

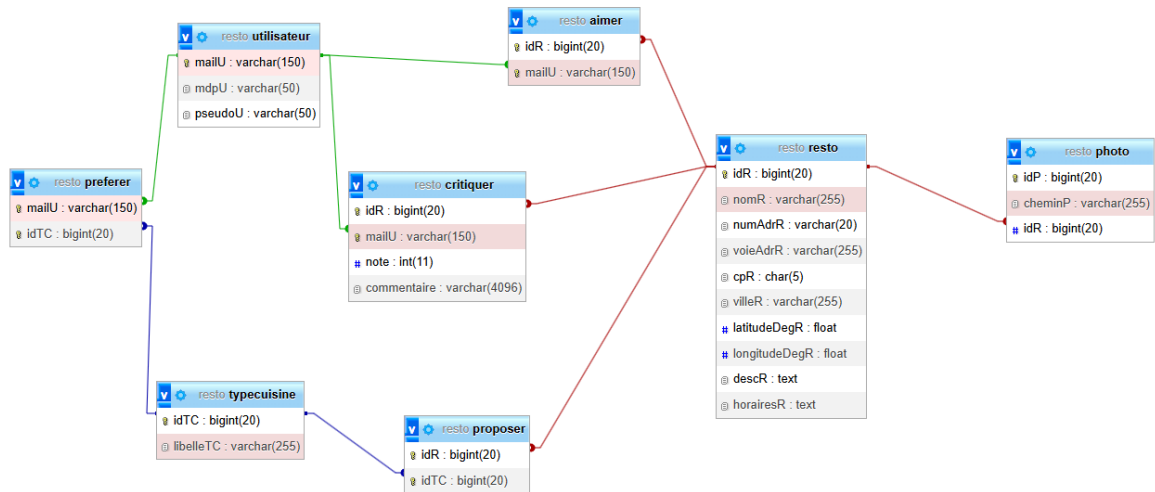
Documentation Projet Resto

Célia Baylet

-Visual Studio Code

-MVC

-PHP



[Accueil](#)  [Connexion](#)

Connexion

[Inscription](#)

Utilisateur de test :
login : test@bts.sio
mot de passe : sio

Accueil

Recherche

★★★★★

r3st0.fr

Connexion

Liste des restaurants



l'entrepote

2 rue Maurice Ravel 33000 Bordeaux

#sud ouest



le bar du charcutier

30 rue Parlement Sainte-Catherine 33000 Bordeaux

#sud ouest



Sapporo

33 rue Saint Rémi 33000 Bordeaux

#orientale



Cidrerie du fronton

Place du Fronton 64210 Arbonne

#sud ouest #sandwich #grillade



Agadir

3 Rue Sainte-Catherine 64100 Bayonne

#orientale



Le Bistrot Sainte Cluque

9 Rue Hugues 64100 Bayonne

#viande



la petite auberge

15 rue des cordeliers 64100 Bayonne

#vegan



La table de POTTOKA

21 Quai Amiral Dubourdieu 64100 Bayonne

#grillade

Accueil

Recherche

★★★★★

r3st0.fr

Connexion

★★★★★

r3st0.fr

Le restaurant

Adresse

Photos

Horaires

Critiques

l'entrepote

★

★★★★★

Type de cuisine

#sud ouest

Description

description

Adresse

2 rue Maurice Ravel
33000 Bordeaux

Photos




Horaires

Ouverture	Semaine	Week-end
Midi	de 11h45 à 14h30	de 11h45 à 15h00
Soir	de 18h45 à 22h30	de 18h45 à 1h
À emporter	de 11h30 à 23h	de 11h30 à 2h

[Accueil](#) [Recherche](#)  [Deconnexion](#)



[Recherche par nom](#)
[Recherche par adresse](#)
[Recherche par type de cuisine](#)

Recherche d'un restaurant

Recherche par type de cuisine :

- ☐ sud ouest
- ☐ japonaise
- ☐ orientale
- ☐ fastfood
- ☐ végétarienne
- ☐ vegan
- ☐ crêpe
- ☐ sandwich
- ☐ tartes
- ☐ viande
- ☐ grillade

Modele

authentification.php :

Ce modèle gère l'authentification des utilisateurs :

- `login($mailU, $mdpU)` : vérifie les identifiants, et si le mot de passe est correct, il connecte l'utilisateur en enregistrant ses infos en session.
- `logout()` : déconnecte l'utilisateur en supprimant ses données de session.
- `isLoggedOn()` : retourne true si l'utilisateur est connecté (et que ses données sont valides).
- `getMailULoggedOn()` : retourne l'adresse mail de l'utilisateur connecté, ou une chaîne vide s'il ne l'est pas.

bd.aimer.php :

Ce modèle gère les opérations liées à l'interaction d'un utilisateur avec un restaurant (ajout ou suppression de "j'aime") :

- `getAimerById($mailU, $idR)` :

Récupère toutes les entrées de la table aimer pour un utilisateur donné (mailU) et un restaurant donné (idR).

Utilise une requête SELECT pour obtenir les données, puis retourne le résultat sous forme d'objet avec PDO::FETCH_OBJ.

- `addAimer($mailU, $idR)` :

Insère une nouvelle ligne dans la table aimer, marquant que l'utilisateur (mailU) aime le restaurant (idR).

Utilise une requête INSERT INTO avec des paramètres liés pour prévenir les injections SQL.

- delAimer(\$mailU, \$idR) :

Supprime une entrée de la table aimer où l'utilisateur (mailU) aime le restaurant (idR).

Utilise une requête DELETE avec des paramètres liés pour enlever l'entrée spécifique.

bd.critiquer.php :

La fonction getCritiquerByIdR(\$idR) récupère toutes les critiques associées à un restaurant spécifique (identifié par idR) dans la table critiquer. Elle utilise une requête préparée pour sécuriser l'accès à la base de données. Les résultats sont renvoyés sous forme d'objets. En cas d'erreur, une exception est capturée et un message d'erreur est affiché.

bd.pdo.php :

Cette fonction gère la connexion à la base de données via PDO :

- connexionPDO() : elle crée une connexion à la base de données MySQL en utilisant les informations de connexion (login, mdp, bd, serveur), configure l'encodage en UTF-8 et définit le mode de gestion des erreurs sur PDO::ERRMODE_EXCEPTION pour faciliter le débogage.
- En cas d'échec, elle attrape l'exception et affiche un message d'erreur puis termine le script avec die().

bd.photo.php :

Le modèle contient deux fonctions pour récupérer des photos associées à un restaurant :

- getLaPhotoByIdR(\$idR) : Récupère une seule photo pour un restaurant donné, en utilisant l'ID du restaurant.
- getLesPhotosByIdR(\$idR) : Récupère toutes les photos d'un restaurant, en utilisant également l'ID du restaurant.

bd.resto.php :

- getLesRestos() : Récupère tous les restaurants de la table resto et retourne les résultats sous forme d'objets.
- getLeRestoByIdR(\$idR) : Récupère un restaurant spécifique en fonction de son idR et retourne les résultats sous forme d'objet.
- getLesRestosByNomR(\$nomR) : Recherche des restaurants dont le nom contient le critère nomR et retourne les résultats sous forme d'objets.

- getLesRestosByAdresse(\$voieAdrR, \$cpR, \$villeR) : Recherche des restaurants en fonction de leur adresse (voie, code postal et ville) et retourne les résultats sous forme d'objets.
- getLesRestosByTC(\$tabIdTC) : Recherche des restaurants en fonction des types de cuisine associés à leur ID et retourne les résultats sous forme d'objets si le tableau tabIdTC contient des valeurs.

bd.typecuisine.php :

- getLesTypesCuisinebyIdR(\$idR) : Récupère les types de cuisine associés à un restaurant spécifique (idR) via une jointure entre les tables typecuisine, proposer, et resto, et retourne les résultats sous forme d'objets.
- getLesTypesCuisine() : Récupère tous les types de cuisine disponibles dans la table typecuisine et retourne les résultats sous forme d'objets.

bd.utilisateur.php :

getLeUtilisateurByMailU(\$mailU) : Cette fonction récupère les informations d'un utilisateur depuis la base de données en utilisant son adresse email (mailU). Elle effectue une requête préparée sur la table utilisateur et retourne les résultats sous forme d'objet.

Vue

entete.html.php :

La vue entete.html.php crée l'en-tête de la page avec un menu de navigation principal contenant des liens vers l'accueil, la recherche, et la gestion de connexion (connexion/déconnexion selon l'état de l'utilisateur). Elle inclut aussi un menu contextuel (menu burger)

vueAuthentification.php :

Cette vue affiche un formulaire de connexion avec des champs pour l'email et le mot de passe, ainsi qu'un lien vers une page d'inscription. Elle propose aussi un utilisateur de test avec ses identifiants (login et mot de passe) pour faciliter l'accès.

vueConfirmation.php :

Cette vue affiche l'adresse électronique de l'utilisateur actuellement connecté en utilisant la fonction getMailULoggedIn(). Elle inclut également un lien permettant à l'utilisateur de se déconnecter via l'action deconnexion.

vueDetailResto.php :

Ce code affiche les informations détaillées d'un restaurant, telles que son nom, sa cuisine, sa description, son adresse et ses horaires. Il génère aussi une section pour afficher les photos du restaurant et les critiques des utilisateurs. Un bouton permet d'ajouter le restaurant aux préférés de l'utilisateur connecté. Les types de cuisine associés au restaurant sont affichés sous forme de tags. Le code utilise plusieurs fonctions pour récupérer ces informations depuis la base de données et les afficher dynamiquement.

vueListeRestos.php :

Ce code affiche une liste de restaurants sous forme de cartes. Pour chaque restaurant, il récupère une photo associée et les types de cuisine, puis génère une carte avec l'image, le nom du restaurant, son adresse et ses types de cuisine. Chaque carte est cliquable et mène à une page de détails du restaurant. Les types de cuisine sont affichés sous forme de tags.

vueRechercheResto.php :

Ce code affiche un formulaire de recherche de restaurant avec différentes options selon le critère sélectionné (nom, adresse ou type de cuisine). Si le critère est "nom", il permet de rechercher par nom; si "adresse", par rue, code postal et ville; et si "type", il propose des cases à cocher pour les types de cuisine. Le formulaire envoie la recherche via la méthode POST. Les résultats de la recherche seront basés sur le critère choisi par l'utilisateur.

Contrôleur

controleurPrincipal.php :

Cette fonction PHP retourne le nom du fichier à inclure selon l'action demandée. Si l'action n'existe pas, elle renvoie le fichier par défaut "listeRestos.php".

connexion.php :

Ce contrôleur gère l'authentification de l'utilisateur.

Il affiche le formulaire de connexion si aucun identifiant n'a été envoyé (mailU ou mdpU manquant).

Si les identifiants sont fournis, il appelle la fonction login() pour tenter de connecter l'utilisateur.

Si la connexion réussit (isLoggedIn()), il affiche une page de confirmation.

Sinon, il réaffiche le formulaire de connexion avec un menu burger contenant les liens "Connexion" et "Inscription".

deconnexion.php :

Il appelle la fonction logout() pour déconnecter l'utilisateur, puis affiche à nouveau le formulaire de connexion.

listeRestos.php :

Ce contrôleur affiche la liste des restaurants.

Il récupère les restaurants via getLesRestos(), puis appelle les vues pour afficher la page avec un titre, l'en-tête, la liste des restos et le pied de page.

detailResto.php :

Ce contrôleur affiche les détails d'un restaurant sélectionné :

Il récupère l'ID du restaurant via \$_GET['idR'].

Il charge les infos du resto (détails, photos, critiques, types de cuisine, avis utilisateur).

Il construit un menu burger pour la navigation dans la page.

rechercheResto.php :

Ce contrôleur gère la **recherche de restaurants** selon un critère choisi par l'utilisateur (nom, adresse ou type de cuisine) .

Il détecte le critère passé en URL (\$_GET['critere']) et traite les données du formulaire (\$_POST).

En fonction du critère sélectionné, il appelle la bonne fonction pour récupérer les restaurants correspondants.

Il affiche ensuite le formulaire de recherche et, si des données ont été soumises, il affiche aussi les résultats.

Page index.php :

La page index sert à déterminer l'action à exécuter en fonction de l'URL et inclut le contrôleur correspondant pour gérer cette action. Elle charge également les fichiers nécessaires à l'authentification et au routage des actions.