**Sommaire**

1. **Introduction**
2. **Installation**
   1. Prérequis
   2. Installation locale
3. **Explication des fichiers principaux du projet**
4. **Modèles (Models)** 
   1. Description des modèles
   2. Relations entre les modèles
5. **Contrôleurs (Controller)**
   1. Description des contrôleurs
   2. Route associée
6. **Vues (View)**
   1. Structure des vues
   2. Utilisation de blade
7. **Middlewares**
   1. Liste des middlewares utilisés
   2. Fonctionnement et utilisation
8. **Evènements (Event)**
   1. Liste des évènement utilisés
   2. Fonctionnement
9. **Base de données**
   1. Schéma de la base de données
   2. Principales requêtes SQL
10. **Authentification**
11. **Tests**
12. **Introduction**
13. **Installation et configuration**
    1. Prérequis

Avant d’utiliser l’application Coherence Risk – CRM en locale, assurez-vous d'avoir les prérequis nécessaires en place tels que :

* **PHP**
* **Composer**
* **Node.js ou NPM**
* **Serveur WEB (WampServer ou Xamp)**
* **MySQL**
* **Git**
  1. Installation locale

Installer localement L’application Coherence Risk – CRM implique plusieurs étapes. Assurez-vous d'avoir Git, Composer et PHP installés sur votre machine avant de commencer. Voici les étapes générales :

* **Cloner le projet depuis Git**

Ouvrez votre invite de commande et taper :

git clone <https://github.com/Davidkouachi/coherence.git>

* **Accéder au répertoire du projet**

Toujours dans l’invite de commande taper :

Cd coherence

* **Installer les dépendances de l’application avec Composer**

Toujours dans l’invite de commande taper :

Composer Install

* **Création du fichier .env**
* Aller à la racine du projet
* Créer un fichier nommer .env sans extension
* Ouvrir le fichier .env.example.txt
* Copier les données du fichier (Ctrl + C)
* Ensuite coller les données dans le fichier .env que vous avez créer
* **Générer la clé d’application**

Taper dans l’invite de commande :

Php artisan key:generate

* **Configurer la base de données**
* Ouvrez votre serveur web
* Créer une base de données nommer coherence
* **Migration de la base de données**

Tapez dans l’invite de commande :

Php artisan migrate

* **Migration des seeders**

Taper dans l’invite de commande :

Php artisan migrate:refresh --seed

* **Lancer le serveur de développement**

Taper dans l’invite de commande :

Php artisan serve

Vous pouvez désormais avoir accès a l’application via l’adresse <http://127.0.0.1:8000> ou <http://localhost:8000>

1. **Explication des fichiers principaux du projet**

Les dossiers de l’application Coherence Risk - CRM suit une structure bien définie pour faciliter l'organisation du code source. Voici une explication des principaux dossiers et de leur objectif dans ladite application :

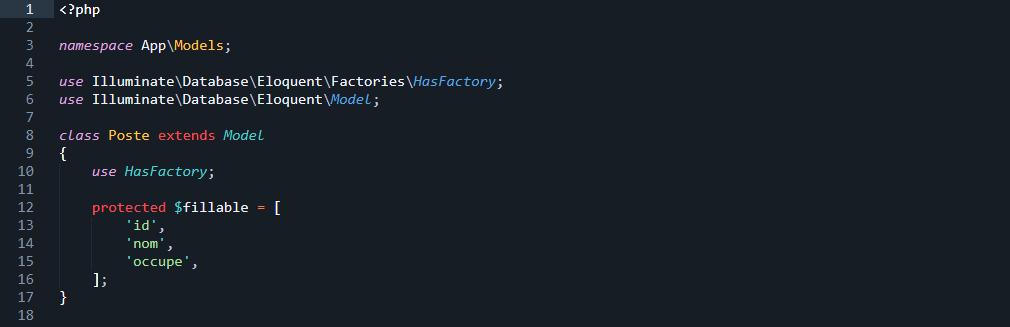
* **App :** Ce dossier contient le code source de l'application Laravel, y compris les modèles (Models), les contrôleurs (Controllers), les middlewares, et d'autres classes PHP spécifiques au projet.
* **Config** : Contient les fichiers de configuration de l'application, y compris les fichiers de configuration pour la base de données, les services, les sessions
* **Public** : Ce dossier est le point d'entrée de l'application. Il contient les fichiers publics tels que les fichiers CSS, JavaScript, les images, les fichiers de polices
* **Database** : Contient les migrations de base de données, les seeders (alimentateurs de données pour les bases de données), et les factories pour générer des jeux de données de test.
* **Ressources** : Contient les fichiers non exécutables comme les fichiers Blade pour les vues, les fichiers de langues, les fichiers Sass, les fichiers JavaScript non compilés :
* Resources/View : Contient les fichiers Blade qui représentent les vues de l'application.
* Resources/Lang : Contient les fichiers de langues pour l'internationalisation
* **Routes** : Contient les fichiers de définition des routes de l'application
* **Storage** : Contient les fichiers générés par l'application, tels que les fichiers de logs, les fichiers de cache, les sessions, etc.
* **Tests** : Contient les fichiers de tests unitaires et d'intégration.
* **Vendor** : Contient les dépendances installées par Composer, le gestionnaire de dépendances de PHP.
* **.env** :

Fichier d'environnement qui stocke les variables d'environnement telles que les informations de base de données, les clés secrètes, etc.

* **Artisan** : Le script artisan, qui fournit une interface en ligne de commande pour diverses tâches de développement.
* **Composer.json et Composer.lock** : Les fichiers de configuration pour Composer, qui gère les dépendances de l'application.

1. **Modèles (Models)** 
   1. Description des modèles
      1. Poste

Le model Poste représente les différents postes présents dans chaque structure qui utilise l’application. Les principaux attributs sont :



* Id = l’identifiant du poste
* Nom = le nom du poste
* Occupe = paramètre permettant de savoir si le poste est déjà occupé ou pas

Configuration de la migration Poste :

Chemin : Database/migrations/ 2013\_11\_09\_084133\_create\_postes\_table.php



* + 1. User

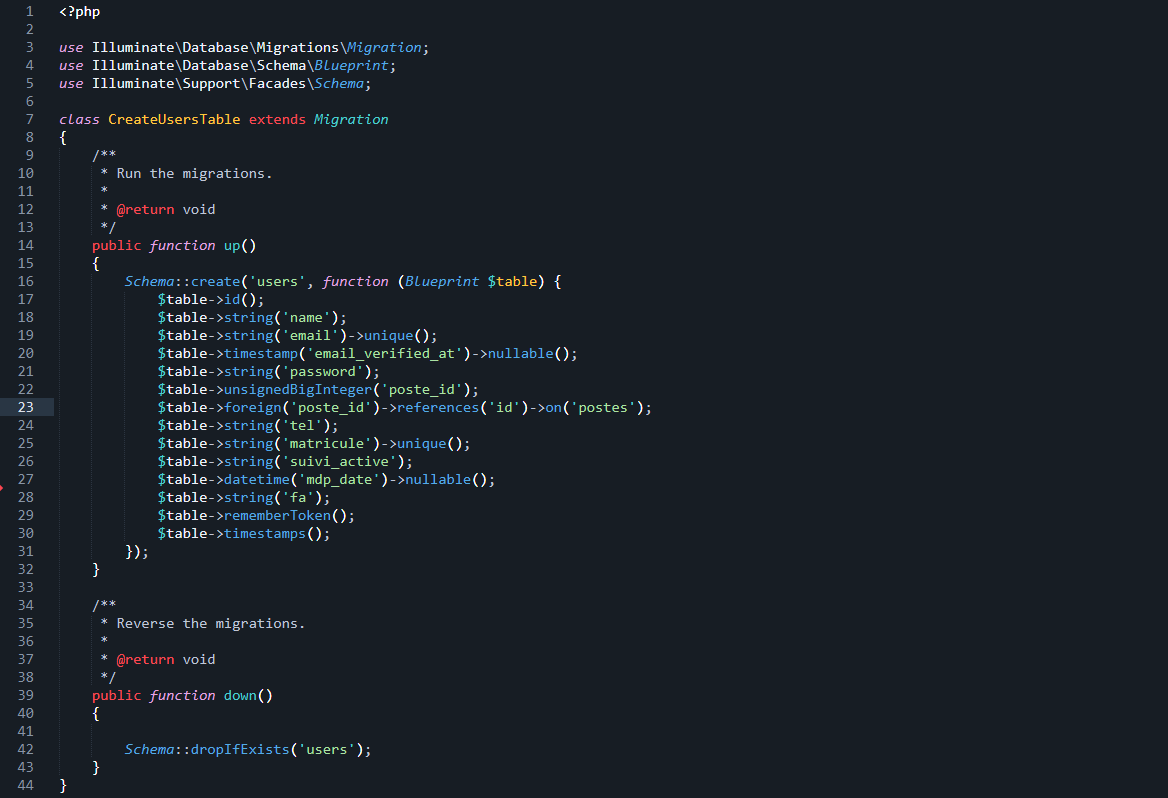
Le model User représente les utilisateurs de l’application. Les principaux attributs sont :



* Id = l’identifiant de l’utilisateur
* Name = le nom et prénom de l’utilisateur
* Email = l’email de l’utilisateur
* Password = le mot de passe de l’utilisateur
* Matricule = le matricule de l’utilisateur
* Poste\_id = l’identifiant du poste auquel l’utilisateur est affecté
* Suivi\_active = paramètre de suivi d’action de l’utilisateur dans l’application (Save, update, delete et search)
* Mdp\_date = date de modification du mot de passe de l’utilisateur
* Fa = paramètre d’identification a double facteur (pas encore employer dans l’application)
* Tel = contact de l’utilisateur

Configuration de la migration User :

Chemin : Database/migrations/ 2014\_10\_11\_000000\_create\_users\_table.php



* + 1. Autorisation

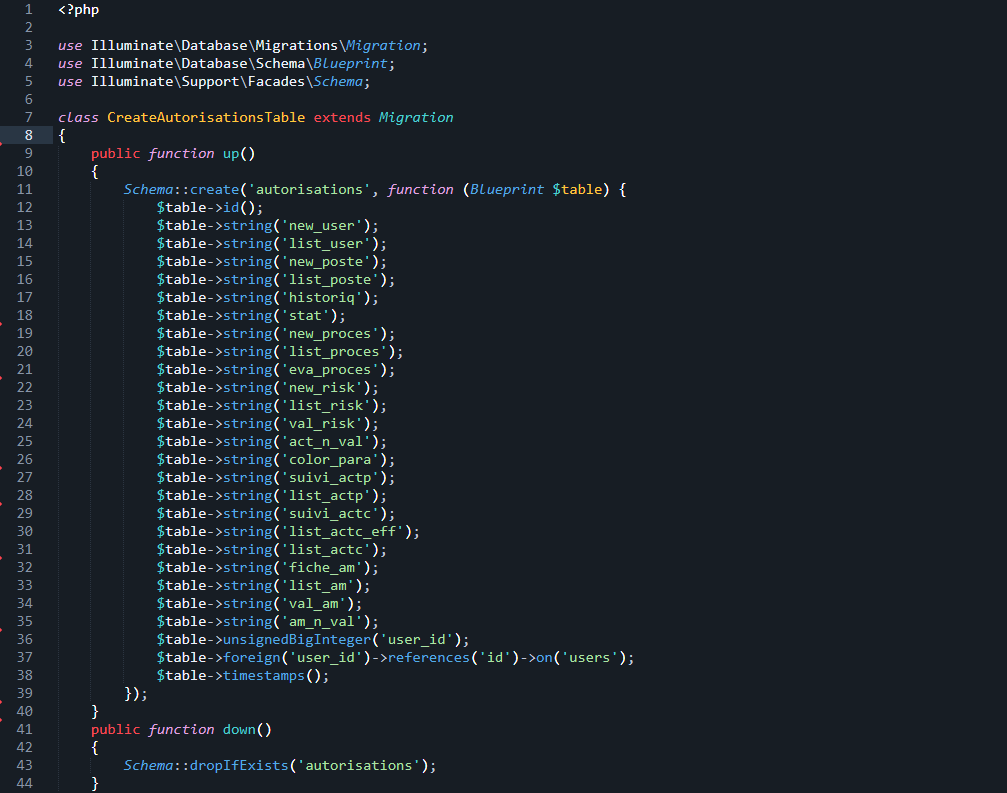
Le model Autorisation représente les accès d’un utilisateur aux différentes pages de l’application. Les principaux attributs sont :



* Id = l’identifiant de l’autorisation
* New\_user = Accès à la page nouveau utilisateur
* List\_user = Accès à la page liste des utilisateurs
* New\_poste = Accès à la page nouveau poste
* Liste\_poste = Accès à la page liste des postes
* Historiq = Accès à la page historique
* Stat = Accès à la page des statistiques
* New\_proces = Accès à la page nouveau processus
* List\_proces = Accès à la page liste des processus
* Eva\_proces = Accès à la page évaluation des processus
* New\_risk = Accès à la page nouveau risque
* List\_risk = Accès à la page liste des risques
* Val\_risk = Accès à la page tableau de validation des risques
* Act\_n\_val = Accès à la page Risques non validés
* Color\_para = Accès à la page paramétrage des couleurs
* Suivi\_actp = Accès à la page suivi des actions préventives
* List\_actp = Accès à la page liste des actions préventives
* Suivi\_actc = Accès à la page suivi des actions correctives
* List\_actc\_eff = Accès à la page liste des actions correctives effectuées
* List\_actc = Accès à la page liste des actions correctives
* Fiche\_am = Accès à la page fiche de résolution d’incident
* List\_am = Accès à la page liste des incidents
* Val\_am = Accès à la page validations des incidents
* Am\_n\_val = Accès à la page incidents non validés
* User\_id = l’identifiant de l’utilisateur liées à ces différents accès

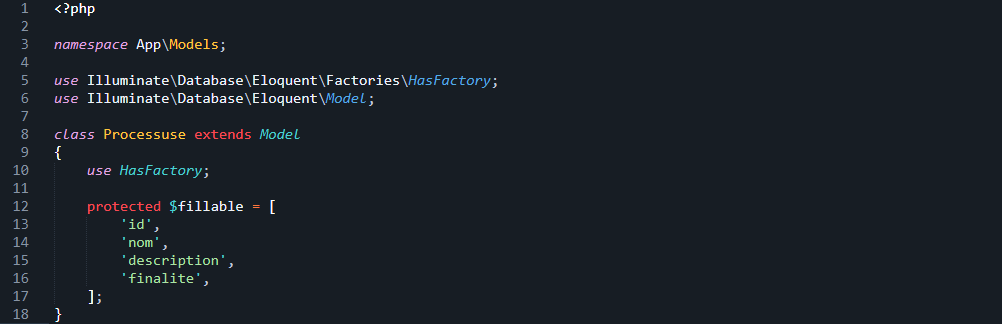
Configuration de la migration Autorisation :

Chemin : Database/migrations/ 2023\_12\_07\_223510\_create\_autorisations\_table.php



* + 1. Processuse

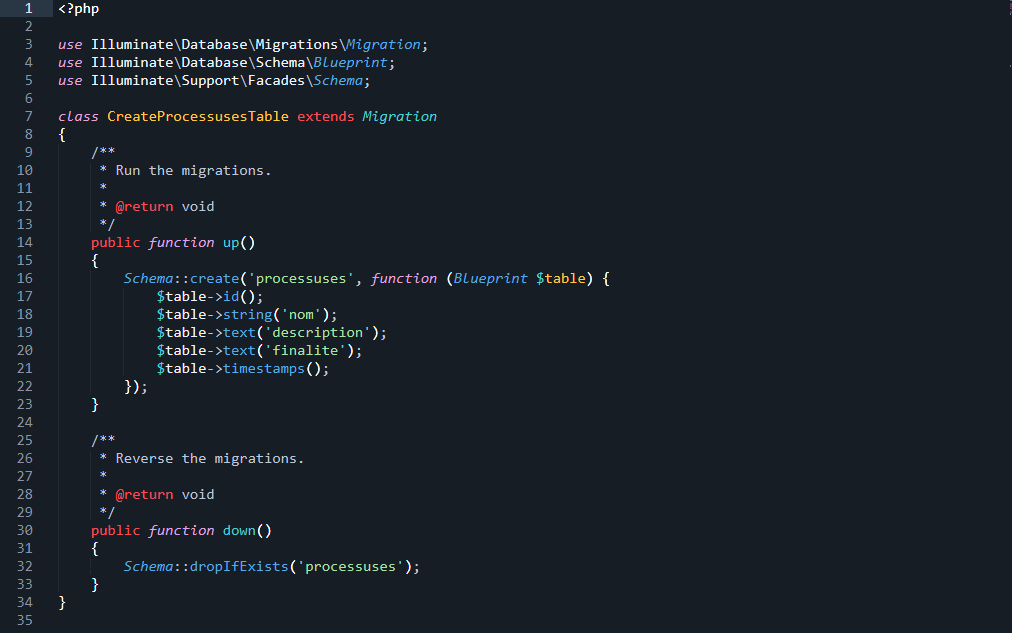
Le model processuse représente le processus de chaque structure qui utilise l’application. Les principaux attributs sont :



* Id = l’identifiant du processus
* Nom = le nom du processus
* Description = la description du processus
* Finalite = la finalité du processus

Configuration de la migration Processuse :

Chemin : Database/migrations/ 2023\_10\_10\_114949\_create\_processuses\_table.php



* + 1. Objectif

Le model objectif représente les objectifs de chaque processus défini dans l’application. Les principaux attributs sont :



* Id = l’identifiant de l’objectif
* Nom = le nom de l’objectif
* Processus\_id = l’identifiant du processus liées a cet objectif

Configuration de la migration objectif :

Chemin : Database/migrations/ 2023\_10\_10\_115038\_create\_objectifs\_table.php



* + 1. Pdf\_file\_processus

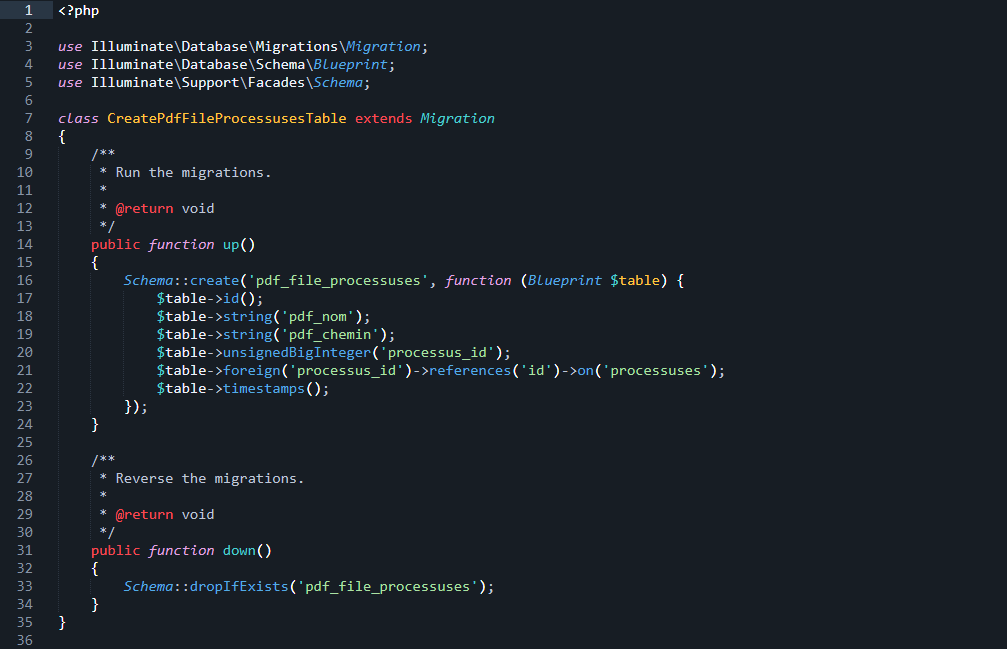
Le model pdf\_file\_processus représente le fichier PDF ajouter a l’enregistrement d’un processus. Les principaux attributs sont :



* Id = l’identifiant de fichier PDF
* Pdf\_nom = le nom du fichier PDF
* Pdf\_chemin = le chemin du fichier PDF
* Processus\_id = l’identifiant du processus auquel le fichier PDF est associé

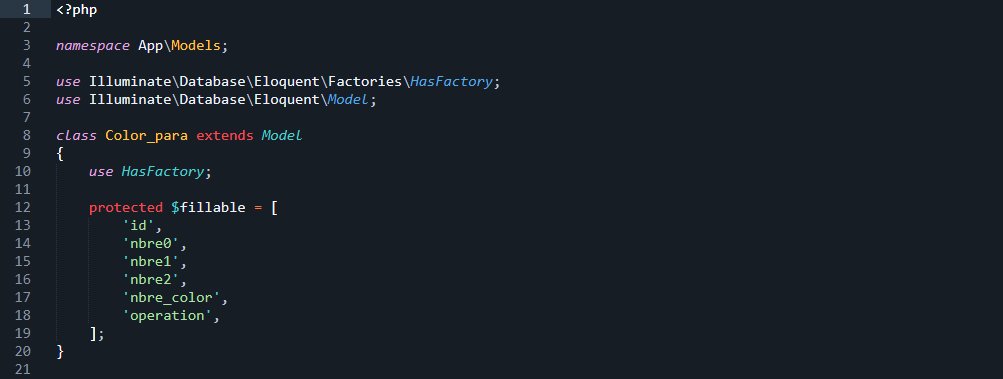
Configuration de la migration pdf\_file\_processus :

Chemin : Database/migrations/ 2023\_11\_18\_212815\_create\_pdf\_file\_processuses\_table.php



* + 1. Color\_para

Le model color\_para représente le paramétrage de base des différentes couleurs et intervalles dans l’application. Les principaux attributs sont :



* Id = l’identifiant du paramètre
* Nbre0 = le nombre sur lequel l’on s’appuie pour apporter des modifications au niveau du paramétrage de base
* Nbre1 = le nombre initial
* Nbre2 = le nombre limite
* Nbre\_color = nombre total d’intervalle ou de couleur
* Operation = l’opération qui sera effectuée dans chaque calcul

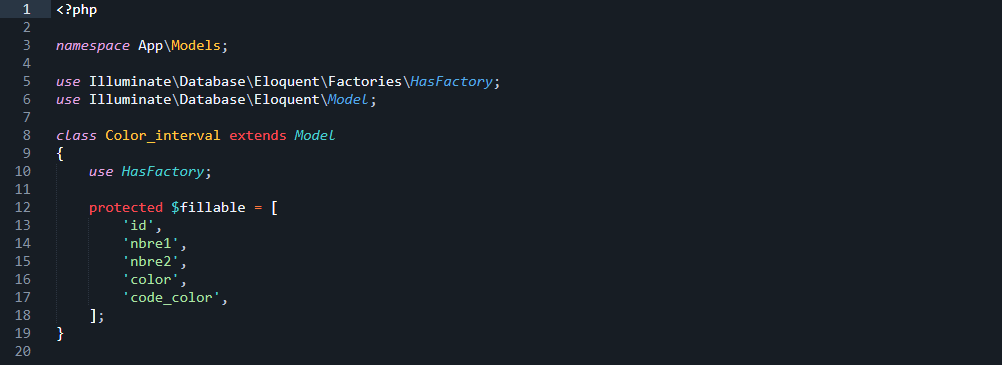
Configuration de la migration color\_para :

Chemin : Database/migrations/ 2024\_01\_05\_130019\_create\_color\_paras\_table.php



* + 1. Color\_interval

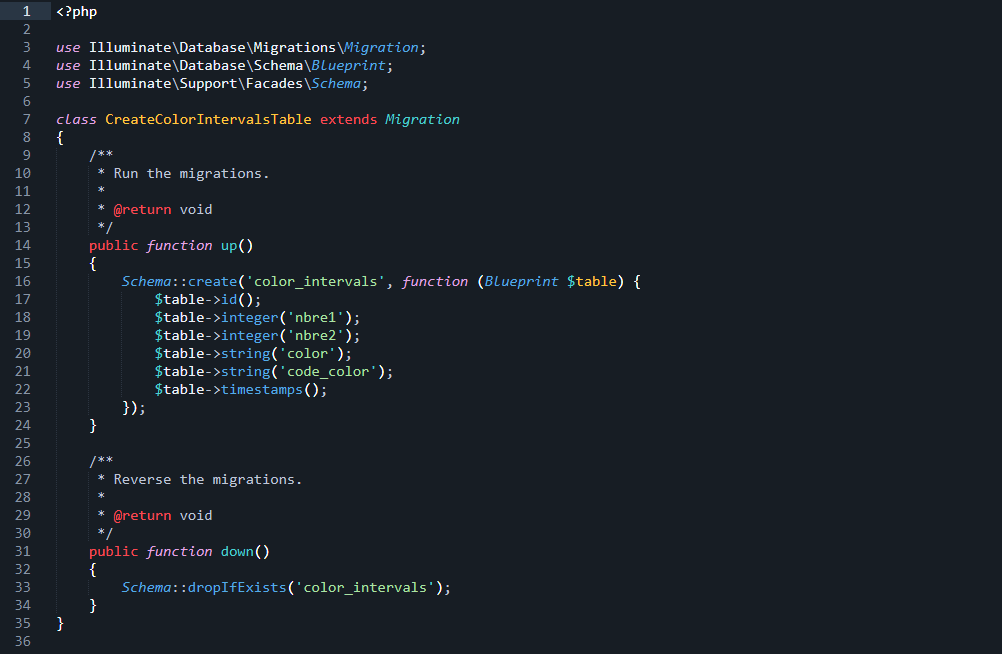
Le model color\_interval représente un intervalle d’évaluation en fonction du paramétrage de base défini. Les principaux attributs sont :



* Id = l’identifiant de l’intervalle
* Nbre1 = le nombre de départ de l’intervalle
* Nbre2 = le nombre d’arriver de l’intervalle
* Color = la couleur de l’intervalle
* Code\_color = le code de la couleur de l’intervalle

Configuration de la migration color\_interval :

Chemin : Database/migrations/ 2024\_01\_05\_130814\_create\_color\_intervals\_table.php



* + 1. Risques

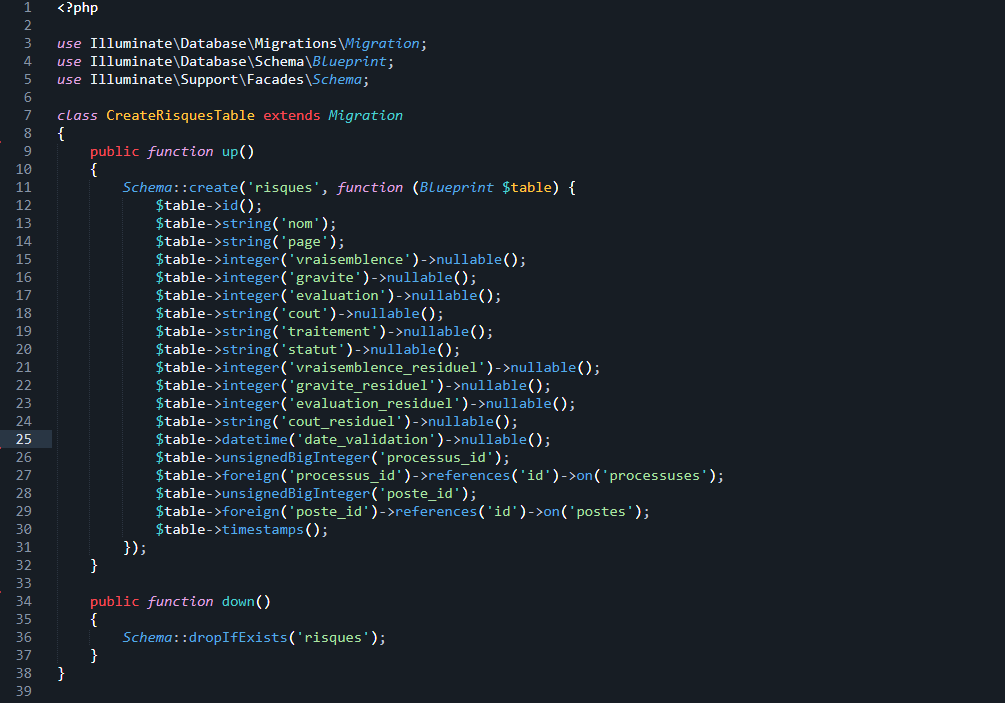
Le model risque représente le risque de chaque structure qui utilise l’application. Les principaux attributs sont :



* Id = l’identifiant du risque
* Nom = le nom du risque
* Page = paramètre permettant de faire la différence entre les risques enregistrer avant et pendant l’incident
* Vraisemblence = indicateur d’évaluation 1
* Gravite = indicateur d’évaluation 2
* Evaluation = le résultat de l’opération (opération est défini dans le paramétrage des couleurs) effectuée entre la vraisemblance et la gravite
* Cout = la somme qui sera dépenser si et seulement si le risque survient
* Vraisemblence\_residuel = indicateur d’évaluation résiduel 1
* Gravite\_residuel = indicateur d’évaluation résiduel 2
* Evaluation\_residuel = le résultat de l’opération (opération est défini dans le paramétrage des couleurs) effectuée entre la vraisemblance résiduelle et la gravite résiduel
* Cout\_residuel = la somme qui sera dépenser si le risque survient et après avoir mener à bien les actions préventives
* Date\_de validation = la date a laquelle la ficher risque a été validé
* Processus\_id = l’identifiant du processus auquel le risque est lié
* Staut = le niveau de traitement de la fiche risque (valider, en attente de validation, etc… )
* Traitement = l’objectif finale si le risque arrive (réduire le risque, accepter le risque, etc…)
* Poste\_id = l’identifiant du poste charger de la validation de la fiche risque

Configuration de la migration risque :

Chemin : Database/migrations/ 2023\_10\_11\_225439\_create\_risques\_table.php



* + 1. Causes

Le model cause représente les causes probables de chaque risque. Les principaux attributs sont :



* Id = l’identifiant de la cause
* Nom = le nom de la cause
* Page = paramètre permettant de faire la différence entre les causes liées à des risques enregistrer avant et pendant l’incident
* Dispositif = le dispositif de contrôle de la cause
* Risque\_id = l’identifiant du risque auquel la cause est liée

Configuration de la migration cause :

Chemin : Database/migrations/ 2023\_10\_11\_225524\_create\_causes\_table.php



* + 1. Pdf\_file

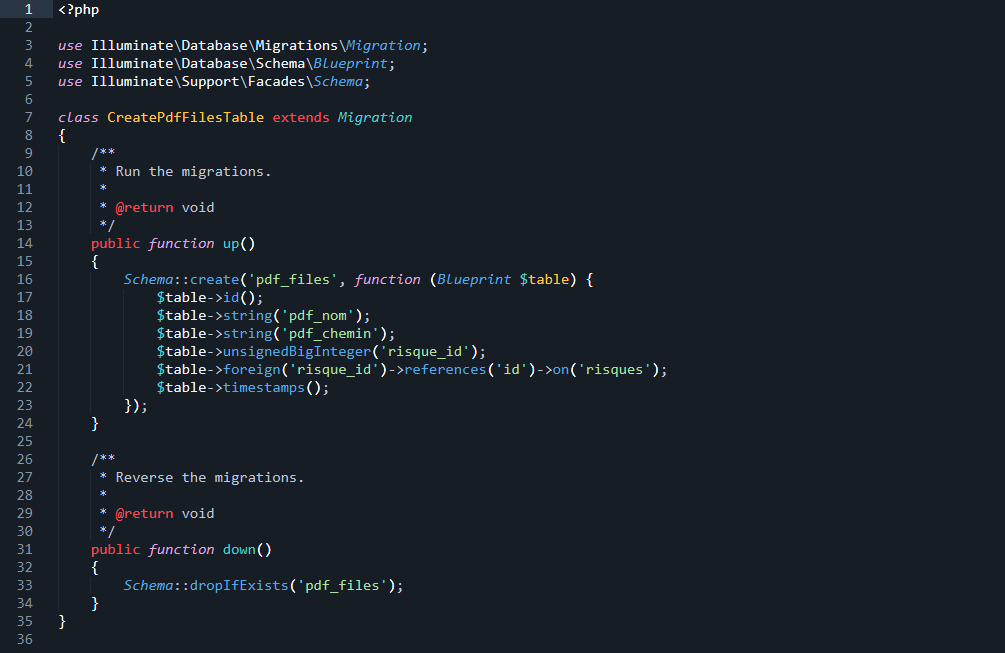
Le model Pdf\_file représente le fichier PDF ajouter à l’enregistrement d’un risque. Les principaux attributs sont :



* Id = l’identifiant de fichier PDF
* Pdf\_nom = le nom du fichier PDF
* Pdf\_chemin = le chemin du fichier PDF
* Risque\_id = l’identifiant du risque auquel le fichier PDF est associé

Configuration de la migration Pdf\_file :

Chemin : Database/migrations/ 2023\_10\_23\_162437\_create\_pdf\_files\_table.php



* + 1. Rejet

Le model rejet représente l’ensemble des fiche risque rejeté après vérification. Les principaux attributs sont :



* Id = l’identifiant du rejet
* Motif = le motif pour lequel le validateur a rejeté le risque
* Risque\_id = l’identifiant du risque rejeté

Configuration de la migration Rejet :

Chemin : Database/migrations/ 2023\_10\_13\_150745\_create\_rejets\_table.php



* + 1. Action

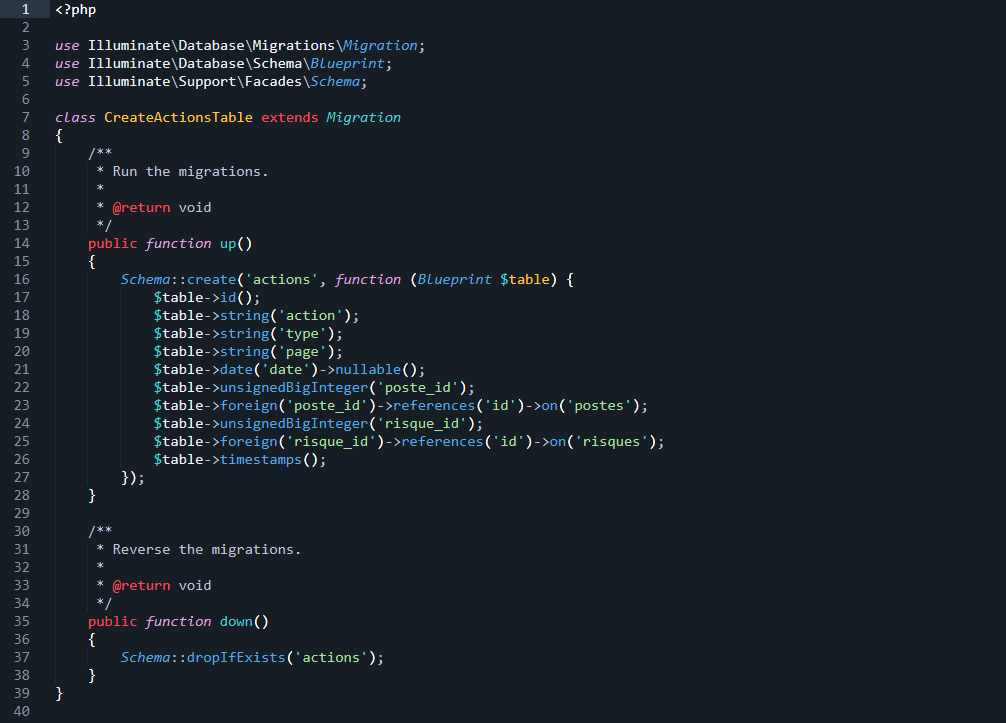
Le model action représente les différentes actions (préventives et correctives) liées a un risque. Les principaux attributs sont :



* Id = l’identifiant de l’action
* Action = le nom de l’action
* Type = le type de l’action (préventive ou corrective)
* Page = le paramètre permettant de savoir ou l’action a été enregistrer
* Date = la date limite de traitement de l’action (action préventive uniquement)
* Poste\_id = l’identifiant du poste charger de mener l’action
* Risque\_id = l’identifiant du risque auquel l’action appartient

Configuration de la migration Action :

Chemin : Database/migrations/ 2023\_10\_14\_221834\_create\_actions\_table.php



* + 1. Suivi\_action

Le model Suivi\_action représente le suivi des actions préventives. Les principaux attributs sont :



* Id =
* Efficacité =
* Commentaire =
* Date\_action =
* Date\_suivi =
* Statut =
* Processus\_id =
* Risque\_id
  + 1. Amelioration
    2. Rejet\_am
    3. Causetrouver
    4. Risquetrouver
    5. Suivi\_amelioration
    6. Historique\_action
  1. Relations entre les modèles

1. **Contrôleurs (Controller)**
   1. Description des contrôleurs
   2. Route associée
2. **Vues (View)**
   1. Structure des vues
   2. Utilisation de blade
3. **Middlewares**
   1. Liste des middlewares utilisés
   2. Fonctionnement et utilisation
4. **Evènements (Event)**
   1. Liste des évènement utilisés
   2. Fonctionnement
5. **Base de données**
   1. Schéma de la base de données
   2. Principales requêtes SQL
6. **Authentification**
7. **Tests**