# Documentation technique

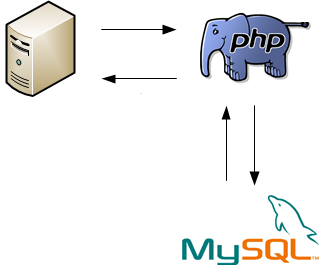
# Introduction

Ce document a pour but de décrire le fonctionnement technique et l’architecture de l’application IOS Giteo dans son état actuel. Cette application fait appel à une base de données distante contenant les informations de description des gites et des utilisateurs par l’intermédiaire d’un web service.

# Partie 1 : Le Web service

## Introduction

L’architecture de l’application est décrite dans le schéma suivant :

L’IPhone interroge le Webservice qui sera hébergé au même endroit que le site web Giteo en envoyant un fichier xml par l’intermédiaire d’une requête http. Le Webservice développé en PHP va ensuite analyser le fichier xml et faire des requêtes au près de la base de donnée MySQL. En fonction de la requête (Get, Post, Put ou Delete), le webservice renverra soit les informations demandées par le téléphone soit une confirmation de la demande d’insertion ou de suppression de champs dans la base de données en utilisant les mêmes protocoles qu’à l’envoi de données.

## Installation

Pour faire fonctionner le web service sur machine, il faut au préalable installer EasyPhp (<http://www.easyphp.org/>) avec Pear (<http://pear.php.net/>). Le dossier doit ensuite être mis dans l’arborescence du dossier défini en localhost.

## Framework

Le Framework utilisé pour le développement de l’application est jelix dans sa version 1.4.4 (<http://jelix.org/fr/>). On peut trouver la documentation de ce framework à l’adresse (<http://docs.jelix.org/fr/manuel-1.4/>). Ce framework pour PHP 5.2 comporte de nombreux composants, permettant entre autre d'architecturer une appli selon le pattern modèle-vue-controlleur (MVC) et d'accéder aux bases de données à travers un mapping objet relationnel ORM.

Ce framework imposant également une structure assez stricte dans l'organisation des fichiers d'une application permet ainsi d’améliorer la facilité dans la maintenance puisque cette organisation permet de mieux s'y retrouver.

Enfin cette architecture est conçue pour être modulaire. Les fichiers (contrôleurs, templates, classes métiers...) d'une application peuvent être regroupés par domaine fonctionnel, offrant une réutilisation forte du code de par son organisation.

## Fonctionnement des requêtes

Le code de traitement des requêtes se situe dans jelix/testapp/modules/testapp/controllers/soap.soap.php. La fonction est info qui prend en paramètre $req qui est une chaine. Cette chaine doit se décomposer de la manière suivante pour que la requête fonctionne : METHODE|nom\_de\_la\_table|nom\_du\_champ=valeur&autre\_nom\_de\_champ=valeur

Il y a 3 méthodes GET, PUT et DELETE (Le POST ne sera pas utilisé car il n’est pas nécessaire dans le cadre du fonctionnement de l’application)   
On peut mettre un « ? » lorsque l’on souhaite récupérer toutes les valeurs de la table.

Exemples d'adresse type:

GET|promotion|id\_promotion=634  
GET|promotion|id\_promotion=?  
PUT|district|id\_district=40&dst\_name=Corica&dst\_urlName=corica&dst\_neighbors=10,12  
DELETE|district|id\_district=40

Pour limiter le nombre d’envoi de fichier, il est possible d’effectuer plusieurs requêtes en un seul envoi en séparant les requêtes avec « || »

Exemple : GET|promotion|id\_promotion=634||PUT|district|id\_district=40&dst\_name=Corica&dst\_urlName=corica&dst\_neighbors=10,12

## Déploiement et maintenance

Le Webservice pour fonctionner devra être mis en ligne et hébergé chez OVH avec le site web actuel. Il sera connecté sur la base de données en fonctionnement du site en utilisant le protocole HTTPS avec une clé de chiffrement SSL. Il ne sera pas accessible depuis un navigateur web pour éviter tout piratage de la base de données et sera accessible uniquement depuis l’application IOS giteo.

Le code a été optimisé pour limiter la maintenance en cas de modification de la base de donnée. Si des tables sont modifiées ou ajoutés, il faudra refaire les daos jelix et modifier une partie du code de l’application IOS si l’on veut implémenter de nouvelles fonctionnalités.

## Conventions de codage

Les conventions de codage utilisées sont celles recommandées par PHP. Elles sont disponibles à cette adresse : [http://pear.php.net/manual/en/standards.php](http://pear.php.net/manual/en/standards.php" \t "_blank). La structure du code et son organisation ont été optimisés pour le rendre le plus clair possible.

## Documentation HTML générée

Pour lire la documentation générée en HTML, il suffit de lancer la page index.html dans le dossier PHPdoc.

# Partie 2 : Xcode

## Introduction

**Xcode** est un environnement de développement pour [Mac OS X](http://fr.wikipedia.org/wiki/Mac_OS_X) ainsi que [iOS](http://fr.wikipedia.org/wiki/IOS_(Apple)). Fourni avec toute une suite logicielle (graphiques, audio, etc.) pour développeurs et programmeurs, il permet de créer des logiciels utilisant toutes les fonctionnalités, la puissance et la stabilité de [Mac OS X](http://fr.wikipedia.org/wiki/Mac_OS_X) et d'[UNIX](http://fr.wikipedia.org/wiki/UNIX).



## Installation

Cet environnement peut être obtenu gratuitement sur le [Mac App Store](http://fr.wikipedia.org/wiki/Mac_App_Store). Il était fourni en standard avec chaque Mac, sur les disques d'installation de Mac OS X 10.6, Mac OS X 10.5, 10.4 et 10.3. Il n'est pas pré-installé sur l'ordinateur et doit être installé séparément. Nous avons utilisé la version 3.6 de XCode sur un Mac Snow Leopard (10.6.8).

## Logiciel de programmation

Le logiciel de programmation utilisé est l’Objective-C. Ce logiciel est principalement utilisé dans les systèmes d'exploitation d'[Apple](http://fr.wikipedia.org/wiki/Apple,_Inc.) : [Mac OS X](http://fr.wikipedia.org/wiki/Mac_OS_X) et son dérivé [iOS](http://fr.wikipedia.org/wiki/IOS_(Apple)), basés sur la bibliothèque de classes [Cocoa](http://fr.wikipedia.org/wiki/Cocoa_(Apple)) mais il existe aussi une [bibliothèque de classes](http://fr.wikipedia.org/wiki/Interface_de_programmation) libre [GNUstep](http://fr.wikipedia.org/wiki/GNUstep) sous [GNU/Linux](http://fr.wikipedia.org/wiki/GNU/Linux).

## Déploiement

Une fois le développement de l’application terminée, nous devons la mettre sur l’App Store. Cela peut prendre plus de deux semaines, les développeurs d’Apple vérifient que le code est conforme et aux normes d’Apple.

## Maintenance

A chaque modification de la structure de la base de données, l’application devra être modifiée pour qu’elle puisse redevenir fonctionnelle si les modifications concernent la structure des tables existantes, en revanche si des tables sont ajoutées, l’application sera toujours fonctionnelle. Si l’on veut intégrer de nouvelles options, il faudra modifier l’application et la mettre à jour sur l’Apple Store.

## Conventions de codage utilisées

Le développement de l’application a respecté les conventions de codage d’Apple. (<http://developer.apple.com/library/ios/#documentation/cocoa/conceptual/ProgrammingWithObjectiveC/Conventions/Conventions.html>)

# Conclusion

Cette application une fois mise en place permettra aux utilisateurs de réserver en toute simplicité une chambre d’hôte ou un gite depuis leur Iphone. Elle nécessite cependant encore des modifications avant sa mise en service.