**©2012 ID-INFORMATIK**

2012

**Webframework ®**

**Librairies**

**Thomas AUGUEY**

Table des matières

[Introduction 2](#_Toc333581676)

[Cahier des charges 2](#_Toc333581677)

[Analyse 2](#_Toc333581678)

[DOM Compatible 2](#_Toc333581679)

[Format de champs 2](#_Toc333581680)

[Exemples 2](#_Toc333581681)

# Introduction

*WebFrameWork* est un kit de développement orienté web.

L’intérêt du Framework est de proposer une librairie de code portable et permettant de développer des sites web dynamique.

# Cahier des charges

Les systèmes et les navigateurs web modernes permettent de créé des sites-web comparable a de véritable application.

La gestion d’une tel application exige une vue d’ensemble

## Résultats d’opération

Tout système d’informations exécutant des opérations à besoin à un moment donné de connaitre, gérer ou enregistrer des états de résultat.

Par exemple pour :

* Une fonction dans un langage procédurale.
* Une requête serveur.
* Une procédure dans un langage SQL
* Etc…

Pour permettre la cohésion de l’information entre les différents systèmes, **Webframework** à définit quelques règles de structuration des résultats.

1. Un résultat doit au minimum définir :
   1. Un **Entier numérique** indiquant la **sévérité de l’erreur** : 0 pour un succès, 1 pour un échec, 2 pour une erreur et ainsi-de-suite suivant le contexte.
   2. Un **Identifiant** précisant le **nom de l’erreur.**
2. Et éventuellement :
   1. X champs **(Identifiant/Valeur)**, détaillant les informations du résultat.

# Modèle de données

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

Le document « **mcd/global.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

### RESULT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribut | Description | Type | Contrainte | Format |
| RESULT | Sévérité de l’erreur | INTEGER | Domaine (ERROR\_CODE) | - |
| INFO | Détaille l’erreur | VARCHAR(64) |  | Identifier |
| … | Toutes autres informations concernant le résultat de l’opération | VARCHAR |  | - |

## Format de données

Les formats de données sont basé sur un format d’expression régulière, ils permettent entre-autre de filtrer les champs de données avant leurs utilisations.

### IDENTIFIER

|  |  |
| --- | --- |
| Nom | Identifier |
| Description | Identificateur |
| Expression Régulière | [a-zA-Z\_]{1}[a-zA-Z0-9\_]\* |
| Exemple | my\_objectId |

### INTEGER

|  |  |
| --- | --- |
| Nom | Entier numérique |
| Description | Nombre positif compris entre 0 et *n* |
| Expression Régulière | 0|([1-9]{1}[0-9]\*) |
| Exemple | 145 |

### NAME

|  |  |
| --- | --- |
| Nom | Nom |
| Description | Un nom permet plus de liberté qu’un identifiant tout en excluant les caractères d’espacement ou caractères spéciaux. |
| Expression Régulière | [a-zA-Z\_]{1}[a-zA-Z0-9\_\-\.]\* |
| Exemple | Ceci.est-un-nom\_valide |

### MAIL

|  |  |
| --- | --- |
| Nom | Adresse électronique |
| Description | Adresse de courrier électronique |
| Expression Régulière | RFC-2822 |
| Exemple | hello@world.org |

### UNIXFileName

|  |  |
| --- | --- |
| Nom | Nom de fichier |
| Description | Nom de fichier au format UNIX |
| Expression Régulière | - |
| Exemple | dummy.dat |

### Password

|  |  |
| --- | --- |
| Nom | Mot-de-passe |
| Description | - |
| Expression Régulière | [a-zA-Z0-9\_\-\@\#\&\+\~]+ |
| Exemple | Mon#mot-De-passe\_50\_ |

### String

|  |  |
| --- | --- |
| Nom | Texte |
| Description | Texte compris entre crochets |
| Expression Régulière | [^"\n\r]\* |
| Exemple | Bienvenue à tous ! |

# 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

### 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 concernes les systèmes définit ci-dessus.

### Détail des codes

**Code Sous-Code Description**

ERR\_OK NO\_ERROR Réussite générique

ERR\_SYSTEM NOT\_SPECIFIED Exception système non gérée

ERR\_SYSTEM SYS\_LINK\_FILE Le lien symbolique ne peut pas être créé

## Cas d’utilisation

### Créer compte

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC1 | | | Fabriquer un template | |  |
| Intentions et objectifs | | | Fabrique un nouveau document depuis le couple (données/modèle) | | |
| Conditions | | | Les champs suivants sont fournis :   * Nom du fichier modèle (template) * Nom du fichier de données | | |
| Résultat | | | Un nouveau fichier est créé dans le système de fichier. | | |
| Scénario principal | | |  |  | |
|  | **Acteur** | **CO** | **Action** | | |
|  | Administrateur | **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) | | |
| 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] | | |
| 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. | | |

# Réalisation

## Applications Web (Front Office)

### Liste des Pages HTML

|  |  |  |
| --- | --- | --- |
| Fichier | Emplacement | Description |
| index.html | private/template/pages | Page d’accueil |
| default.xml | private/ | Schéma de l’application locale |
| default.xml | / | Schéma de l’application locale |

## Base de données SQL

### Liste des tables

|  |  |
| --- | --- |
| Nom | Usage |
| wfw.globals | Crée un utilisateur |
| wfw.dyn\_table | Crée une session |

### Liste des types

|  |  |
| --- | --- |
| Objet | Description |
| wfw.result | Objet de resultat de procédure |

### Liste des Fonctions

|  |  |
| --- | --- |
| Fonction | Description |
| public.bool\_to\_char | Convertie un boolean en caractère |
| wfw.set\_global | Définit une variable globale |
| wfw.get\_global | Obtient une variable globale |
| wfw.del\_global | Supprime une variable globale |
| wfw.create\_table | Crée une table dynamique |
| wfw.drop\_table | Supprime une table dynamique |
| wfw.add\_table\_column | Ajoute une entree à une table dynamique |

### Liste des Scripts

|  |  |  |  |
| --- | --- | --- | --- |
| Exécution | Fichier | Description | Rôle |
| 1 | create.sql | Crée le schéma | Intégration |
| 2 | tables.sql | Crée les tables et les contraintes | Intégration |
| 3 | function.sql | Crée les fonctions et objets | Intégration |
| 4 | populate.sql | Initialise les données de la base | Jeu de test |