





# Documentation de l'API - Gestion des Factures

Base URL : <http://localhost:3000/api/invoices>

## Introduction

Cette API permet de gérer les factures (**CRUD**) :

-  **Créer une facture**
-  **Récupérer toutes les factures**
-  **Mettre à jour une facture**
-  **Supprimer une facture**

## 1. Récupérer toutes les factures

- ♦ **Endpoint** : GET /api/invoices
- ♦ **Description** : Récupère la liste de toutes les factures enregistrées.

### Exemple de requête



GET <http://localhost:3000/api/invoices>

### Réponse attendue

```
[
  {
    "id": "65c2f1a7b9a4f5123cdd50d3",
    "date": "2025-01-01",
    "customer": {
      "firstName": "Alice",
      "lastName": "Dupont",
      "phoneNumber": "0654321987",
      "billingAddress": {
```

```
{
  "address": "12 rue des Lilas",
  "zipCode": "75001",
  "city": "Paris",
  "country": "France"
},
{
  "products": [
    { "name": "Organic Apple", "quantity": 3, "price": 1.5 },
    { "name": "Whole Milk", "quantity": 2, "price": 0.99 }
  ],
  "totalAmount": 45.75,
  "status": "Paid"
}
]
```

## Codes de réponse

| Code | Description   |
|------|---|
| 200  |  Succès - Liste des factures retournée |
| 500  |  Erreur serveur                        |

## 2. Créer une facture

- ◆ **Endpoint** : POST /api/invoices
- ◆ **Description** : Crée une nouvelle facture dans la base de données.

### Exemple de requête

POST <http://localhost:3000/api/invoices>  
Content-Type: application/json

## ◆ Corps de la requête




```
{
  "customer": {
    "firstName": "Alice",
    "lastName": "Dupont",
    "phoneNumber": "0654321987",
    "billingAddress": {
      "address": "12 rue des Lilas",
      "zipCode": "75001",
      "city": "Paris",
      "country": "France"
    }
  },
  "products": [
    { "name": "Organic Apple", "quantity": 3, "price": 1.5 },
    { "name": "Whole Milk", "quantity": 2, "price": 0.99 }
  ],
  "totalAmount": 45.75,
  "status": "Paid"
}
```

## Réponse attendue

```
{
  "id": "65c2f1a7b9a4f5123cdd50d3",
  "customer": {
    "firstName": "Alice",
    "lastName": "Dupont",
    "phoneNumber": "0654321987",
    "billingAddress": {
      "address": "12 rue des Lilas",
      "zipCode": "75001",
      "city": "Paris",
      "country": "France"
    }
  },
  "products": [
    { "name": "Organic Apple", "quantity": 3, "price": 1.5 },
    { "name": "Whole Milk", "quantity": 2, "price": 0.99 }
  ]
}
```

```
],  
  "totalAmount": 45.75,  
  "status": "Paid",  
  "createdAt": "2025-02-10T12:34:56.789Z"  
}
```

### Codes de réponse

| Code | Description  |
|------|--|
| 201  |  Facture créée avec succès                            |
| 400  |  Requête invalide (données manquantes ou incorrectes) |
| 500  |  Erreur serveur                                       |

## 3. Mettre à jour une facture

- ◆ **Endpoint** : PUT /api/invoices/:id
- ◆ **Description** : Met à jour une facture existante.

### Exemple de requête

PUT <http://localhost:3000/api/invoices/65c2f1a7b9a4f5123cdd50d3>  
Content-Type: application/json




### ◆ Corps de la requête

```
{  
  "status": "Pending"  
}
```

### Réponse attendue

```
{
  "id": "65c2f1a7b9a4f5123cdd50d3",
  "status": "Pending",
  "message": "Facture mise à jour avec succès"
}
```

### Codes de réponse

| Code | Description   |
|------|---|
| 200  |  Facture mise à jour avec succès |
| 404  |  Facture non trouvée             |
| 500  |  Erreur serveur                  |

## 4. Supprimer une facture

- ◆ **Endpoint** : DELETE /api/invoices/:id
- ◆ **Description** : Supprime une facture existante.

### Exemple de requête

DELETE <http://localhost:3000/api/invoices/65c2f1a7b9a4f5123cdd50d3>

### Réponse attendue

```
{
  "message": "Facture supprimée avec succès"
}
```

## ⚠ Codes de réponse

| Code | Description                     |
|------|---------------------------------|
| 200  | ✅ Facture supprimée avec succès |
| 404  | ❌ Facture non trouvée           |
| 500  | ❌ Erreur serveur                |

## 🔗 Exemple de modèle de données (Invoice.js)

Si tu veux voir à quoi ressemble le modèle Mongoose de Invoice.js, voici un exemple :





```
const mongoose = require('mongoose');

const invoiceSchema = new mongoose.Schema({
  customer: {
    firstName: String,
    lastName: String,
    phoneNumber: String,
    billingAddress: {
      address: String,
      zipCode: String,
      city: String,
      country: String
    }
  },
  products: [
    {
      name: String,
      quantity: Number,
      price: Number
    }
  ],
  totalAmount: Number,
  status: { type: String, default: "Pending" },
  createdAt: { type: Date, default: Date.now }
});
```

```
module.exports = mongoose.model('Invoice', invoiceSchema);
```

## Conclusion

Cette API permet la gestion complète des factures :

-  **Lister** toutes les factures (GET /api/invoices)
-  **Créer** une facture (POST /api/invoices)
-  **Mettre à jour** une facture (PUT /api/invoices/:id)
-  **Supprimer** une facture (DELETE /api/invoices/:id)