

Exercice : Pagination d'articles en PHP

Objectif pédagogique

Apprendre à paginer des résultats issus d'une base de données en PHP :

- Partie 1 : créer une page HTML qui affiche les articles paginés.
 - Partie 2 : créer une API REST qui renvoie les articles en JSON avec pagination.
-

Base de données utilisée

Nous travaillerons sur la table **article** de la base de donnée bière, déjà utilisée lors de précédents exercices SQL.

Partie 1 : Page HTML avec pagination

Consignes Page

1. Écrire un script PHP **articles.php** qui :

- Se connecte à la base de données.
- Récupère un nombre défini d'articles par page (ex : 10).
- Affiche les articles dans un tableau HTML.
- Affiche les liens de pagination (page précédente, suivante, numéros de page...).

2. Ajouter les fonctionnalités suivantes :

- Indication de la page actuelle.
- Gestion des cas limites (page négative, trop grande, etc.).
- Protection contre les injections SQL (requêtes préparées obligatoires).

Indications

- Utiliser **LIMIT** et **OFFSET** en SQL.
 - Calcul de l'offset : $(page - 1) * articles_par_page$.
 - Exemple de lien : **articles.php?page=2**.
-

Partie 2 : API paginée (format JSON)

Consignes API

Créer un fichier **api/articles.php** qui :

- Renvoie un JSON des articles paginés.
- Accepte les paramètres **page** et **limit** via **GET**.
- Retourne également des méta-informations dans le JSON (ex. : **total**, **current_page**, **last_page**).

Exemple de sortie attendue

```
{
  "current_page": 2,
  "per_page": 10,
  "total": 58,
  "last_page": 6,
  "data": [
    {
      "ID_ARTICLE": 11,
      "NOM_ARTICLE": "Vin rouge doux",
      "PRIX_ACHAT": 12.5
    }
  ]
}
```

Bonus (optionnel)

- Permettre de filtrer les articles par `ID_TYPE` ou `ID_MARQUE` (via des paramètres GET supplémentaires).
- Retourner un code HTTP 400 en cas de paramètres invalides (`page` négative, etc.).

Conseils

- Faites d'abord fonctionner la logique de pagination côté SQL avant d'intégrer à PHP.
- Testez vos URLs manuellement dans le navigateur (`?page=1&limit=20`).
- Utilisez `json_encode()` avec l'option `JSON_PRETTY_PRINT` pour tester.

Test de votre API

Utilisez [Postman](#) ou le navigateur pour tester les endpoints :

```
GET http://localhost/api/articles.php?page=1&limit=10
```

Structure suggérée

```
/project-root
├── articles.php
└── api/
    └── articles.php
```