```php
<?php

namespace App\Controllers;

use App\Models\Auteur;
use App\Models\Genre;
use App\Models\Livre;
use App\Models\Serie;
use App\Models\Utilisateur;
use Exception;

class WebServiceController extends BaseController {

    // ----------------------GET LIVRE---------------------------------

    public function getLivre() {
        // Récupère tous les livres de la base de données
        $livres = Livre::all();
        /*
        $json = json_encode($livres,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [$livres];
        // Retourne les livres au format JSON
        return $this->response->setJSON($data);
    }

    public function getLivreById($id){
        // Récupère le livre avec l'ID spécifié
        $livre = Livre::find($id);
        /*
        $json = json_encode($livre,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [$livre];
        // Retourne le livre au format JSON
        return $this->response->setJSON($data);
    }

    public function getLivreByGenreId($genreId){
        // Récupère le genre avec l'ID spécifié
        $genre = Genre::find($genreId);
        // Accède aux livres du genre
        $livres = $genre->livres;
        /*
        $json = json_encode($livres,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [$livres];
```

```
        return $this->response->setJSON($data);
    }

    public function getLivreByGenre($genre){

    }

    public function getLivreByAuteur($auteur){

    }

    public function getLivreByTitre($titre){
        // Récupère le livre avec le titre spécifié
        $livre = Livre::where('Nom_Livre', $titre)->first();
        /*
        $json = json_encode($livre,JSON_UNESCAPED_UNICODE);
        echo $json;*/
        $data = [$livre];
        return $this->response->setJSON($data);
    }

    public function getLivreBySerieId($serieId){
        $livres = Livre::where('Id_Serie', $serieId)->get();
        /*
        $json = json_encode($livres,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [$livres];
        return $this->response->setJSON($data);
    }

    public function getLivreBySerie($serie){

    }

    public function getLivreByISBN($isbn) {
        $livres = Livre::where('ISBN',$isbn)->get();
        $data = [$livres];
        return $this->response->setJSON($data);
    }

    // ---------------------- GET AUTEUR ----------------------------------

    public function getAuteur(){
        /**
     * Cette fonction récupère tous les auteurs de la base de données et les retourne au format JSON.
     */
        $auteurs = Auteur::all();
        /*
        $json = json_encode($auteurs,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
```

```
        $data = [// Méthode chargée de traiter les données de livres reçues sous forme de chaîne JSON
            'Auteurs' => $auteurs
        ];
        return $this->response->setJSON($data);
    }

    public function getAuteurById($id){

    }

    public function getAuteurByNom($nom){

    }

    public function getAuteurByPrenom($prenom){

    }

    public function getAuteurByLivre($livre){

    }


    // ---------------------- GET GENRE -------------------------------

    public function getGenre(){
        /**
         * Fonction pour récupérer tous les genres
         */
        $genres = Genre::all();
        /*
        $json = json_encode($genres,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [
            $genres
        ];
        return $this->response->setJSON($data);
    }

    public function getGenreById($id){

    }

    public function getGenreByName($name){

    }

    // ---------------------- GET SERIES -------------------------------

    public function getSerie() {
        /**
```

```
     * Récupère toutes les séries.
     */
        $series = Serie::all();
        /*
        $json = json_encode($series,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [
            'Serie' => $series
        ];
        return $this->response->setJSON($data);
    }

    public function getSerieById($id) {
        // Récupère une série par son ID.
        $serie = Serie::find($id);
        /*
        $json = json_encode($serie,JSON_UNESCAPED_UNICODE);
        echo $json;
        */
        $data = [
            'Serie' => $serie
        ];
        return $this->response->setJSON($data);
    }

    public function getSerieByName($name) {

    }

    public function getSerieByLivre($livre) {

    }

    // ---------------------- GET UTILISATEUR --------------------------------

    public function getUtilisateur() {

    }

    public function getProfile() {
        // Importe le helper JW
        helper('jwt');
        $user = getConnectedUtilisateur();
        unset($user->password);
        $data = $user;
        // Convertit l'objet utilisateur en JSON et le retourne
        return $this->response->setJSON($data);
    }
```

```php
// ---------------------- GET Inscritption ---------------------------------

public function register(){
    // **Récupération des données du formulaire**
    try {
        $nom = $this->request->getVar('nom');
        $prenom = $this->request->getVar('prenom');
        $pseudo = $this->request->getVar('pseudo');
        $email = $this->request->getVar('email');
        $password = $this->request->getVar('password');
        // **Hachage du mot de passe**
        $pass = password_hash($password, PASSWORD_DEFAULT);
        // **Vérification de l'existence du pseudo**
        $utilisateur = Utilisateur::where('Pseudonyme',$pseudo)->first();
        if ($utilisateur) {
            // **Pseudo déjà utilisé**
            return $this->getResponse(['message' => 'Pseudo déjà utiliser']);
        } else {
            // **Création d'un nouvel utilisateur**
            $user = new Utilisateur();
            $user->Nom = $nom;
            $user->Prenom = $prenom;
            $user->Pseudonyme = $pseudo;
            $user->password = $pass;
            $user->Email = $email;
            $user->Date_Inscription = date("Y-m-d");
            $user->Image = "img/user/user.png";
            $user->Id_Role = 1;
            $user->save();
            // **Vérification de l'enregistrement**
            $utilisateur = Utilisateur::where('Pseudonyme',$pseudo)->first();
            if ($utilisateur) {
                // **Inscription validée**
                return $this->getResponse(['message' => 'Inscription valide']);
            } else {
                // **Erreur lors de l'enregistrement**
                return $this->getResponse(['message' => 'Erreur lors de l\'inscription']);
            }
        }

    } catch (Exception $th) {
        // **Gestion des exceptions**
        return $this->getResponse(['message' => 'Erreur lors de l\'inscription']);
    }
}
}
```