

# Spécifications des fonctions du serveur

## Les fonctions du contrôleur

### **Controller**

`__construct()`

# définit le constructeur du contrôleur.

`action_default()`

# action par défaut du contrôleur (méthode à redéfinir par les classes filles).

`render(nomVue : String, data : tableau)`

# afficher la page HTML en fonction du nom de la vue nomVue et des données à transmettre à la vue data.

`action_error(mesg : String)`

# affiche une page d'erreur en fonction du message d'erreur mesg.

### **Controller\_affichage**

`action_default()`

# choisir l'action en fonction des besoins (conditions if)

`action_infoInventaire()`

# affiche les produits et leur quantité

`action_inventaireHistorique()`

# affiche l'historique des clients

`action_achatsHistoriqueClient()`

# affiche l'historique des achats du client

`action_infoClient()`

# affiche les informations du client

### **Controller\_inventaire**

action\_default()

# choisir l'action en fonction des besoins (conditions if) c'est-à-dire qu'il choisit soit action\_ajouterProduit, soit action\_supprimerProduit, soit action\_modifierProduit.

action\_ajouterProduit()

# ajouter les informations d'un nouveau produit dans la base de données

action\_supprimerProduit()

# supprimer les informations d'un produit existant dans la base de données

action\_modifierProduit()

# modifier les informations d'un produit existant dans la base de données

### **Controller\_connexion**

action\_default()

# choisir la méthode action\_verifieConnexion() par défaut

action\_verifieConnexion()

# vérifie si les informations rentrées lors du formulaire de connexion existent dans la base de données.

### **Controller\_inscription**

action\_default()

# choisir la méthode action\_ajoutInscription() par défaut

action\_ajoutInscription()

# ajoute les informations du formulaire d'inscription dans la base de données

### **Controller\_panier**

action\_default()

# choisir l'action en fonction des besoins (conditions if) c'est-à-dire qu'il choisit soit action\_ajouterArticlePanier, soit action\_supprimerArticleProduit

action\_ajouterArticlePanier()

```
# ajouter un produit à la table Panier

action supprimerArticlePanier
# supprimer un produit de la table Panier
```

## Les fonctions du modèle

### **Connexion**

```
enregistreUtilisateur()
```

```
# enregistre un utilisateur dans la table Compte de la base de données par rapport aux données saisies du formulaire d'inscription.
```

### **Inscription**

```
verifierUtilisateur()
```

```
#vérifie si un utilisateur existe dans la table Compte de la base de données par rapport aux données saisies lors du formulaire de connexion.
```

### **Catalogue**

```
recupererListeProduits()
```

```
# récupère l'ensemble des articles pour les présenter à l'utilisateur.
```

```
ajoutProduitPanier()
```

```
# ajoute le produit à la table Panier de la base de données.
```

### **Compte de fidélité**

```
recupererHistoriqueAchats(num : entier)
```

```
# récupère l'historique d'achat de l'étudiant en fonction de son numéro étudiant num.
```

```
recupereInformationsEtudiant()
```

```
# affiche le numéro étudiant, le mail étudiant et ses points de fidélité.
```

### **Inventaire**

```
recupereInventaire()
```

```
# récupère l'article et sa quantité correspondante.
```

getProduitPrix()

# récupérer le prix du produits

getProduitInventaire()

# récupérer un

recupereHistoriqueClient()

# affiche pour chaque client, la liste des achats (chaque articles séparés par des tirets), la date de l'achat et l'heure de l'achat

recuperePanier()

# récupère les articles ajoutés au panier dans la page Achat dans le but de les afficher.

affichePrixTotal()

# calculer le prix total et l'affiche.

ajouterCodePromo()

# ajouter un code promo à la base de données