

Résanet - Modèle

Services fournis par le modèle (fichier `modeles/modeleResanet.py`)

dict **seConnecterGestionnaire**(str login , str mdp)

S'authentifier en tant que gestionnaire.

Paramètres :

login : Nom de connexion

mdp : Mot de passe

Valeur retournée :

- Un dictionnaire qui recense le nom de connexion (*login*), le nom et le prénom du gestionnaire si l'authentification a réussi
- Un dictionnaire vide si l'authentification a échoué
- None si la base de données est inaccessible

dict **seConnecterUsager**(int numeroCarte , str mdpCarte)

S'authentifier en tant qu'usager.

Paramètres :

numeroCarte : Numéro de la carte de restauration

mdpCarte : Mot de passe

Valeur retournée :

- Un dictionnaire avec numéro de carte, solde, état de la carte, nom et prénom de l'usager si l'authentification a réussi
- Un dictionnaire vide si l'authentification a échoué
- None si la base de données est inaccessible

float **getSolde**(int numeroCarte)

Obtenir le solde d'une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- Le solde de la carte de restauration si la carte est recensée
- "inconnu" si la carte de restauration n'est pas recensée
- None si la base de données est inaccessible

float **getTarifRepas**(int numeroCarte)

Obtenir le tarif appliqué à une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- Le tarif si la carte est recensée
- "inconnu" si la carte de restauration n'est pas recensée
- None si la base de données est inaccessible

int debiterSolde(int numeroCarte)

Débiter une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- 1 si la carte de restauration a été débitée
- 0 si la carte de restauration n'a pas été débitée
- None si la base de données est inaccessible

int crediterSolde(int numeroCarte)

Créditer une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- 1 si la carte de restauration a été créditée
- 0 si la carte de restauration n'a pas été créditée
- None si la base de données est inaccessible

int enregistrerReservation(int numeroCarte , str dateReservation)

Enregistrer une réservation.

Paramètres :

numeroCarte : Numéro de la carte de restauration

dateReservation : Date du repas au format ISO 8601

Valeur retournée :

- 1 si la réservation a été enregistrée
- 0 si la réservation n'a pas été enregistrée
- None si la base de données est inaccessible

int annulerReservation(int numeroCarte , str dateReservation)

Annuler une réservation.

Paramètres :

numeroCarte : Numéro de la carte de restauration

dateReservation : Date du repas au format ISO 8601

Valeur retournée :

- 1 si la réservation a été annulée
- 0 si la réservation n'a pas été annulée
- None si la base de données est inaccessible

list getReservationsCarte(int numeroCarte , str dateDebut , str dateFin)

Obtenir la liste des réservations d'une carte de restauration sur une période définie.

Paramètres :

numeroCarte : Numéro de la carte de restauration

dateDebut : Date de début de la période au format ISO 8601

dateFin : Date de fin de la période au format ISO 8601

Valeur retournée :

- Liste des dates des réservations
- None si la base de données est inaccessible

int modifierMdpUsager(int numeroCarte, str ancienMDP , str nouveauMDP)
Modifier le mot de passe d'un compte de restauration.

Paramètres :

numeroCarte : Numéro de la carte de restauration
ancienMDP : Ancien mot de passe
nouveauMDP : Nouveau mot de passe

Valeur retournée :

- 1 si le nouveau mot de passe a été pris en compte
- 0 si le nouveau mot de passe n'a pas été pris en compte
- None si la base de données est inaccessible

list getReservationsDate(str dateResa)

Obtenir la liste des réservations enregistrées pour une date donnée.

Paramètre :

dateResa : Date du repas au format ISO 8601

Valeur retournée :

- Une liste de dictionnaires. Chaque dictionnaire correspond à une réservation et recense le numéro de la carte de réservation, le nom, le prénom et le nom du service de l'utilisateur.
- None si la base de données est inaccessible

list getPersonnelsSansCarte()

Obtenir la liste des membres du personnel de la société qui ne possèdent pas de carte de restauration.

Valeur retournée :

- Une liste de dictionnaires. Chaque dictionnaire correspond à un membre du personnel de la société et recense le matricule, le nom, le prénom et le nom du service de l'utilisateur.
- None si la base de données est inaccessible

list getPersonnelsAvecCarte()

Obtenir la liste des membres du personnel de la société qui possèdent une carte de restauration.

Valeur retournée :

- Une liste de dictionnaires. Chaque dictionnaire correspond à un membre du personnel de la société et recense le numéro de carte de restauration, le solde, l'état de la carte (carte activée ou carte bloquée), le matricule, le nom, le prénom et le nom du service de l'utilisateur.
- None si la base de données est inaccessible

list getHistoriqueReservationsCarte(int numeroCarte)

Obtenir l'historique des réservations d'une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- Liste des dates de réservation au format ISO 8601
- None si la base de données est inaccessible

int activerCarte(int numeroCarte)

Activer une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- 1 si la carte de restauration a été activée
- 0 si la carte de restauration n'a pas été activée
- None si la base de données est inaccessible

int bloquerCarte(int numeroCarte)

Bloquer une carte de restauration.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- 1 si la carte de restauration a été bloquée
- 0 si la carte de restauration n'a pas été bloquée
- None si la base de données est inaccessible

int creerCarte(int matricule , bool activee = False)

Créer un compte de restauration pour un membre du personnel de la société.

Paramètres :

matricule : Matricule du membre du personnel

activee : True si la carte de restauration doit être activée à sa création et False dans le cas contraire

Valeur retournée :

- 1 si le compte de restauration a été créé
- 0 si le compte de restauration n'a pas été créé
- None si la base de données est inaccessible

int crediterCarte(int numeroCarte , float somme)

Créditer un compte de restauration.

Paramètres :

numeroCarte : Numéro de la carte de restauration

somme : Somme à créditer

Valeur retournée :

- 1 si le compte de restauration a été crédité
- 0 si le compte de restauration n'a pas été crédité
- None si la base de données est inaccessible

int reinitialiserMDP(int numeroCarte)

Réinitialiser le mot de passe d'un compte de restauration. Le mot de passe se voit affecter pour valeur, l'année de naissance de l'utilisateur.

Paramètre :

numeroCarte : Numéro de la carte de restauration

Valeur retournée :

- 1 si le mot de passe a été réinitialisé
- 0 si le mot de passe n'a pas été réinitialisé
- None si la base de données est inaccessible