnants stockés à l'intérieur d'un tableau "laureates".

```

▼ object {}
  ▼ prizes [658]
    ▼ 0 {}
      year : 2021
      category : chemistry
      ▼ laureates [2]
        ▼ 0 {}
          id : 1002
          firstname : Benjamin
          surname : List
          motivation : \"for the development of asymmetric organocatalysis\"
          share : 2
        ▼ 1 {}
          id : 1003
          firstname : David
          surname : MacMillan
          motivation : \"for the development of asymmetric organocatalysis\"
          share : 2

```

la liste de chaque prix par année et les lauréats regroupés.

Indice: Vous pouvez supprimer **{"prizes":** du début du fichier et la lettre **}** de la fin du fichier si vous trouvez qu'il est plus facile de parser le fichier de cette façon.

Tâches à faire :

- Créer un nouveau projet NodeJS.
- Créez une API REST à l'aide du modèle RCS ou RCSM.
 - RCS: Routers - Controllers - Services
 - RCSM: Routers - Controllers - Services - Models
- Essayez de vous référer aux travaux effectués au cours des trois premières semaines pour structurer votre application.
 - Sem 1: Lire/Ecrire dans un fichier
 - Sem 2: Express + RCS
 - Sem 3: Express + RCSM
- Créez le point d'entrée de l'API, les routeurs, les contrôleurs et les services (ou modèles). Faites attention que vous pouvez avoir plusieurs routeurs. Pensez au nombre d'entités que contient ce document JSON.

REST API:

- **Votre API doit offrir les fonctionnalités suivantes :**
 - **F1:** Lister tous les lauréats (id, prénom, nom).
 - Pensez à la pagination.
 - Attention aux doublons !!
 - **F2:** Étant donné un identifiant, affichez les informations du lauréat avec cet identifiant (prénom, nom).
 - **F3:** Comptez le nombre de prix offerts.
 - Attention, il y a des années où aucun prix Nobel n'a été décerné.
 - Attention le nombre de prix offerts n'est pas le même que le nombre de lauréats. Par exemple en 2021, 13 personnes ont remporté des prix Nobel en chimie, physique, économie, paix, médecine et littérature. Le nombre de prix doit être de 6 (un pour chaque catégorie) et non pas 13.
 - **F4:** Comptez le nombre de lauréats qui ont reçu des prix nobel.
 - Attention aux doublons !!
 - **F5:** Combien ont remporté plus d'un prix Nobel ?
 - (prénom, nom, nombre de prix gagnés)
 - Ex: Marie Curie, 2
 - **F6:** Lister toutes les catégories des prix nobel
 - Chemistry, economics, etc
 - **F7:** Déterminez quelle catégorie a produit le plus grand nombre de lauréats du prix Nobel.
 - **F8:** Pour chaque année, indiquez combien de lauréats avaient remporté un prix nobel.
 - Ex: 2021, 13

- **F9:** Pour un identifiant de lauréat donné, affichez les prix remportés (prénom, nom, année, catégorie et motivation).
 - Ex: ID = 6
 - Marie Curie
 - 1911 chemistry in recognition of her services to the...
 - 1903 physics in recognition of the extraordinary services...
- **F10:** Afficher toutes les années au cours desquelles aucun prix Nobel n'a été décerné.
- **F11:** Afficher toutes les années de prix nobel triées par nombre de lauréats ascendant/descendant.
 - ?sort=+laureates ⇒ ascendant ⇒ commencer par les années avec le plus petit nombre de lauréats
 - ?sort=-laureates ⇒ descendant ⇒ commencer par les années avec le plus grand nombre de lauréats
 - Exclure les années où aucun prix Nobel n'a été décerné
- **F12:** À partir du prénom, ou du nom, ou de la catégorie, affichez tous les lauréats qui correspondent au filtre.
 - Nous devrions pouvoir filtrer par différents champs.
- **F13:** Supprimer un lauréat avec un identifiant donné dans une année donnée et une catégorie donnée.
- **F14:** Mettre à jour la motivation d'un lauréat avec un identifiant donné dans une année donnée et une catégorie donnée.
- **F15:** Ajouter un nouveau lauréat à une année donnée et à une catégorie donnée.
 - Si le prénom et le nom existent, utilisez l'identifiant du lauréat existant. Sinon, créez un nouvel identifiant unique (ex: max(id) + 1).

Documentation:

Ajoutez une documentation Swagger à toutes les fonctionnalités F1-F15. Dédiez la route **/api-docs** pour la documentation Swagger.

Vues:

- Créez un modèle (template) à l'aide de Handlebars ou EJS, dans lequel il y a une liste déroulante avec toutes les catégories remplies. Lorsque l'utilisateur choisit une catégorie

dans la liste, affichez tous les lauréats du prix Nobel de cette catégorie (prénom, nom, année).

Cette liste déroulante doit être remplie dynamiquement.

Physics
Chemistry
Medicine
Literature
Economics
Peace

Lorsqu'une catégorie est choisie, le tableau se remplit avec les lauréats correspondant à cette catégorie.

First Name	Last Name	Year

- Créez un modèle (template) à l'aide de Handlebars ou EJS avec un formulaire dans lequel vous pouvez créer un nouveau lauréat pour une année et une catégorie données.
 - Si le prénom et le nom existent, utilisez l'identifiant du lauréat existant. Sinon, créez un nouvel identifiant unique (ex: $\max(\text{id}) + 1$).
 - Envisagez un middleware de validation.
 - Ex: L'utilisateur ne peut pas entrer une année qui n'existe pas dans le fichier JSON.
 - Prénom et nom supérieurs à 3 caractères.
 - Motivation pas vide.

Cette liste déroulante doit être remplie dynamiquement.

Physics
Chemistry
Medicine
Literature
Economics
Peace

Year

First name

Surname

Motivation

Save