

# Documentation API - Outil d'Analyse des Paniers d'Achat

## Table des matières

1.	Endpoint : GET /analytics/total_sales .....	1
2.	Endpoint : GET /analytics/trending_products .....	3
2.	Endpoint : GET /analytics/category_sales.....	6
1.	Endpoint : GET /products .....	8

Cette API permet aux administrateurs d'un site e-commerce d'accéder à des statistiques et des données analytiques concernant les ventes et les produits. Chaque endpoint est conçu pour répondre à un besoin spécifique en fournissant des informations claires et exploitables.

## 1. Endpoint : GET /analytics/total\_sales

Cet endpoint calcule et retourne le montant total des ventes sur une période donnée (7 jours, 30 jours ou 12 mois).

- Paramètres

Nom	Type	Requis	Description
period	String	Oui	Définit la période pour laquelle les ventes doivent être calculées. Les valeurs possibles sont : - 7d : 7 derniers jours - 30d : 30 derniers jours - 12m : 12 derniers mois

- Exemple de requête

```
GET http://localhost:5000/analytics/total_sales?period=12m
```

- Exemple de réponse réussie (200)

Si des ventes sont enregistrées dans la période demandée :

```
{  
  
  "totalSales": 258939842.94,  
  
  "period": "12m"
```

```
}
```

Si aucune vente n'est enregistrée pour la période sélectionnée :

```
{  
  "totalSales": 0,  
  "period": "12m"  
}
```

- **Cas d'erreur**

#### **Période invalide (400 - Bad Request)**

Si le paramètre period n'est pas l'une des valeurs autorisées (7d, 30d, 12m) :

```
{  
  "error": "Invalid period selected. Use 7d, 30d, or 12m."  
}
```

#### **Erreur serveur (500 - Internal Server Error)**

Si un problème interne survient lors de l'exécution de l'agrégation des données :

```
{  
  "error": "Internal Server Error. Please try again later."  
}
```

- **Détails Techniques**

#### **Traitement des paramètres :**

La période est transformée en une plage de dates (startDate et currentDate) selon les règles suivantes :

- 7d : Calcul à partir des 7 derniers jours.
- 30d : Calcul à partir des 30 derniers jours.
- 12m : Calcul à partir des 12 derniers mois.

Si la période n'est pas valide, une erreur est levée (Invalid period selected).

#### **Pipeline MongoDB utilisé :**

- **Filtrage** : Seules les ventes dont la date (Date) est comprise entre startDate et currentDate sont sélectionnées.
- **Jointure** : Les détails des produits (ProductID, Price) sont récupérés via la collection Products.

- **Calcul** : Le total des ventes est obtenu en multipliant la quantité vendue (Quantity) par le prix unitaire (Price) de chaque produit.

**Performance :**

- L'indexation de la colonne Date dans la collection Sale améliore les performances des requêtes.
- L'utilisation de \$lookup pour joindre les données est optimisée si la collection Products est correctement indexée sur le champ ProductID.

- **Diagramme de traitement**

**Requête reçue** : /analytics/total\_sales?period=30d

**Validation** : Vérification de la validité de la période (7d, 30d, ou 12m).

**Construction de la plage de dates :**

- currentDate = 2024-11-27
- startDate = 2024-10-28 (pour une période de 30 jours).

**Agrégation MongoDB :**

- Filtrage des ventes dans la période.
- Jointure des détails des produits.
- Calcul du montant total des ventes.

**Réponse** : Renvoi des données calculées ou d'une erreur.

## 2. Endpoint : GET /analytics/trending\_products

Ce endpoint retourne les 5 produits les plus vendus, triés par quantité totale vendue. Pour chaque produit, il affiche son nom, sa quantité vendue, son prix unitaire et le montant total des ventes.

- **Exemple de Requête**

**GET** [http://localhost:5000/analytics/trending\\_products](http://localhost:5000/analytics/trending_products)

- **Exemple de Réponse Réussie (200)**

```
[
  {
    "_id": 53,
    "totalQuantity": 16902,
    "ProductName": "Keyboard",
```

```
"Price": 36.12,
"totalSales": 610500.24
},
{
  "_id": 23,
  "totalQuantity": 16804,
  "ProductName": "History Book",
  "Price": 491.14,
  "totalSales": 8253116.56
},
{
  "_id": 96,
  "totalQuantity": 16659,
  "ProductName": "Water Bottle",
  "Price": 203.14,
  "totalSales": 3384109.26
},
{
  "_id": 34,
  "totalQuantity": 16617,
  "ProductName": "Dress",
  "Price": 170.79,
  "totalSales": 2838017.4299999997
},
{
  "_id": 66,
  "totalQuantity": 16617,
  "ProductName": "Salad Spinner",
  "Price": 336.89,
```

```
"totalSales": 5598101.13
}
]
```

- **Cas d'Erreur**

**Erreur interne du serveur (500 - Internal Server Error)**

Si une erreur survient lors de l'exécution de la requête :

```
{
  "error": "Impossible de récupérer les produits tendance"
}
```

- **Détails Techniques**

**Pipeline MongoDB utilisé :**

- **\$group** : Regroupe les ventes par ProductID et calcule la quantité totale vendue pour chaque produit.
- **\$lookup** : Effectue une jointure avec la collection products pour récupérer les détails des produits (nom, prix unitaire, etc.).
- **\$unwind** : Transforme les données jointes en un document unique par produit.
- **\$project** : Sélectionne les champs nécessaires (ProductName, Price, totalQuantity, totalSales) et calcule le total des ventes en multipliant la quantité vendue par le prix unitaire.
- **\$sort** : Trie les produits par quantité totale vendue en ordre décroissant.
- **\$limit** : Limite les résultats aux 5 produits les plus vendus.

**Performance :**

- L'indexation des champs ProductID dans les collections Sale et products améliore l'efficacité des jointures et des tris.
- La limitation (\$limit) permet de réduire le volume des données retournées.

- **Diagramme de Traitement**

**Requête reçue :** /analytics/trending\_products.

**Pipeline MongoDB exécuté :**

- Regroupement des données par produit.
- Calcul des quantités vendues et des montants totaux.

- Jointure avec les détails des produits.
- Tri des résultats par quantité vendue.
- Limitation aux 5 premiers produits.

**Résultat généré :** Retourne une liste des produits les plus vendus avec leurs détails.

## 2. Endpoint : GET /analytics/category\_sales

Ce **endpoint** retourne la répartition des ventes par catégorie. Pour chaque catégorie, il affiche :

- Le nom de la catégorie.
- Le nombre total de ventes pour cette catégorie.
- Le pourcentage que représente cette catégorie par rapport au total des ventes.

- **Exemple de Requête**

GET [http://localhost:5000/analytics/category\\_sales](http://localhost:5000/analytics/category_sales)

- **Exemple de Réponse Réussie (200)**

```
{
  "message": "Répartition des ventes par catégorie récupérée avec succès.",
  "data": [
    {
      "category": "Clothing",
      "totalSales": 317992,
      "percentage": "19.91"
    },
    {
      "category": "Sports & Outdoors",
      "totalSales": 320690,
      "percentage": "20.08"
    },
    {
      "category": "Home & Kitchen",
      "totalSales": 319174,
      "percentage": "19.99"
    }
  ]
}
```

```

    },
    {
      "category": "Books",
      "totalSales": 318935,
      "percentage": "19.97"
    },
    {
      "category": "Electronics",
      "totalSales": 320157,
      "percentage": "20.05"
    }
  ]
}

```

- **Cas d'Erreur**

#### Erreur interne du serveur (500 - Internal Server Error)

Si une erreur survient pendant l'exécution :

```

{
  "error": "Impossible de récupérer les ventes par catégorie"
}

```

- **Détails Techniques**

#### Pipeline MongoDB utilisé :

- **\$lookup** : Jointure entre la collection sales et la collection products pour obtenir les informations sur la catégorie de chaque produit vendu.
- **\$unwind** : Transforme les données de jointure (qui sont en tableau) en documents individuels pour chaque produit.
- **\$group** : Regroupe les données par catégorie et calcule le total des quantités vendues pour chaque catégorie.

#### Calcul des pourcentages :

- Le **total global des ventes** est calculé en additionnant les ventes de toutes les catégories.

- Le **pourcentage des ventes par catégorie** est ensuite dérivé par la formule suivante :  

$$\text{Pourcentage} = \left( \frac{\text{Ventes de la catégorie}}{\text{Ventes totales}} \times 100 \right)$$
- Ce pourcentage est formaté avec deux décimales pour un affichage clair.

#### Performance :

- Assurez-vous que les champs ProductID dans les collections sales et products sont indexés pour accélérer la jointure.
- Si le volume des données est élevé, envisagez d'utiliser des solutions de cache pour optimiser les requêtes fréquentes.

- **Diagramme de Traitement**

**Requête reçue :** /analytics/category\_sales.

**Pipeline MongoDB exécuté :**

- Jointure entre sales et products pour récupérer les catégories.
- Agrégation des quantités vendues par catégorie.

**Calcul des pourcentages :** Basé sur le total des ventes de toutes les catégories.

**Résultat retourné :** Une liste des catégories avec les quantités vendues et leur part en pourcentage.

## 1. Endpoint : GET /products

Ce **endpoint** retourne un tableau de produits avec des informations détaillées, y compris :

- Le nom du produit.
- Sa catégorie.
- Son prix unitaire.
- Le nombre total de ventes pour chaque produit.

- **Exemple de Requête**

**GET** <http://localhost:5000/products>

- **Exemple de Réponse Réussie (200)**

```
{
  "_id": 2,
  "totalSales": 15926,
  "ProductName": "Laptop",
```



```
"Category": "Electronics",
  "Price": 468.75
},
{
  "_id": 70,
  "totalSales": 16243,
  "ProductName": "Egg Cooker",
  "Category": "Home & Kitchen",
  "Price": 436.68
},
{
  "_id": 37,
  "totalSales": 16193,
  "ProductName": "Hat",
  "Category": "Clothing",
  "Price": 31.69
},
{
  "_id": 36,
  "totalSales": 15770,
  "ProductName": "Sandals",
  "Category": "Clothing",
  "Price": 410.14
},
```

- **Cas d'Erreur**

**Erreur interne du serveur (500 - Internal Server Error)**

Si une erreur survient lors de la récupération des données :

```
{
  "error": "Impossible de récupérer la liste des produits"
}
```

- **Détails Techniques**

**Pipeline MongoDB utilisé :**

- **\$group** : Regroupe les documents de la collection sales par ID de produit (ProductID) pour calculer le total des ventes pour chaque produit.
- **\$lookup** : Jointure avec la collection products pour récupérer les détails des produits liés à chaque ProductID.
- **\$unwind** : Décompresse les détails des produits récupérés sous forme de tableau pour les transformer en documents individuels.
- **\$project** : Sélectionne uniquement les champs pertinents à retourner (nom, catégorie, prix, total des ventes).

**Champs retournés dans la réponse :**

- **ProductName** : Nom du produit.
- **Category** : Catégorie du produit.
- **Price** : Prix unitaire.
- **totalSales** : Quantité totale vendue.

- **Diagramme de Traitement**

**Requête reçue :** /products.

**Étape 1 :**

Exécution du pipeline MongoDB pour agréger les données des ventes et relier les détails des produits.

**Étape 2 :**

Extraction des informations nécessaires et calcul des ventes par produit.

**Étape 3 :**

Résultat structuré envoyé au client.