

# TP 3: Création d'une API REST avec NodeJS



Dr Mad généré par une IA

## Objectifs

L'objectif de ce TP est de vous familiariser avec la création d'API REST en utilisant Node.js et Express. Vous apprendrez à gérer des routes, des contrôleurs et des services, et à effectuer diverses opérations telles que GET, PUT, DELETE, et POST, ainsi que le filtrage et le tri des données.

## Introduction

Ce TP3 est la continuation de notre série de travaux pratiques. Dans TP2, vous avez appris les bases de Node.js et Express. Maintenant, nous allons approfondir ces connaissances en créant une API REST complète.

## Mise en place du projet

- Créez un répertoire datasource et placez-y le fichier "data.js" fourni.
- Structurez votre application en utilisant le modèle "Routeurs-Contrôleurs-Services".

### *data.js:*

```
const items=[
{"_id":"646cee044f07a537bf388bcb","links":["https://fr.wikipedia.org/wiki/Variole"],"name":"variole","description":"varirole","stock":0,"wait":{"$date":"2023-05-23T16:47:00.354Z"},"sold":false,"price":75000,"promotion":[],"object":{"code":"ACBCBBACDDCADDBCDDCABBBCADDCCADDBCD"}},
{"_id":"646cee044f07a537bf388bc9","links":["https://fr.wikipedia.org/wiki/Prion_(prot%C3%A9ine)"],"name":"prion","description":"prion","stock":1,"wait":{"$date":"2023-05-23T16:47:00.353Z"},"sold":true,"price":10000,"promotion":[{"_id":"646cee044f07a537bf388bca","discount":10,"amount":2}],object":{"code":"ABCACACABCAB"}},
{"_id":"646cee044f07a537bf388bcc","links":["https://fr.wikipedia.org/wiki/Virus_Ebola"],"name":"ebola","description":"ebola","stock":0,"wait":{"$date":"2023-05-23T16:47:00.354Z"},"sold":true,"price":65000,"promotion":[],"object":{"code":"AAAABBBBCCCCDDDDDDDDCCCB BBBBAAA"}},
{"_id":"646cee044f07a537bf388bbf","links":["https://fr.wikipedia.org/wiki/Adenoviridae"],"name":"adeno","description":"adenovirus","stock":10,"wait":{"$date":"2023-05-23T16:47:00.348Z"},"sold":true,"price":5000,"promotion":[{"_id":"646cee044f07a537bf388bc0","discount":5,"amount":2},{_id":"646cee044f07a537bf388bc1","discount":10,"amount":5}],object":{"code":"ABABABABABAB"}},
{"_id":"646cee044f07a537bf388bc2","links":["https://fr.wikipedia.org/wiki/SARS-CoV-2"],"name":"covid","description":"covid","stock":50,"wait":{"$date":"2023-05-23T16:47:00.352Z"},"sold":true,"price":1000,"promotion":[{"_id":"646cee044f07a537bf388bc3","discount":10,"amount":10},{_id":"646cee044f07a537bf388bc4","discount":20,"amount":50}],object":{"code":"ABADBADCCCBADCBABADBADCCCBADCB"}},
{"_id":"646cee044f07a537bf388bc5","links":["https://fr.wikipedia.org/wiki/Staphylococcus"],"name":"staphy","description":"staphylocoque","stock":100,"wait":{"$date":"2023-05-23T16:47:00.352Z"},"sold":true,"price":500,"promotion":[{"_id":"646cee044f07a537bf388bc6","discount":5,"amount":5},{_id":"646cee044f07a537bf388bc7","discount":10,"amount":10},{_id":"646cee044f07a537bf388bc8","discount":20,"amount":50}],object":{"code":"ABBCDDDDCBBB"}},
]

const shopusers=[
{"_id":"646cee044f07a537bf388be3","name":"tester","login":"tester","password":"$2a$10$SIw00Y4vXE05YbXZ7vN/eDtJS7fN3YB9mbyc4SDYJ8uRtNLLr6aa","email":"tester@nowhere.org","session":"00000000-0000-0000-0000-000000000000","orders":[]},
]
```

```
{ "_id": "646cee044f07a537bf388be6", "name": "doctor  
mad", "login": "drmad", "password": "$2a$10$X6zMvs4XkZn2VMF0MSpLzuPamrnCiKEx2JWG6sDGRE4trqv48SPRxq", "e  
mail": "drmad@bad.org", "orders": [] },  
]  
  
const orders=[  
{ "_id": "646cee044f07a537bf388beb", "items": [ { "_id": "646cee044f07a537bf388bec", "item": { "name": "aden  
o", "description": "adenovirus", "price": 5000, "promotion": [ { "_id": "646cee044f07a537bf388bc0", "discou  
nt": 5, "amount": 2 }, { "_id": "646cee044f07a537bf388bc1", "discount": 10, "amount": 5 } ], "object": {  
  \ "code\ "  
  \ "ABABABABABAB\ " } }, "amount": 2 }, { "_id": "646cee044f07a537bf388bef", "item": { "name": "covid", "descrip  
tion": "covid  
19", "price": 1000, "promotion": [ { "_id": "646cee044f07a537bf388bc3", "discount": 10, "amount": 10 }, { "_id"  
: "646cee044f07a537bf388bc4", "discount": 20, "amount": 50 } ], "object": { \ "code\ "  
  \ "ABADBADCBCBADCBABADBADCBCBADCB\ " } }, "amount": 1 } ], "date": { "$date": "2023-05-23T16:47:00.601Z" }, "t  
otal": 10500, "status": "finalized", "uuid": "01010101-0000-0000-0000-010101010101" },  
]  
  
const bankaccounts=[  
{ "_id": "646cee044f07a537bf388bd3", "number": "FRBAD0RG78901234567890-0000001", "amount": 1000000 },  
{ "_id": "646cee044f07a537bf388bd6", "number": "FRBAD0RG78901234567890-0000002", "amount": 3000000 },  
{ "_id": "646cee044f07a537bf388bd9", "number": "FRDRMAD578901234567890-0000666", "amount": 2000000 },  
]  
  
const transactions=[  
{ "_id": "646cee044f07a537bf388bdc", "amount": -10500, "account": "646cee044f07a537bf388bd9", "date": { "$  
date": "2023-05-23T16:47:00.445Z" }, "uuid": "99990000-0000-0000-0000-000000000001" },  
{ "_id": "646cee044f07a537bf388bde", "amount": 75000, "account": "646cee044f07a537bf388bd9", "date": { "$d  
ate": "2023-05-23T16:47:00.456Z" }, "uuid": "99990000-0000-0000-0000-000000000002" },  
{ "_id": "646cee044f07a537bf388be0", "amount": -75000, "account": "646cee044f07a537bf388bd3", "date": { "$  
date": "2023-05-23T16:47:00.457Z" }, "uuid": "99990000-0000-0000-0000-000000000003" },  
]  
  
const auctionusers=[  
{ "_id": "646cee044f07a537bf388bf4", "pseudo": "drmad", "password": "$2a$10$xcdgIsyfcZI105FhFu9jIeI4y0v  
ZGNdKr7rIa5pav0rcoceGy8SdG", "email": "drmad@bad.org", "account": "FRDRMAD578901234567890-0000666" },  
{ "_id": "646cee044f07a537bf388bf7", "pseudo": "crime1", "password": "$2a$10$QpMyRfQoAR/DCKRR06U5pepw.M  
JBVHxzVxSkwx/k95SImDB93AS2.", "email": "crime1@bad.org", "account": "FRBAD0RG78901234567890-0000001" }  
,  
{ "_id": "646cee044f07a537bf388bfa", "pseudo": "crime2", "password": "$2a$10$HY7nEbdSoWSJngy5Nv2DB.N4nq  
BUSdUb5dURwsA/rF4Cpb1BZ0ge0", "email": "crime2@bad.org", "account": "FRBAD0RG78901234567890-0000002" }  
,  
]  
  
const auctionrequests=[  
{ "_id": "646cee044f07a537bf388bfd", "buyer": "646cee044f07a537bf388bf7", "price": 50000, "lethality": 10  
, "kidneysVirulence": 1, "survival": { "air": 100 } },
```

```

{"_id":"646cee044f07a537bf388bff","buyer":"646cee044f07a537bf388bfa","price":10000,"lethality":3,
"lungsVirulence":1,"brainVirulence":2,"heartVirulence":3,"temperature":{"min":-10,"max":40},"survival":{"ground":10,"water":1000}},
]
const auctiontransactions=[
{"_id":"646cee044f07a537bf388c02","buyer":{"ref":"646cee044f07a537bf388bf7","pseudo":"crime1","email":"crime1@bad.org","account":"FRBAD0RG78901234567890-0000001"},"seller":{"ref":"646cee044f07a537bf388bf4","pseudo":"drmad","email":"drmad@bad.org","account":"FRDRMAD578901234567890-0000666"},"name":"madovirus","date":{"$date":"2023-05-23T16:47:00.817Z"},"price":75000,"lethality":40,"brainVirulence":9,"lungsVirulence":7,"heartVirulence":2,"kidneysVirulence":2,"temperature":{"min":-30,"max":70},"survival":{"air":1000,"water":200,"ground":30}},
]

module.exports = {
  items,
  shopusers,
  orders,
  bankaccounts,
  transactions,
  auctionusers,
  auctionrequests,
  auctiontransactions,
}

```

## Exercices

Pour chaque endpoint de routage, il doit y avoir une fonction correspondante dans le contrôleur et le service. Par exemple, pour **GET /items**, il doit y avoir une fonction **getItems** dans le contrôleur et une autre dans le service.

Pour tout ce qui concerne les requêtes POST et PUT, envisagez l'utilisation d'un middleware de validation pour assurer l'intégrité des données entrantes. Cela inclut la vérification des formats, des valeurs obligatoires et de la cohérence des données.

### Exercice 1 : Routeur Shop

- GET /items : Récupérer tous les items.
- GET /items/:id : Récupérer un item par son ID.
- GET /items/:id/promotion : Récupérer la promotion d'un item spécifique par ID.
- GET /shopusers : Récupérer tous les utilisateurs du shop.
- POST /items : Ajouter un nouvel item.
- PUT /items/:id : Mettre à jour un item par son ID.
- DELETE /items/:id : Supprimer un item par son ID.

## Exercice 2 : Routeur BankAccount

- GET /bankaccounts : Récupérer tous les comptes bancaires avec possibilité de trier par montant (ascendant ou descendant).
- PATCH /bankaccounts/:id : Ajouter ou soustraire un montant du compte bancaire spécifié.
- POST /bankaccounts : Ajouter un nouveau compte bancaire.

## Exercice 3 : Routeur Auction

- GET /auctionusers?email=[email] : Filtrer les utilisateurs d'enchères par email.
- GET /auctionrequests : Récupérer toutes les demandes d'enchères triées par légalité (ascendant ou descendant).
- PUT /auctionusers : Mettre à jour le "pseudo" d'un utilisateur d'enchères en fonction de son email.