|  |
| --- |
| **Webframework Module Utilisateur** |
| Document Technique |
|  |
| Version 2.0 |
|  |
| **AUGUEY Thomas** |
| **23/12/2012** |
|  |

Contenu

[Introduction 3](#_Toc344019661)

[Termes 4](#_Toc344019662)

[Environnement 5](#_Toc344019663)

[Webframework 5](#_Toc344019664)

[Base de données 5](#_Toc344019665)

[Environnement de développement intégré (IDE) 5](#_Toc344019666)

[Suivit de version 5](#_Toc344019667)

[Modèle de données 5](#_Toc344019668)

[Arborescence 6](#_Toc344019669)

[Modèle de données 6](#_Toc344019670)

[Entités 6](#_Toc344019671)

[USER 6](#_Toc344019672)

[SESSION 6](#_Toc344019673)

[CONNECTION 6](#_Toc344019674)

[IDENTITY 7](#_Toc344019675)

[ADDRESS 7](#_Toc344019676)

[Domaines 8](#_Toc344019677)

[STREET\_PREFIX 8](#_Toc344019678)

[SEX 8](#_Toc344019679)

[Formats 9](#_Toc344019680)

[Relations 10](#_Toc344019681)

[Contraintes 10](#_Toc344019682)

[Operations système 11](#_Toc344019683)

[Acteurs 11](#_Toc344019684)

[Visiteur 11](#_Toc344019685)

[Utilisateur 11](#_Toc344019686)

[Administrateur 11](#_Toc344019687)

[Système 11](#_Toc344019688)

[Résultats d’opération 11](#_Toc344019689)

[Détail des codes 11](#_Toc344019690)

[Cas d’utilisation 12](#_Toc344019691)

[Créer compte 12](#_Toc344019692)

[Connexion client 12](#_Toc344019693)

[Check Connexion 14](#_Toc344019694)

[Déconnexion d’un utilisateur 15](#_Toc344019695)

[Déconnexion de tous les utilisateurs 16](#_Toc344019696)

[Activer un compte utilisateur 17](#_Toc344019697)

[Contrats d’opération 17](#_Toc344019698)

[Create 17](#_Toc344019699)

[Connect 18](#_Toc344019700)

[Développement 19](#_Toc344019701)

[Point d’entrée de l’application 19](#_Toc344019702)

[Modèle MVC 19](#_Toc344019703)

[API Références 19](#_Toc344019704)

[Modèle objet et Classes Managers 19](#_Toc344019705)

# Introduction

Le projet Module Utilisateur permet de gérer dynamiquement des comptes utilisateur pour votre site web.

Fonctionnalités :

* Gestion des comptes utilisateurs
* Gestion des connexions par sessions et adresses IP
* Gestion des informations d’indentification (adresses, identité)

# Termes

Webframework

Le projet principal dont se module est une extesion.

# Environnement

## Webframework

Ce module est développé pour être utilisé avec le modèle minimal MVC du projet Webframework.

Votre application web doit au minimum implémenter l’interface IApplication et définir la variable globale $app pour fonctionner normalement (voir le modèle **/wfw/minimal** du projet **Webframework**).

## Base de données

Le **système de base de données** utilisé est de type **PostgreSQL** en adéquation avec le projet **Webframework**.

## Environnement de développement intégré (IDE)

Le module est développé sous l’environnement **NetBeans 7.2.1** en langage **PHP/Javascript/XHTML.**

## Suivit de version

L’Utilitaire **GIT** est utilisé pour le suivit et l’évolution du projet.

**Utile** : Le script archive.sh permet de créer une archive compressée du projet ne contenant que les fichiers importants. L’archive est stockée dans le dossier « **../archives »** et nommée automatiquement avec la version du projet en cours.

## Modèle de données

La conception de la **base de données** et du **modèle orienté objet** à été réalisé sur le principe de **l’UML** avec le logiciel **PowerAMC 15.1**.

**Important** :

Pour pouvoir générer correctement le code PHP du modèle orienté objet, veuillez copier le fichier « documents/php.xol » dans le répertoire d’installation de votre programme **PowerAMC** (ex : C:\Program Files (x86)\Sybase\PowerAMC 15\Fichiers de ressources\Langages objet\php.xol).

Ce fichier permet de générer le code **PHP** et les classes **managers** pour l’interaction avec la base de données.

# Arborescence

Voici une vue d’ensemble des dossiers du projet.

* DOCUMENTS : Documents divers relatif à ce projet
* MCD : Fichiers du Modèle conceptuel de données (PowerAMC)
* nbproject : Fichiers du projet **NetBeans**
* SQL : Fichiers SQL de création de la base de données
* wfw-X.X : Librairies et sources du projet à lier avec votre application

# Modèle de données

Le modèle des données persistantes est conçu sur le modèle Merise.

Le document « **mcd/user.mcd** » propose une version schématisé du modèle conceptuel de données, ce modèle est utilisé pour générer le script d’installation SQL. [Vous pouvez ouvrir ce fichier avec le logiciel **PowerAMC** de **Sybase**].

## Entités

Liste les tables ajoutées à la base de données.

Le tableau des entités est définit ainsi :

* Le **type** de donnée est écrit de façon générique (VARCHAR, CHAR, INTEGER, REAL, …)
* La **contrainte** est appliquée sur les champs par la base de données (Domaines, Clés, Check, …)
* Le **format**, définit plus bas, définit la syntaxe pour les chaines de caractères

### USER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribut | Description | Type | Contrainte | Format |
| USER\_ID | Identifiant | VARCHAR(64) | PRIMARY KEY | Identifier |
| USER\_PWD | Mot-de-passe | VARCHAR(64) | NOT NULL | Password |
| CLIENT\_ID | Identifiant du dossier client | VARCHAR(64) | NOT NULL | Name |
| USER\_MAIL | Adresse électronique de contact | VARCHAR(80) | NOT NULL | Mail |

### SESSION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribut | Description | Type | Contrainte | Format |
| SESSION\_ID | Identifiant | CHAR(64) | PRIMARY KEY | Name |
| LOCAL\_PATH | Emplacement de la ressource dans le système de fichier (PRIVE) | CHAR(260) | NOT NULL | UnixPath |

### CONNECTION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribut | Description | Type | Contrainte | Format |
| CLIENT\_IP | Adresse IP du client | VARCHAR(15) | PRIMARY KEY | IPv4 |
| LAST\_ACCESS | Date de dernier accès à la session | DATE | NOT NULL |  |
| LIFE\_TIME | Temps de vie avant déconnexion | INTEGER | NOT NULL |  |
| LINK\_PATH | Emplacement du lien symbolique dans le système de fichier (PUBLIC) | CHAR(260) | NOT NULL |  |

### IDENTITY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribut | Description | Type | Contrainte | Format |
| IDENTITY\_ID | Identifiant | INTEGER | PRIMARY KEY |  |
| FIRST\_NAME | Prénom | CHAR(80) | NOT NULL |  |
| LAST\_NAME | Nom | CHAR(80) | NOT NULL |  |
| BIRTH\_DAY | Date de naissance | DATE | NOT NULL |  |
| SEX | Sexe | DOMAIN(SEX) | NOT NULL |  |

### ADDRESS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribut | Description | Type | Contrainte | Format |
| ADDRESS\_ID | Identifiant | INTERGER | PRIMARY KEY |  |
| STREET\_NAME | Nom de rue | VARCHAR(128) | NOT NULL |  |
| STREET\_PREFIX | Préfix de rue | DOMAIN (STREET\_PREFIX) |  |  |
| STREET\_NUMBER | Numéro de rue | NUMBER(4,0) | NOT NULL |  |
| ZIP\_CODE | Code postal | VARCHAR (10) | NOT NULL |  |
| CITY\_NAME | Ville | VARCHAR (80) | NOT NULL |  |
| COUNTRY\_NAME | Pays | VARCHAR (80) | NOT NULL |  |
| BUILD\_NUMBER | Numéro de bâtiment | NUMBER(4,0) |  |  |
| APT\_NUMBER | Numéro appartement | NUMBER(4,0) |  |  |

## Domaines

### STREET\_PREFIX

Préfix d’une rue.

|  |  |
| --- | --- |
| Valeur : CHAR(3) | Description |
| BIS | Bis |
| TER | Ter |

### SEX

Sexe de la personne.

|  |  |
| --- | --- |
| Valeur : CHAR(1) | Description |
| M | Homme |
| F | Femme |

## Formats

Tous les formats utilisés hérites du projet **Webframework**, reportez-vous au document technique lié au projet pour plus d’informations.

## Relations

1. USER : CONNECTION
   1. Un **utilisateur** peut ouvrir plusieurs **connexions**
   2. Une **connexion** peut être ouverte par un **utilisateur**
2. CONNECTION : SESSION
   1. Une **session** peut être attaché à plusieurs **connexions**
   2. Une **connexion** peut être attachée à une seule **session**
3. USER : IDENTITY
   1. Un **utilisateur** peut renseigner une **identité**
   2. Une **identité** peut être renseignée par plusieurs **utilisateurs**
4. IDENTITY: ADDRESS
   1. Une **identité** peut posséder une **adress**e
   2. Une **adresse** peut être posséder par plusieurs **identités**

## Contraintes

1. Une **connexion** expire après un certain temps d’inactivité, dans ce cas, la connexion est automatiquement supprimée.
2. Pour être utilisé, un **compte utilisateur** doit d’abord être validé avec un code confidentiel envoyé par mail lors de sa création ou par demande du client.
3. Un utilisateur ne peut pas se connecter sous plusieurs adresses IP à la fois, cependant plusieurs utilisateurs peuvent utiliser la même adresse IP. (*Unicité de l’attribut* ***USER\_ID*** *dans la table* ***CONNECTION***).
4. Un lien symbolique est propre à une seule connexion (Unicité de l’attribut **LINK\_PATH** dans la table **CONNECTION).**
   1. Si un lien était partagé entre plusieurs connexions, une connexion fermée (supprimée) pourrait continuer à utiliser ce lien.

# Operations système

Les opérations systèmes sont conçues sur le modèle UML.

Le document « **mcd/operations.moo** » propose une version schématisé des cas d’opération. [Vous pouvez ouvrir ce fichier avec le logiciel **PowerAMC** de **Sybase**].

## Acteurs

### Visiteur

Utilisateur non enregistré.

### Utilisateur

Utilisateur enregistré, ayant créé un compte utilisateur.

### Administrateur

Utilisateur le l’interface de gestion.

### Système

Définition du système avec ses différentes couches de dépendances.

**Système Langage**

1. Application HTML/Javascript
2. Interface Serveur PHP
   1. Système embarqué (système de fichier) Console UNIX
3. Interface de Base de données PG/SQL PL/pgSQL
   1. Base de données SQL SQL

## Résultats d’opération

Les résultats d’opération sont basés sur le modèle du projet **WebframeWork**. Pour plus de détails à ce sujet reportez-vous au document technique concerné.

Noté que ses codes d’erreurs viennent s’ajouter aux codes déjà existant dans le projet **Webframework**. Certains d’entre eux peuvent être utilisé par les opérations présentes ci-dessous.

### Détail des codes

**Code Sous-Code Description**

ERR\_OK USER\_CREATED L’Utilisateur est créé

ERR\_OK USER\_DELETED L’Utilisateur été Supprimé

ERR\_FAILED USER\_EXISTS L’Utilisateur existe déjà

## Cas d’utilisation

### Créer compte

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC1 | | | Créer un compte utilisateur | |  |
| Intentions et objectifs | | | Ouvrir une connexion sur une session client (automatique) | | |
| Conditions | | | Les champs suivants sont fournis :   * Adresse eMail * Nom d’utilisateur * Mot-de-passe | | |
| Résultat | | | Une entrée USER est crée dans la base de données.  Un dossier client est créé dans le système de fichier. | | |
| Déclenchement | | | Le visiteur demande la création d’un compte utilisateur | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **CO** | **Action** | | |
|  | Visiteur | **S1** | Saisie des informations | | |
|  | Système (SYS) | **S2** | Crée le dossier de données dans le module client (E1) | | |
|  | Système (SGDB) | **S3** | Crée le compte utilisateur (E2) | | |
|  | Système (SYS) | **S4** | Crée les **liens d’accès** dans le système de fichier (E3) | | |
|  | Système (SYS) | **S5** | Crée **le lien de vérrou** dans le système de fichier (E3) | | |
|  | Système (SGBD) | **S6** | Envoie un mail d’activation (A1) | | |
|  | Système (SYS) | **S7** | Affiche un message de confirmation (A2) | | |
| Scénario alternatif | | |  | | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Système | **A1** | Le mail d’activation ne peut pas être envoyé.  [Appel du cas d’utilisation : Activer un compte utilisateur] | | |
|  | Système | **A2** | L’Utilisateur confirme l’activation du compte.  [Appel du cas d’utilisation : Activer un compte utilisateur] | | |
| Exceptions | | |  | | |
|  | **Code d’erreur** | **Etape** | **Action** | | |
|  | Client : [UC.Create] | **E1** | Le compte client ne peut pas être créé. | | |
|  | User : [CO.Create] | **E2** | Le compte utilisateur ne peut pas être créé. | | |
|  | SYS\_LINK\_FILE | **E3** | Le lien symbolique ne peut pas être créé. | | |

### Connexion client

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC1 | | | Connexion à une session utilisateur | |  |
| Intentions et objectifs | | | Ouvrir une connexion sur une session client (automatique) | | |
| Conditions | | | L’Adresse IP du client est connue.  Le temps de vie de la connexion est connu.  Le login et le mot-de-passe utilisateur est connue | | |
| Résultat | | | Une connexion est créée dans la base de données.  Une session ‘automatique’ est créée dans la base de données. | | |
| Déclenchement | | | Le visiteur demande une connexion à sa session utilisateur | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Visiteur | **S1** | Saisie du mot-de-passe et du nom d’utilisateur | | |
|  | Système (REQ) | **S2** | Vérifie le nom d’utilisateur et le mot-de-passe (A1, E1) | | |
|  | Système (REQ) | **S3** | Vérifie l’existence d’une **connexion** pour cet utilisateur (E2) | | |
|  | Système (REQ) | **S4** | Crée les **liens d’accès** dans le système de fichier (E3) | | |
|  | Système (REQ) | **S5** | Crée la **tache de fermeture** automatique de connexion à la **date d’expiration** (E4) | | |
|  | Système (SGBD) | **S6** | Vérifie l’existence de la **session** automatique liée à cet utilisateur | | |
|  | Système (SGBD) | **S7** | Crée une **connexion** avec l’adresse IP du client, l’identifiant de session, l’identifiant d’utilisateur et la date d’accès. | | |
|  | Système (REQ) | **S8** | Retourne l’identifiant de connexion à l’utilisateur | | |
| Scénario alternatif | | |  | | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Système | **A1** | Si le compte n’existe pas, informe l’utilisateur que le mot de passe et/ou le nom d’utilisateur est invalide.  [Retour à l’étape S1] | | |
|  | Système | **A2** | Si la session n’existe pas, la session est créée avec le nom de dossier client associé au compte utilisateur et le chemin d’accès. | | |
| Exceptions | | |  | | |
|  | **Code d’erreur** | **Etape** | **Action** | | |
|  | USER\_UNACTIVATED | **E1** | Le compte n’est pas actif. | | |
|  | USER\_ALERADY\_CONNECTED | **E2** | Une connexion existe déjà avec cette adresse IP. | | |
|  | SYS\_CREATE\_FILE | **E3** | Le lien symbolique ne peut pas être créé. | | |
|  | SYS\_CREATE\_TASK | **E4** | La tâche ne peut pas être créée. | | |

### Check Connexion

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC2 | | | Actualise l’état de la connexion | |  |
| Intentions et objectifs | | | Vérifier l’état de la connexion et renvoie le chemin d’accès aux fichiers. | | |
| Conditions | | | L’identifiant de connexion est connu  ou  Le login, le mot-de-passe et l’IP de l’utilisateur est connue | | |
| Résultat | | | La date d’expiration est actualisée (temps de vie + date en cours).  Le chemin d’accès au système de fichier est retourné. | | |
| Déclenchement | | | L’utilisateur actualise l’état de sa connexion | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Utilisateur | **P1** | Saisie l’identifiant de connexion | | |
|  | Système | **P2** | Vérifie l’existence de la connexion (E1, E2) | | |
|  | Système | **P3** | Actualise la date d’expiration (date system + durée de vie) | | |
|  | Système | **P4** | Actualise la tâche fermeture automatique (E3) | | |
|  |  |  |  | | |
|  |  |  |  | | |
|  |  |  |  | | |
| Scénario alternatif | | |  | | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Système | **A1** | Si la connexion n’existe pas, informe l’utilisateur que le mot de passe et/ou le nom d’utilisateur est invalide.  [Retour à l’étape P1] | | |
|  | Système | **A2** | Si la session n’existe pas, la session est créée avec le nom de dossier client associé au compte utilisateur et le chemin d’accès. | | |
| Exceptions | | |  | | |
|  | **Code d’erreur** | **Etape** | **Action** | | |
|  | USER\_CONNECTION\_NOT\_FOUND | **E1** | La connexion n’existe pas. | | |
|  | USER\_CONNECTION\_IP\_REFUSED | **E2** | L’Adresse IP ne correspond pas. | | |
|  | SYS\_UPDATE\_TASK | **E3** | La tâche ne peut pas être mise à jour. | | |

### Déconnexion d’un utilisateur

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC3 | | | Déconnexion d’un utilisateur | |  |
| Intentions et objectifs | | | Terminer une connexion entre un compte et une session | | |
| Conditions | | |  | | |
| Résultat | | | La connexion et supprimée de la base de données. | | |
| Déclenchement | | | L’Utilisateur demande une déconnection | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Utilisateur | **P1** | Saisie son mot de passe et son nom d’utilisateur | | |
|  | Système | **P2** | Vérifie l’exactitude des informations de connexion (A1) | | |
|  | Système | **P3** | Vérifie si la session est automatique (A2). | | |
|  | Système | **P4** | Supprime la session. | | |
|  | Système | **P5** | Retourne le code de résultat ERR\_OK. | | |
| Scénario alternatif | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  |  |  |  | | |
|  | Système | **A2** | [Retour à l’étape P5] | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  |  |  |  | | |
| Exceptions | | |  |  | |
|  |  | **Code** | **Action** | | |
|  |  |  |  | | |

### Déconnexion de tous les utilisateurs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC4 | | | Déconnexion de tous les utilisateur | |  |
| Intentions et objectifs | | | Terminer toutes les connexions utilisateur | | |
| Conditions | | |  | | |
| Résultat | | | Toutes **les connexions** sont supprimées de la base de données.  Toutes **les sessions** **automatiques** sont supprimées de la base de données.  Tous les liens de sessions automatiques sont supprimés du **système de fichiers**. | | |
| Déclenchement | | | L’Administrateur demande la fermeture de toutes les connexions | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Administrateur | **P1** | Confirme la fermeture des connexions | | |
|  | Système | **P2** | Supprime toutes les connexions | | |
|  | Système | **P3** | Supprime toutes les sessions automatiques | | |
|  | Système | **P5** | Retourne le code de résultat ERR\_OK. | | |
| Scénario alternatif | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  |  |  |  | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  |  |  |  | | |
| Exceptions | | |  |  | |
|  |  | **Code** | **Action** | | |
|  |  |  |  | | |

### Activer un compte utilisateur

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC5 | | | Activer un compte utilisateur | |  |
| Intentions et objectifs | | | Permet d’activer l’utilisation du compte client | | |
| Conditions | | |  | | |
| Résultat | | | Le compte est actif. | | |
| Déclenchement | | | L’Utilisateur demande activation de compte | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Utilisateur | **S1** | Saisie du code d’activation et du nom d’utilisateur (A1) | | |
|  | Système | **S2** | Vérifie l’exactitude des informations de connexion (A2) | | |
|  | Système | **S3** | Supprime le lien d’inactivation pour cet utilisateur dans le système de fichier. | | |
|  | Système | **S4** | Averti l’utilisateur que le compte est actif et prêt à être connecté. | | |
| Scénario alternatif | | |  | | |
|  | **Acteur** | **Etape** | **Action** | | |
|  | Système | **A1** | Si l’utilisateur ne connaît pas son code d’activation, le système lui renvoi par mail.  [Retour à l’étape S1] | | |
|  | Système | **A2** | Si l’utilisateur entre un code d’activation incorrecte, le système demande à nouveau la saisie des informations.  [Retour à l’étape S1] | | |
| Exceptions | | |  | | |
|  |  | **Code** | **Action** | | |
|  |  |  |  | | |

## Contrats d’opération

### Create

|  |  |  |  |
| --- | --- | --- | --- |
| Create | | Créé un utilisateur |  |
| Intentions et objectifs | | Crée une entrée utilisateur | |
| Système | | SGBD | |
| Résultat | | Une entrée utilisateur est créée dans la base de données. | |
| Algorithme | |  | |
|  | | Insertion dans la table USER | |
| Implémentation | |  | |
| PL/PGSQL | | Fichier : /sql/func.sql  Fonction : wfw\_user.create() | |
| Exceptions | |  | |
|  | USER\_EXISTS | L’Utilisateur existe déjà | |  |
| Entrées | |  | |
|  | USER\_ID | Identificateur (Nom d’utilisateur) | |
|  | USER\_PWD | Mot-de-Passe | |
|  | CLIENT\_ID | Identificateur (Dossier client) | |
|  | USER\_MAIL | Adresse Mail | |
| Retour | |  | |
|  | USER\_CREATED | L’Utilisateur est créé | |
|  | USER\_EXISTS | L’Utilisateur existe déjà. | |
|  | NOT\_SPECIFIED | Exception système non gérée. | |

### Connect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UC3 | | Déconnexion d’une session | |  |
| Intentions et objectifs | | Terminer une connexion entre un compte et une session | | |
| Conditions | |  | | |
| Résultat | | La connexion et supprimée de la base de données. | | |
| Déclenchement | | L’Utilisateur demande une déconnection | | |
| Algorithme | |  |  | |
|  | SI ( L’Utilisateur existe )  Mes a jour   * l’Id de session * le temps de vie de la connexion   SINON  Crée une connexion avec   * l’Id utilisateur * l’IP du client * l’Id de session * le temps de vie de la connexion   FINSI | | | |
| NOM OPERATION | | Connect |  | |
| CODES : | |  |  | |
| PL/PGSQL | | <mod\_user>/sql/func.sql |  | |

# Développement

## Point d’entrée de l’application

Conformément au projet **Webframework**, ce projet implémente l’interface IModule.

Vous trouverez le code de la classe UserModule dans le dossier « wfw-1.x/lib » avec l’ensemble des classes du modèle, vous devez inclure le fichier « module.php » à votre application pour pouvoir utiliser les fonctionnalités du module **Utilisateur**.

## Modèle MVC

Le projet est construit sur le modèle <**Modèle-Vue-Contrôleur**>.

* Les classes C++ du modèles sont stockées dans le dossier « wfw-1.x/lib **»**.
* Les classes de vues sont stockées dans le dossier « public/view »
* Les classes contrôleurs sont stockées à la racine du dossier « **/**public**».**

## API Références

La documentation du code source est disponible dans le dossier « doc », il contient une arborescence web (HTML) généré à partir du code source.

Les fichiers de projet **Doxygen** est situé dans le dossier «  doxygen ».

## Modèle objet et Classes Managers

Le modèle de données a son équivalent en classes objets sous **PHP**.