

# Guide développeur

Dans ce guide, nous proposons de passer en revue le fonctionnement du programme, en explicitant les méthodes de chaque classe qui le compose.

## ➔ ArticleScrap

Cette classe sert à instancier des objets ArticleScrap représentant chaque article trouvé validant les critères de recherche. Chaque objet ArticleScrap a pour attribut : un titre (String), un genre (String), une année de parution (int), un prix (double), une description (String), ainsi qu'une url pour y accéder (String). La classe ArticleScrap comprend un constructeur ainsi que des getter et setter.

### toString()

Cette méthode renvoie une chaîne de caractères comprenant toutes les informations de l'article en question.

## ➔ Scrap

Cette classe sert à effectuer le scraping sur les différents sites. Les différentes méthodes de scraping étant fortement similaires entre les différents sites, nous proposons de passer en revue le fonctionnement d'une d'entre elles :

### scrapFnac(String titre, double min, double max, int annee,String genreChoisi)

Cette méthode effectue le scraping du site de la Fnac. Elle prend en paramètres le titre de la recherche, un prix minimum et un prix maximum, une année ainsi qu'un genre. Une fois connecté au site en question via l'url incluant le titre de notre recherche, on récupère tous les articles en passant par un XPath, puis on les range dans une liste. En parcourant ensuite cette liste, on vient cliquer sur le lien de l'article afin d'accéder à sa page produit. Une fois sur cette page on récupère la description de l'article, sa date de parution, ainsi que son prix (toujours via un XPath). Sur le site de la Fnac, la date de parution est représentée par un mois ainsi qu'une année. Dans la date que nous avons récupéré, il faut donc supprimer tous les caractères qui ne sont pas numériques (afin d'éliminer le mois) en appelant la méthode replaceAll(), puis caster la date en int. Ainsi, nous pouvons comparer l'année de parution avec l'année choisie par l'utilisateur. Si l'album est sorti avant la date limite choisie, on peut ensuite vérifier si le prix entre bien dans notre intervalle de recherche. Pour cela, on appelle la fonction stringToDouble() dont nous parlerons plus loin dans ce guide. Après comparaison du prix et si celui-ci entre bien dans notre intervalle, on récupère la description de l'article (si elle existe). Enfin, tous les critères de recherche étant validés, on peut créer un nouvel objet ArticleScrap qu'on vient ensuite ajouter à l'ArrayList contenant tous les résultats de notre recherche.

### stringToDouble(String prixInitial)

Nous l'avons vu plus tôt, le prix que nous récupérons sur la page de l'article est un String, rendant impossibles les opérations de comparaison afin de vérifier si celui-ci est bien compris dans notre intervalle de recherche. Ainsi, cette méthode a pour but de caster ce prix en double. Pour cela, on commence par supprimer tous les caractères contenus dans cette String qui ne sont ni des chiffres, ni une virgule, en utilisant la fonction replaceAll(). On utilise ensuite la méthode replace() afin de

remplacer la virgule du prix par un point. Notre string est maintenant prêt à être casté en double, rendant possibles les opérations de comparaison avec les bornes de notre intervalle de prix souhaité.

#### genreDiscogs(String genre)

Certains sites comme Discogs permettent de faire une recherche de disques par genre. On peut passer le genre souhaité directement dans l'url, mais pour cela il faut respecter le formalisme mis en place par le site. Cette méthode permet donc de retourner l'expression du genre appropriée pour Discogs, en fonction du genre entré. Ainsi sur ce site, les catégories Funk et Soul fusionnent en une seule catégorie « Funk+2%F+Soul », et le genre Electro devient « Electronic ». Le genre Dubstep, quant à lui, n'existe tout simplement pas sur le site de Discogs. Tous les autres genres gardent leur même nom.

#### genreVinylCorner(String genre)

Cette méthode fonctionne grossièrement de la même façon que la précédente, au détail près que Vinyl Corner utilise un système de nombre afin de classer les différentes catégories. Ainsi cette fonction renvoie un entier correspondant au genre désiré pour la recherche.

### ➔ EnvoiMail

Cette classe regroupe les méthodes relatives à la fonctionnalité d'envoi de mail.

#### save(String res)

Cette méthode a pour but de sauvegarder les résultats d'une recherche dans un fichier texte afin de pouvoir le mettre en pièce jointe d'un mail par la suite.

#### envoyerMail(String adresseMail,String nomFichier)

Cette méthode a pour but d'envoyer un mail contenant les résultats d'une recherche à l'adresse mail renseignée. Pour cela on utilise l'API de Sendinblue afin d'utiliser un serveur SMTP en passant par une clé API générée pour l'occasion.

### ➔ EnregistrementBDD

Cette classe prend en paramètre la requête SQL que nous souhaitons appliquer à la base de données, dans une String.

#### saveBDD(String titre,String description, double prix, int genre, int annee)

Cette fonction a pour but d'envoyer les résultats d'une recherche dans une base de données. Pour cela, on récupère tout d'abord les informations de connexion à la base contenues dans un fichier texte enregistré plus tôt, puis on tente la connexion à la base en utilisant ces informations. Une fois la connexion établie, on peut appliquer la requête SQL en passant en paramètres les différentes informations de chaque article, envoyant ainsi tous les résultats d'une recherche dans la base de données.

## ➔ VinyleController

Cette classe gère tout ce qui est relatif à l'interface principale de l'application (la fenêtre qui s'ouvre quand on lance le programme). Elle a pour attribut une ArrayList de ArticleScrap contenant tous les résultats de la recherche en cours.

### affichageListe()

Cette méthode parcourt l'ArrayList et retourne chaque article qui la compose sous forme d'une chaîne de caractères.

### clear()

Cette méthode sert à réinitialiser tous les champs du formulaire de recherche lorsque l'utilisateur clique sur le bouton « effacer ».

### refreshComboBox()

Cette méthode a pour but de faire prendre à la comboBox sa valeur par défaut (soit le promptText).

### quitter()

Cette méthode sert à fermer la fenêtre principale lorsque l'utilisateur clique sur le sous-menu « Quitter ».

### masqueSaisie()

Cette méthode vérifie si les champs du formulaire contenant des prix contiennent bien des nombres. Elle retourne un boolean prenant la valeur true si c'est le cas, et false sinon.

### nbScrap()

Cette méthode comptabilise le nombre de sites sélectionnés pour la recherche et renvoie un entier.

### onRechercherClick()

Cette méthode est appelée lorsque l'utilisateur clique sur le bouton « Rechercher ». On appelle ensuite la fonction masqueSaisie() et si celle-ci retourne true, on appelle la fonction doTheScrap() qui effectuera le scraping.

### doTheScrap()

Cette méthode sert à effectuer le scraping sur les différents sites sélectionnés. L'essentiel de cette méthode est effectué dans une tâche (classe Task héritant de la classe Thread) afin de permettre à l'utilisateur de continuer à utiliser l'application pendant la recherche. On commence par récupérer toutes les contenus des champs du formulaire de recherche. On parse les prix en double car on les récupère en String puis on crée un nouvel objet Scrap, et on appelle la méthode nbScrap() afin de savoir combien de sites sont sélectionnés. Pour chaque site sélectionné, on appelle la fonction de Scrap correspondante en passant en paramètres les informations nécessaires. Une fois le scrap de chaque site effectué, on met à jour la progression de la barre de progression afin de faire connaître à l'utilisateur l'avancée de la recherche.

### onEnregistrerFichierClick()

Cette méthode sert à enregistrer les résultats d'une recherche dans un fichier texte, lorsque l'utilisateur clique sur le sous-menu « Enregistrer dans un fichier ». Une fenêtre contextuelle s'ouvre alors afin de permettre à l'utilisateur de sélectionner l'emplacement d'enregistrement.

### popUpMail()

Cette méthode est appelée par la méthode onMailClick() lorsque l'utilisateur clique sur le sous-menu « Envoyer un mail ». Elle ouvre une fenêtre pop-up permettant à l'utilisateur de rentrer son adresse mail. Lorsqu'il clique sur « Envoyer », la méthode onEnvoyerMailClick() est appelée afin d'envoyer un mail à l'adresse entrée, contenant en pièce jointe le fichier texte comprenant les résultats de la recherche.