

Nicolas BOUGET
✉ nicolas.bouget@esial.net
3A TRS1

Marc PINHÈDE
✉ marc.pinhede@esial.net
3A LE

Julien GUÉPIN
✉ julien.guepin@esial.net
3A IL

Julien VAUBOURG
✉ julien@vaubourg.com
3A TRS2

Interface web d'administration

Documentation

Projet industriel
encadré par Guillaume ROCHE
pour la société B.H. Consulting

Le 15 octobre 2012



Table des matières

1	Le framework CakePHP	3
1.1	Solutions de frameworks PHP	3
1.2	CakePHP	3
2	Le framework Bootstrap	4
3	Intégration de Bootstrap dans CakePHP	4
4	Le SGBD MySQL	5

Ce document détaille les choix techniques qui ont été pris pour construire une interface web simple et intuitive de gestion de l'authentification Radius et de sauvegarde des configurations.

1 Le framework CakePHP

L'interface web doit être développée dans le langage **PHP**. Nous avons décidé d'utiliser un framework afin d'accélérer le développement et d'obtenir une base solide et sécurisée pour l'application.

1.1 Solutions de frameworks PHP

Après une analyse du marché des frameworks PHP et une comparaison des solutions, il est sorti une liste des produits populaires et puissants :

- CakePHP
- Symfony
- Yii
- CodeIgniter
- Zen Framework

Nous avons sélectionné **CakePHP** pour diverses raisons. Il s'agit d'un des premiers framework qui ait été développé en PHP, en suivant le modèle de Ruby on Rails. Le logiciel dispose donc aujourd'hui d'une grosse communauté de développeurs, et la documentation est complète et détaillée. La prise en main du logiciel est rapide, tout en laissant une liberté d'adaptation pour tous les usages. CakePHP n'offre pas toutes les fonctionnalités avancées de *Symfony* ou de *Zend Framework*, qui permettent l'interaction avec d'autres services web (comme Google), mais les fonctionnalités disponibles suffisent largement au projet. Les performances du produit sont également très bonnes. *Yii* permet de construire des applications qui tiennent mieux la charge, même avec des milliers d'utilisateurs, mais l'interface de configuration d'un réseau n'est pas destinée à accueillir une telle charge.

De plus, la société **b.h. consulting** utilise déjà ce framework dans certains de ses produits.

1.2 CakePHP

CakePHP est un framework PHP permettant de développer des applications web. Il est disponible sous *licence libre MIT*. Le projet a démarré en 2005, afin d'imiter le fonctionnement de *Ruby on Rails*. La communauté de développeurs est grande, et la documentation est détaillée. Le document de référence est le **Cookbook**.

Le patron de conception **MVC** (Modèle-Vue-Contrôleur) est respecté dans CakePHP, afin de séparer les données (le modèle), les contrôles de l'utilisateur (les contrôleurs), et les interfaces (les vues). CakePHP utilise également le patron **Active Record** pour faciliter la connexion à la base de données, directement liée au modèle. Le schéma des tables représente les attributs de classes, et les méthodes du modèle intègrent les opérations **CRUD** (create, read, update, delete).

De nombreuses autres fonctionnalités sont disponibles, telles qu'un dispatcheur d'URL, la validation des données, le moteur de templates facilitant le formatage (utilisation de AJAX, HTML, Javascript, ...), gestion du cache, composants de sécurité et scripts permettant la génération de code.

2 Le framework Bootstrap

Afin d'obtenir une interface web conviviale, nous utiliserons le framework CSS **Bootstrap**. Il s'agit d'un projet développé par Twitter qui facilite la création d'interfaces web : une feuille de style CSS basique est proposée, et il est possible d'utiliser de nombreux éléments d'interface, comme les boutons, les formulaires, ... La mise en page avec barre de navigation, menu, colonnes et pied de page est simplifiée. Il est également possible d'intégrer des extensions Javascript qui ajoutent des effets de transition et facilitent le développement Javascript, à la manière de jQuery.

De plus, Bootstrap utilise le "**responsive design**" pour supporter tous les supports d'accès au web : ordinateurs, téléphones mobiles, tablettes... Tous les éléments de l'interface s'adaptent en fonction de la taille de l'écran ou de la fenêtre du navigateur. Ainsi, toutes les fonctionnalités du projet seront accessibles, quelle que soit la plateforme de l'utilisateur.

3 Intégration de Bootstrap dans CakePHP

Par défaut, CakePHP propose un style pour les vues qu'il génère, mais nous utilisons Bootstrap pour ce projet. Il existe alors deux solutions pour intégrer Bootstrap aux vues de CakePHP : créer des "helpers" CakePHP (éléments de vue) qui utilisent un style Bootstrap, ou modifier les styles de Bootstrap pour utiliser les mêmes noms que ceux générés par CakePHP. Nous utilisons cette deuxième solution afin de rester compatible CakePHP, et non pas spécialisée Bootstrap. Ainsi, il sera toujours possible par la suite de changer de style ou de framework CSS sans modifier le code PHP.

Nous utilisons pour cela le projet [Cakephp-Bootstrappifier](#), qui ajoute un plugin Javascript à CakePHP pour réécrire tous les styles générés par CakePHP en style compatible Bootstrap.

4 Le SGBD MySQL

Le serveur **FreeRadius** permet de stocker ses données de plusieurs manières, notamment dans une base de données. Nous avons choisi d'utiliser le système de gestion de bases de données **MySQL**, puisqu'il est performant, sous licence libre, populaire et répandu. FreeRadius ira chercher toute la configuration dont il a besoin dans la base. L'interface web permettant de configurer le serveur Radius devra donc modifier la base de données MySQL. La connexion à une base MySQL est très simple en PHP, d'autant plus en utilisant le framework CakePHP. Celui-ci adopte les design patterns Active Record et MVC. Le modèle représente les données de l'application, qui sont ici stockées dans la base de données. Active Record permet de lier directement le schéma de la base aux classes du modèle. Ainsi, on pourra appeler des méthodes génériques sur les classes du modèle et modifier simplement le contenu de la base.

Le schéma de la base de données utilisée par FreeRadius est le suivant :