http://127.0.0.1:8000/create-test-users

Utilisateurs de test créés avec succès !  
  
**Admin (avec 2FA) :** admin@pngdi.ga / password123  
**Agent (avec 2FA) :** agent@pngdi.ga / password123  
**Opérateur (sans 2FA) :** operator@pngdi.ga / password123

admin = \App\Models\User::firstOrCreate([

'email' => 'admin@pngdi.ga'

], [

'name' => 'Admin PNGDI',

'password' => bcrypt('admin123'),

'role' => 'admin',

'email\_verified\_at' => now(),

'is\_active' => true,

'last\_login\_at' => now()

]);

php artisan tinker

// Récupérer l'admin avec le bon password

$admin = \App\Models\User::where('email', 'admin@pngdi.ga')->first();

// Vérifier le password correct

echo "Test password 'password123': ";

if (Hash::check('password123', $admin->password)) {

echo "✅ Password correct\n";

} else {

echo "❌ Password incorrect\n";

}

// Test de la méthode hasRole()

echo "\nTest méthode hasRole():\n";

try {

$hasAdminRole = $admin->hasRole('admin');

echo "hasRole('admin'): " . ($hasAdminRole ? 'true' : 'false') . "\n";

$hasArrayRole = $admin->hasRole(['admin', 'agent']);

echo "hasRole(['admin', 'agent']): " . ($hasArrayRole ? 'true' : 'false') . "\n";

} catch (Exception $e) {

echo "❌ Erreur méthode hasRole(): " . $e->getMessage() . "\n";

}

// Test simulation connexion

echo "\nSimulation connexion Laravel:\n";

try {

Auth::attempt(['email' => 'admin@pngdi.ga', 'password' => 'password123']);

if (Auth::check()) {

$user = Auth::user();

echo "✅ Connexion réussie pour: " . $user->name . "\n";

echo "Role: " . $user->role . "\n";

Auth::logout();

} else {

echo "❌ Échec de connexion\n";

}

} catch (Exception $e) {

echo "❌ Erreur connexion: " . $e->getMessage() . "\n";

}

exit