SAÉ 3.01

Création et exploitation d'une base de données

GROMAT Jonathan

HAMZAOUI Karim

BOUAIED Abdallah

Livrable 3 – Collecte et analyse de

Compréhension des besoins en données

Analyse des Exigences du Projet

Dans le cadre de notre projet, une analyse minutieuse des exigences a mis en lumière les besoins spécifiques en termes de données issues de l'INPN. Ces données sont cruciales pour alimenter les différentes fonctionnalités de l'application, notamment :

Informations sur les Espèces : Les données essentielles incluent les noms scientifiques, les noms vernaculaires, les habitats, la répartition géographique, et les classifications taxonomiques. Ces informations serviront de base à la création et à l'enrichissement des naturothèques et des fiches d'observations.

Données de Localisation : Des informations précises sur les lieux d'observation des espèces sont nécessaires. Elles permettront aux utilisateurs d'enregistrer et de partager des observations géolocalisées dans le cadre de leur contribution à la base de données naturalistes.

Images et Médias Associés : La collecte de photographies et d'autres médias liés aux espèces est indispensable. Ces éléments visuels joueront un rôle clé dans l'enrichissement visuel des naturothèques et des observations, offrant ainsi une expérience utilisateur plus immersive et informative.

L'acquisition et l'intégration de ces données dans notre application permettront non seulement d'offrir une riche base de données sur la biodiversité, mais aussi de faciliter l'interaction des utilisateurs avec des informations naturalistes précises et diversifiées.

Identification des Types de Données Nécessaires

Pour l'analyse des besoins en données issues de l'INPN, nous avons dû nous concentrer sur les informations spécifiques fournies par leur API. Compte tenu de l'exemple fourni, voici les points clés que nous avons identifiés pour notre projet :

Informations Taxonomiques : Les détails taxonomiques comme le nom scientifique, le genre, la famille, l'ordre, la classe, le phylum et le règne sont cruciaux pour classer et comprendre chaque espèce.
Noms Vernaculaires : Les noms vernaculaires en français et en anglais, lorsqu'ils sont disponibles, sont utiles pour rendre les informations sur les espèces accessibles à un public plus large.
Localisation Géographique : Les codes régionaux (comme 'FR', 'gf', etc.) indiquent la distribution géographique des espèces, ce qui est vital pour les observations naturalistes.
Habitat et Groupes Vernaculaires : Les données sur l'habitat et les groupes vernaculaires (comme 'Arachnides') aident à comprendre l'environnement naturel et la catégorisation des espèces.
Liens vers des Ressources Complémentaires : Les liens vers des médias, des noms vernaculaires supplémentaires, et des pages web de l'INPN offrent des ressources additionnelles pour une compréhension approfondie des espèces.

Exploration des données de l'INPN

Familiarisation avec les Données de l'INPN

L'exploration des données de l'INPN a commencé par une phase de familiarisation intensive. Nous avons consacré plusieurs heures à examiner en détail les différentes sources d'informations fournies par l'INPN.

Nous avons exploré les différentes catégories de données, y compris les informations taxonomiques, les noms vernaculaires, les données de localisation, les informations sur l'habitat et les groupes vernaculaires, ainsi que les liens vers des ressources complémentaires. Cette exploration nous a permis de comprendre la richesse et la diversité des données disponibles, et de déterminer comment ces données peuvent être utilisées pour enrichir notre application.

Au cours de cette phase, nous avons également pris le temps de comprendre les méthodes de collecte de ces données et les protocoles utilisés pour garantir leur précision et leur fiabilité. Cela nous a permis de comprendre les défis potentiels que nous pourrions rencontrer lors de l'extraction des données, et de planifier en conséquence.

Il est important de noter que les données de l'INPN sont structurées au format JSON, ce qui a des implications significatives sur la manière dont nous interagissons avec ces données. Le format JSON est largement utilisé pour le stockage et l'échange de données, et offre une grande flexibilité en termes de types de données qu'il peut représenter.

Étude de la Structure des Données

Après nous être familiarisés avec les données, nous avons entrepris une étude approfondie de la structure des données de l'INPN.

Nous avons examiné les différents champs de données disponibles et les relations entre ces champs. Nous avons également cherché à comprendre comment ces données sont organisées et comment elles peuvent être extraites de manière efficace et précise.

Cette étude de la structure des données a nécessité une compréhension approfondie de la manière dont les données sont organisées et structurées. Nous avons examiné les différents champs de données disponibles, ainsi que les relations entre ces champs.

Nous avons également cherché à comprendre comment ces données sont liées les unes aux autres, et comment elles peuvent être utilisées pour répondre aux besoins spécifiques de notre application.

Cette phase a été essentielle pour comprendre comment naviguer efficacement dans les données de l'INPN. En comprenant la structure des données, nous avons pu identifier les informations clés nécessaires pour notre application et déterminer comment les extraire de manière efficace.

En outre, cette phase nous a permis de prévoir les défis potentiels que nous pourrions rencontrer lors de l'extraction des données. Par exemple, nous avons identifié des problèmes potentiels tels que les données manquantes ou inexactes, et nous avons élaboré des stratégies pour les surmonter.

En somme, cette phase d'exploration des données a été une étape cruciale de notre projet. Elle nous a permis de comprendre les données disponibles, de planifier comment nous allons les extraire et de préparer notre application à intégrer ces données de manière efficace et précise. Cette phase a jeté les bases de toutes les étapes suivantes de notre projet, et a été essentielle pour garantir le succès de notre application.

Identification des Défis Potentiels

Au cours de notre exploration des données, nous avons également identifié plusieurs défis potentiels que nous pourrions rencontrer lors de l'extraction des données.
Ces défis comprennent la gestion des données manquantes ou inexactes, la manipulation de grandes quantités de données, et la nécessité de transformer les données dans un format approprié pour notre application.
En identifiant ces défis à l'avance, nous sommes mieux préparés à les surmonter lors de l'extraction et de l'utilisation des données.

Extraction de données

Identification des Méthodes ou Outils pour l'Extraction des Données de l'INPN

API de l'INPN :
L'API de l'INPN est le point de départ essentiel de notre processus d'extraction de données. Elle constitue la passerelle qui nous permet d'accéder à une mine précieuse d'informations sur la biodiversité. Chaque espèce a son propre identifiant unique, et nous utilisons ces identifiants pour effectuer des requêtes spécifiques. Par exemple, une requête typique pourrait ressembler à ceci :
https://taxref.mnhn.fr/api/taxa/{id_espece}
Cette requête retourne une réponse sous forme de données JSON, qui comprend des détails tels que les noms scientifiques, les noms vernaculaires, les informations taxonomiques, les habitats, les régions de distribution, et bien d'autres informations cruciales.

Scripting en PHP :
Pour interagir avec l'API de l'INPN de manière programmatique, nous utilisons des scripts PHP. Ces scripts utilisent la bibliothèque cURL (Client URL) pour établir une connexion avec l'API, envoyer des requêtes HTTP, et récupérer les données. Ils sont essentiels pour automatiser le processus d'extraction. Voici un exemple de script PHP qui effectue une requête à l'API et récupère les données JSON :

```php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://taxref.mnhn.fr/api/taxa/{id_espece}");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$apiResponse = curl_exec($ch);
curl_close($ch);
// Traitement de la réponse JSON
```

Traitement JSON en PHP :
Une fois que nous avons obtenu la réponse JSON de l'API, nous utilisons des fonctions PHP, notamment json_decode, pour analyser cette réponse et la convertir en un tableau associatif PHP. Cette étape est cruciale car elle nous permet de naviguer facilement dans les données et de les extraire pour une utilisation ultérieure. Voici comment nous procédons au traitement des données JSON en PHP :

```php
$data = json_decode($apiResponse, true);
foreach ($data as $item) {
    // Extraction et traitement des informations spécifiques
}
```

Interaction Approfondie avec MySQL via PDO dans le Processus d'Extraction des Données

1. **Connexion Sécurisée à MySQL avec PDO :**

Pour établir une connexion sécurisée avec la base de données MySQL, nous utilisons PDO, qui offre une interface de programmation robuste pour les interactions avec la base de données. Voici un exemple de connexion sécurisée à la base de données :

```php
$pdo = new PDO("mysql:host=host;dbname=dbname;charset=utf8", "username", "password", [
    PDO::ATTR_ERRMODE ⇒ PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE ⇒ PDO::FETCH_ASSOC
]);
```

2. **Requêtes Préparées pour la Sécurité :**

Pour éviter les injections SQL et garantir la sécurité des données, nous utilisons des requêtes préparées avec PDO. Cela implique la préparation d'une requête SQL avec des paramètres liés, qui sont ensuite exécutés avec des valeurs spécifiques. Par exemple :

```php
$stmt = $pdo→prepare("INSERT INTO especes (nom_scientifique, habitat) VALUES
(:nom_scientifique, :habitat)");
$stmt→bindParam(':nom_scientifique', $nomScientifique);
$stmt→bindParam(':habitat', $habitat);
$stmt→execute();
```

(-__-)

3. **Insertions et Mises à Jour :**

Les scripts PHP manipulent les données pour les insérer ou les mettre à jour dans la base de données. Nous utilisons INSERT INTO pour ajouter de nouvelles données et UPDATE pour modifier les données existantes. Par exemple, une mise à jour pourrait ressembler à :

```
$stmt = $pdo->prepare("UPDATE especes SET habitat = :habitat WHERE nom_scientifique
= :nom_scientifique");
$stmt->bindParam(':habitat', $nouvelHabitat);
$stmt->bindParam(':nom_scientifique', $nomScientifique);
$stmt->execute();
```

4. **Gestion des Transactions :**

Pour garantir la cohérence des données, nous utilisons des transactions PDO. Cela permet de regrouper plusieurs opérations en une seule unité de travail. En cas d'erreur, toutes les modifications peuvent être annulées pour préserver l'intégrité des données. Par exemple :

```
$pdo->beginTransaction();
// Effectuer plusieurs insertions/mises à jour
if (toutesLesOperationsReussies) {
    $pdo->commit();
} else {
    $pdo->rollBack();
}
```

5. **Extraction et Analyse des Données :**

Une fois les données extraites et stockées dans la base de données, des requêtes supplémentaires sont utilisées pour les analyser et les préparer pour l'affichage dans l'application. Par exemple, pour récupérer des données spécifiques :

```
$stmt = $pdo->query("SELECT * FROM especes WHERE habitat = 'Forêt'");
while ($row = $stmt->fetch()) {
    // Traiter chaque ligne de résultat
}
```

Nettoyage et préparation des données

Code d'Analyse des Données Extraites de l'INPN

Pour analyser les données extraites de l'INPN, nous avons mis en place le script PHP suivant :

```php
<?php

// Fonction pour exécuter une requête cURL
function executeCurl($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    return json_decode($response, true);
}

$nombreEspeces = 20;  // Nombre d'espèces à extraire
$especesExtraite = [];  // Tableau pour stocker les informations extraites
$doublons = [];  // Tableau pour stocker les doublons détectés

for ($i = 1; $i ⩽ $nombreEspeces; $i++) {
    $urlEspece = "https://taxref.mnhn.fr/api/taxa/$i";
    $urlMedia = "https://taxref.mnhn.fr/api/taxa/$i/media";

    // Extraction des données de l'espèce et des médias associés
    $donneesEspece = executeCurl($urlEspece);
    $donneesMedia = executeCurl($urlMedia);

    // Vérification des erreurs dans la réponse
    if (!$donneesEspece || !$donneesMedia) {
        echo "Pas de données reçu pour l'id $i<br>";
        continue;
    }

    // Ajout des données extraites au tableau
    $especesExtraite[] = [
        'id' ⇒ $i,
        'nomScientifique' ⇒ $donneesEspece['scientificName'],
        'habitat' ⇒ $donneesEspece['habitat'],
        'media' ⇒ $donneesMedia['_embedded']['media']
    ];
}

// Affichage des résultats
echo "Nombre d'espèces extraites: " . count($especesExtraite) . "<br>";
if (!empty($doublons)) {
    echo "Doublons détectés: " . implode(', ', $doublons) . "<br>";
}

echo "Espèces extraites :<br>";
echo "<pre>";
print_r($especesExtraite);
echo "</pre>";
?>
```

Le script parcourt une série d'ID d'espèces (de 1 à 20) et effectue des requêtes API pour chaque ID. Voici les étapes clés du script :

- Exécution des Requêtes cURL : Pour chaque ID d'espèce, le script effectue une requête à l'API de l'INPN pour récupérer les données de l'espèce et des médias associés.
- Gestion des Réponses : Chaque réponse est analysée. Si des données sont manquantes pour un ID donné, le script enregistre cette absence et continue avec le prochain ID.
- Stockage des Données : Les données extraites sont stockées dans un tableau pour une analyse ultérieure.

D'après les résultats de votre script et l'analyse des données reçues de l'API de l'INPN, voici notre évaluation :

Analyse des Données Extraites

1. **Absence de Données pour Plusieurs ID d'Espèces :**

- Nous avons constaté que pour de nombreux ID (1 à 20), aucune donnée n'a été reçue. Cela indique soit que ces ID n'existent pas dans la base de données de l'INPN, soit que les données pour ces espèces spécifiques ne sont pas disponibles.

```
Pas de données reçu pour l'id 1
Pas de données reçu pour l'id 2
Pas de données reçu pour l'id 3
Pas de données reçu pour l'id 4
Pas de données reçu pour l'id 5
Pas de données reçu pour l'id 6
Pas de données reçu pour l'id 7
Pas de données reçu pour l'id 8
Pas de données reçu pour l'id 9
Pas de données reçu pour l'id 10
Pas de données reçu pour l'id 11
Pas de données reçu pour l'id 12
Pas de données reçu pour l'id 13
Données reçu pour l'id 14
Pas de données reçu pour l'id 15
Pas de données reçu pour l'id 16
Données reçu pour l'id 17
Données reçu pour l'id 18
Pas de données reçu pour l'id 19
Pas de données reçu pour l'id 20
Nombre d'espèces extraites: 3
Espèces extraites :
```

- Action : Nous devrions vérifier la validité de ces ID directement via l'API de l'INPN ou envisager d'utiliser une plage d'ID différente qui pourrait avoir des données disponibles.

2. **Données Manquantes ou Incomplètes :**

- Pour les espèces dont les données ont été extraites, il est possible que certaines informations soient manquantes ou incomplètes. Par exemple, des détails

spécifiques sur l'habitat ou des données média pourraient ne pas être disponibles pour toutes les espèces.


```
Array
(
    [0] => Array
        (
            [id] => 14
            [nomScientifique] => Hydromantes ambrosii
            [habitat] => 3
            [media] => Array
                (
                    [0] => Array
                        (
                            [id] => 206388
                            [taxon] => Array
                                (
                                    [id] => 79251
                                    [scientificName] => Speleomantes strinatii
                                    [fullNameHtml] => Speleomantes strinatii (Aellen, 1958)
                                    [referenceId] => 79251
                                    [parentId] => 197788
                                    [referenceNameHtml] => Speleomantes strinatii (Aellen, 1958)
                                )

                            [copyright] => S. Sant
                            [title] =>
                            [licence] => CC BY-NC-SA
                            [licenceUrl] => https://creativecommons.org/licenses/by-nc-sa/4.0/
                            [mimeType] => image/jpeg
                            [_links] => Array
                                (
                                    [self] => Array
                                        (
                                            [href] => https://taxref.mnhn.fr/api/media/206388
                                        )

                                    [taxon] => Array
                                        (
                                            [href] => https://taxref.mnhn.fr/api/taxa/79251
                                        )

                                    [file] => Array
                                        (
                                            [href] => https://taxref.mnhn.fr/api/media/download/inpn/206388
                                        )

                                    [thumbnailFile] => Array
                                        (
                                            [href] => https://taxref.mnhn.fr/api/media/download/thumbnail/206388
                                        )

                                )

                        )
```


- Action : Nous devrions mettre en place des mécanismes pour gérer ces lacunes, comme l'ajout de placeholders ou de données par défaut lorsqu'une information spécifique n'est pas disponible.

3. **Problèmes Potentiels de Doublons :**

- Bien que les données extraites n'indiquent pas explicitement des doublons, la possibilité de doublons existe, notamment si l'API renvoie les mêmes informations pour différents ID.
- Action : Nous pourrions implémenter une vérification de doublons dans notre script, en nous basant sur des combinaisons de champs uniques comme le nom scientifique et l'habitat.

Nettoyage et transformation des données en effectuant des opérations de filtrage, de correction ou de complétion et dans un format approprié pour les besoins de l'application.

Le script a été conçu pour identifier et traiter les données incomplètes, manquantes ou en double. En fournissant une structure claire pour l'affichage des données, il facilite la compréhension et l'analyse des informations récupérées de l'INPN.

Fonctionnalités Clés du Code

1. Exécution de Requêtes cURL : Le script utilise cURL pour interroger l'API de l'INPN et récupérer les données des espèces.

2. Gestion des Réponses : Le script analyse chaque réponse de l'API. S'il manque des données pour un ID spécifique, le script enregistre cette information et passe à l'ID suivant.
3. Affichage Structuré : Le script affiche les informations clés pour chaque espèce, y compris le nom scientifique, le nom vernaculaire, l'habitat, la répartition géographique, et les médias associés.

```php
<?php

// Fonction pour exécuter une requête cURL
function executeCurl($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    return json_decode($response, true);
}

$nombreEspeces = 20; // Nombre d'espèces à extraire

for ($i = 1; $i ⩽ $nombreEspeces; $i++) {
    $urlEspece = "https://taxref.mnhn.fr/api/taxa/$i";
    $donneesEspece = executeCurl($urlEspece);

    // Vérification et gestion des erreurs et des données manquantes
    if (!$donneesEspece || empty($donneesEspece['scientificName'])) {
        echo "Données incomplètes ou absentes pour l'ID $i<br>";
        continue;
    }

    // Affichage des informations de chaque espèce
    echo "ID: " . $i . "<br>";
    echo "Nom Scientifique: " . ($donneesEspece['scientificName'] ?? 'Non spécifié') . "<br>";
    echo "Nom Vernaculaire: " . ($donneesEspece['vernacularName'] ?? 'Non spécifié') . "<br>";
    echo "Habitat: " . ($donneesEspece['habitat'] ?? 'Non spécifié') . "<br>";
    echo "Répartition Géographique: " . ($donneesEspece['geographicDistribution'] ?? 'Non
spécifié') . "<br>";
    echo "Médias: ";
    if (!empty($donneesEspece['media'])) {
        foreach ($donneesEspece['media'] as $media) {
            echo $media['url'] . " ";
        }
    } else {
        echo "Aucun média disponible";
    }
    echo "<br><br>";
}

?>
```

Conclusion du Livrable sur la Collecte et Analyse des Données

Notre travail concernant ce livrable a impliqué plusieurs étapes cruciales pour garantir l'utilisation efficace des données de l'INPN dans notre application :

- Analyse des Besoins en Données : Nous avons identifié les types de données nécessaires, y compris les informations taxonomiques, les noms vernaculaires, les données de localisation, et les médias associés.
- Exploration des Données de l'INPN : Nous avons exploré et compris la structure et l'organisation des données de l'INPN, ce qui nous a permis de planifier efficacement leur extraction.
- Extraction et Analyse des Données : Nous avons mis en place des méthodes pour extraire les données de l'INPN, en utilisant des scripts PHP et l'API de l'INPN. Notre analyse a permis d'identifier les lacunes et les doublons potentiels dans les données.
- Nettoyage et Préparation des Données : Enfin, nous avons nettoyé les données et les avons préparées pour une utilisation optimale dans notre application, en accord avec les besoins identifiés.