

CARBONNIER Nicolas
DUT Informatique – 2ème année

IUT de Saint-Dié, Université de Lorraine
11 Rue de l'Université
88100 Saint-Dié-des-Vosges



Rapport de stage

Développeur Web

03 Avril – 08 Juin 2018



SSR Disponibilité

Réalisation d'un module Xoops indiquant la disponibilité
des Spécialistes Systèmes et Réseaux

Tuteur : Philippe DIDIOT



**Office National des Forêts – Direction
Territoriale Grand-Est**
Agence Territoriale de Bar-le-Duc
60 Boulevard Raymond Poincaré
55000 Bar-le-Duc

Remerciements

Je tiens tout d'abord à remercier mon maître de stage, M. Philippe DIDOT, Spécialiste Systèmes et Réseaux au sein de l'ONF Grand-Est, qui m'a été de bon conseil durant la phase de réflexion et lors de l'amélioration du module, ainsi que durant la rédaction de ce rapport.

Je remercie également les membres de l'équipe informatique de la Direction Territoriale Grand-Est, pour m'avoir aidé, pour avoir donné leur avis sur ma production et pour m'avoir fait confiance pour mener à bien le développement et la mise en production du module.

Enfin, je tiens à remercier l'agence de Bar-le-Duc pour m'avoir accueilli dans ses locaux.

Sommaire

Remerciements.....	2
Sommaire.....	3
Résumé en Anglais.....	4
I. Introduction.....	5
II. Présentation de l'entreprise.....	6
A. Fiche de Synthèse.....	6
B. Présentation détaillée.....	7
1. L'organisation de l'Office National des Forêts.....	7
2. Les missions de l'ONF.....	9
3. L'informatique à l'ONF.....	10
III. Cahier des charges.....	11
A. Présentation.....	11
B. Analyse.....	12
IV. Méthodologie.....	13
V. Mise en œuvre.....	15
A. Début de la réalisation du module.....	15
B. La page d'administration.....	17
C. Le bloc du module.....	20
D. Téléchargement des calendriers et mise à jour de la table event.....	24
E. Classement des événements.....	26
F. Mise en production.....	28
VI. Conclusion.....	29
VII. Après-texte.....	30
A. Annexes.....	30
B. Glossaire.....	58
C. Bibliographie.....	58

Résumé en Anglais

In order to close my training in DUT Computer Science, I was welcomed at the Office National des Forêts, more precisely in the Grand-Est Direction, at the Bar-le-Duc agency.

The IT team assigned me the task of creating a module for the content management system Xoops. This module joins the CMS project, a project designed to facilitate exchange between staff members. The module must allow users to know the availability of the members of the IT team at the time of the consultation of the intranet site.

The module that I developed consists of two elements : a block that displays the list of SSRs, and an administration page for adding, modifying and deleting an SSR.

The block therefore displays the list of SSRs, with a color depending on their availability. If he is busy at the time of the consultation, the event in which the SSR participates is indicated under the name, as well as the location of the latter. Clicking on the name of one of the SSRs opens his online calendar, and hovering over it displays information such as the workplace, phone number and email address. SSR calendars are recovered through Outlook share links.

The administration page has a form to add an SSR and a table showing the list of those already added. To add an SSR, all you need is his ID and his two Outlook share links, other data such as name, first name or phone are retrieved from the LDAP server.

This internship allowed me to exploit my knowledge in PHP and JavaScript, and to acquire new methods of organization. It also allowed me to discover a professional life very different from what I expected. Indeed, I discovered a mood and a pace of work much more relaxed than I imagined.

I. Introduction

Dans la cadre de ma formation en DUT Informatique, et afin de clore cette dernière, j'ai été accueilli à l'Office National des Forêts, plus précisément dans la Direction Territoriale Grand-Est, à l'agence de Bar-le-Duc.

L'équipe informatique de la région Grand-Est m'a confié la tâche de réaliser un module pour le système de gestion de contenu Xoops. Ce dernier entre dans le projet CMS, un projet destiné à faciliter l'échange entre les membres du personnel. Le module dont on m'a confié la réalisation doit permettre aux utilisateurs de connaître la disponibilité des membres de l'équipe informatique au moment de la consultation. Celui répond donc à un réel besoin, car il permet aux utilisateurs d'économiser beaucoup de temps. En effet, ils n'auront plus à chercher un des Spécialistes Systèmes et Réseaux lorsque ce dernier n'est pas dans son bureau, ou ne répond pas au téléphone.

Dans un premier temps, nous allons découvrir plus en détail l'entreprise dans laquelle j'ai réalisé mon stage, avant de poursuivre avec la présentation et l'analyse du cahier des charges. Ensuite, nous aborderons la méthodologie, avec notamment la liste des livrables et le calendrier prévisionnel, puis continuerons avec la mise en œuvre du projet. Enfin, nous clôturerons ce rapport avec la conclusion.

II. Présentation de l'entreprise

A. Fiche de Synthèse



Logo de l'Office National des Forêts

Raison sociale : Office National des Forêts

Activités : Gestion des forêts publiques, Sylviculture, Police de l'environnement, Défense des forêts contre les incendies, Restauration des terrains en montagne, etc

Siège social : 2 Avenue de Saint-Mandé 75012 Paris, Île-de-France

Forme juridique : établissement public à caractère industriel et commercial

Effectif en 2017 : 9 101

Chiffre d'affaire en 2017 : 847 300 000 €

Directeur Général : Christian Dubreuil

Président du Conseil d'administration : Jean-Yves Caullet

Téléphone : 01 40 19 58 00

SIRET : 662 043 116 00018

NAF : 0240Z

B. Présentation détaillée

L'Office National des Forêts est un établissement public à caractère industriel et commercial français, placé sous la tutelle ministère de l'Agriculture, de l'Agroalimentaire et de la Forêt et du ministère de l'Écologie, du Développement durable et de l'Énergie. Créée en 1964 et succédant à l'Administration des Eaux et Forêts, l'entreprise est chargée de gérer les 11 millions d'hectares de forêts publiques françaises, métropole et outre-mer confondus.

1. L'organisation de l'Office National des Forêts



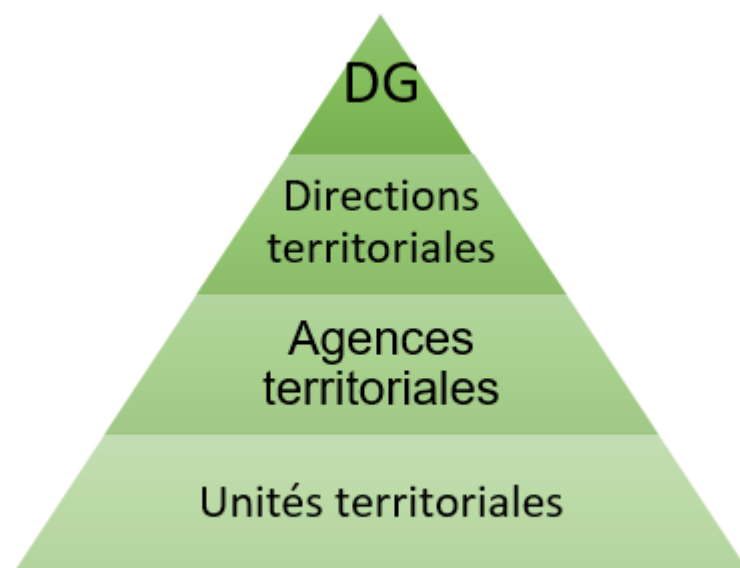
Carte de l'organisation de l'ONF (Janvier 2017)

L'ONF est organisé en 320 unités territoriales et 51 agences territoriales regroupés en 6 Directions Territoriales (Auvergne-Rhône-Alpes, Bourgogne-Franche-

Comté, Centre-Ouest-Aquitaine, Grand-Est, Midi-Méditerranée, Seine-Nord) et 5 Directions Régionales (Corse, La Réunion, Guyane, Guadeloupe, Martinique).

Chaque Direction Territoriale est divisée en agences, qui sont constituées d'un service de direction et de plusieurs unités territoriales. Ces unités territoriales sont composées d'une équipe d'agents forestiers affectés à la gestion d'une surface forestière (les triages*).

Chaque entité citée juste avant dispose d'un périmètre de décision et en réfère à une entité supérieure. Ainsi, la Direction Générale, et les Directions Territoriales et Régionales sont des entités de direction. Les agences territoriales sont des entités de gestion et les unités territoriales sont des entités de production.



Organigramme hiérarchique de l'ONF

La Direction Territoriale Grand-Est est séparée en deux pôles : le pôle Ouest (Aube, Ardennes, Marne, Haute-Marne, Meurthe-et-Moselle, Meuse, Moselle nord, Vosges ouest) et le pôle Est (Moselle sud, Bas-Rhin, Haut-Rhin, Vosges est). Cette séparation est justifiée par l'importance de cette région forestière, autant en personnels qu'en surface de forêts gérées (25 % du personnel national, 30 % de la production nationale en bois).

2. Les missions de l'ONF

L'établissement intervient donc en France métropolitaine et dans les départements d'outre-mer, où il assure plusieurs missions :

- La gestion des forêts publiques : l'ONF est un acteur majeur de la filière forêt-bois. Les forestiers veillent à l'entretien, au développement et au renouvellement des forêts publiques.
- La sylviculture : l'entreprise doit répondre aux besoins de l'homme grâce à la production et à la récolte du bois. Les forestiers vont donc effectuer plusieurs tâches comme la protection des jeunes plants contre le gibier ou le nettoyage de la végétation concurrente (buis, ronce, etc).
- L'aménagement : l'établissement réalise les aménagements nécessaires à l'accueil du public (parkings, signalétiques, etc) et s'occupe aussi des espaces dédiés aux sports en plein air (sports équestres, VTT, ski, etc) et des aménagements adaptés aux handicapés.
- La protection de l'environnement : afin d'évaluer l'impact environnemental et de préserver la faune et la flore, l'ONF peut effectuer des études d'impact lors de projets comme l'aménagement d'une ZAC ou la construction d'une route. Il peut aussi intervenir dans des travaux de restauration de milieux artificialisés ou dégradés.

L'ONF réalise également des missions de service public pour le compte de l'État dans les domaines de la prévention et de la gestion des risques naturels :

- Protection du littoral : réhabilitation et stabilisation des dunes, installation de rideaux brise-vent, etc.
- Restauration des terrains en montagne : reboisement, protection contre les avalanches et les éboulements, etc.
- Défense des forêts contre les incendies : installation de points d'eau, travaux de débroussaillage, etc.

Afin de mieux comprendre le travail d'un agent de terrain, j'ai passé une journée en forêt avec Vincent LEBRUN, technicien forestier territorial sur l'agence de Bar-le-Duc. Durant cette journée, nous avons effectué du cubage* de résineux*, visiter une exploitation et assister à un chantier d'aménagement d'un accès handicapé à une aire de pique-nique. Cette journée fut riche en échange, et j'ai pu découvrir plus en détail l'activité de M. LEBRUN.

L'ONF est engagé dans le système de certification PEFC pour la gestion durable des forêts publiques. PEFC est un système de certification forestière ayant pour objectif de promouvoir la gestion durable des forêts afin de répondre aux besoins de l'homme tout en assurant la prospérité des forêts et d'apporter une garantie environnementale, sociale et économique aux acheteurs et aux transformateurs sur l'origine du bois. L'entreprise est également membre de l'association PEFC France et participe ainsi à la gouvernance du système et à la définition du schéma de certification.



Logo de la certification PEFC

3. L'informatique à l'ONF

L'infrastructure informatique de l'ONF est essentiellement basée sur l'environnement Windows (Windows 8 et 10, Windows Server 2012 R2, etc) et sur les outils de Microsoft (Outlook, Skype Entreprise, Active Directory*, etc). La majorité de l'infrastructure de l'entreprise est située dans un datacenter* à Toulouse et c'est dans ce dernier que sont regroupés les principaux serveurs de l'ONF (SCCM*, Active Directory, DHCP*, DNS*, etc).

Le personnel informatique de l'ONF Grand-Est est composé de deux Responsables Informatique Territoriaux, Michel SCHUTZ pour le pôle Ouest, et François BENAOLI pour le pôle Est, ainsi que de 14 Spécialistes Systèmes et Réseaux.

Les Spécialistes Systèmes et Réseaux (SSR) sont chargés de gérer le réseau informatique de l'ONF, d'assurer la maintenance du matériel, ainsi que la configuration du matériel et des logiciels. Ils ont également pour mission de former et d'assister les autres employés.

III. Cahier des charges

A. Présentation

Les membres du personnel de l'ONF disposent d'un SSR référent qu'ils contactent en cas de problème ou de besoin. Chaque SSR s'occupant d'un périmètre géographique assez vaste, il arrive parfois qu'il ne soit pas disponible.

Pour éviter aux membres du personnel de chercher leur SSR référent, alors que ce dernier est, par exemple, en déplacement, en réunion ou en congé, il a été décidé de réaliser un module Xoops chargé d'indiquer la disponibilité de chaque SSR de la Direction Territoriale Grand-Est.

Ce module est intégré au projet CMS, un projet ayant pour objectif de faciliter l'échange entre les membres du personnel, en prenant la forme d'un site intranet utilisant le système de gestion de contenu Xoops. Le site possède notamment un module de rédaction et de mise en ligne d'article (AMS), permettant la mise à disposition des guides, chartes et autres documents utiles aux utilisateurs. Il possède également un module de création de sondages et de formulaires (Liaise), déjà utilisé pour deux formulaires : un premier pour l'ouverture d'incident auprès du service informatique et un second pour la gestion d'un nouvel arrivant.

Le module doit récupérer les calendriers Outlook des SSR de manière autonome, pour ensuite analyser et classer les événements en 3 catégories :

- Indisponible : les congés, RTT ou réunions
- Occupé : les autres événements
- Disponible : aucun événement (ou les événements laissant disponible, comme l'accueil d'un stagiaire)

Pour l'utilisateur final (les membres du personnel), le module devra afficher la liste des SSR ainsi que leur disponibilité sur la page d'accueil (ou une autre page). Le nom des SSR intégrera le lien hypertexte vers leur agenda en ligne.

Pour les webmestres (les SSR), le module devra permettre d'ajouter, de modifier et de supprimer un SSR de la liste en cas de départ ou de nouvel arrivant.

B. Analyse

Les contraintes posées par le CMS Xoops imposent plusieurs choses. Tout d'abord, les langages utilisés seront des langages Web comme le PHP, le JavaScript et l'HTML. Pour l'affichage de la liste des SSR, la création du module n'offre que deux solutions, la première est d'avoir une page dédiée au module et la seconde est de mettre en place un bloc pouvant être placé où on souhaite dans la page ainsi que sur plusieurs pages. L'utilisation d'un bloc, offrant plus de modularité, sera donc la solution retenue.

Pour récupérer les calendriers Outlook, la première solution envisagée était de les récupérer via l'API REST* de Outlook, la seconde était de passer par le lien d'abonnement fourni lorsque l'on publie son calendrier Outlook, qui fournit un fichier au format .ics facilement exportable en tableau PHP et dans une base de donnée.

Concernant le traitement des événements, la seule solution possible est de filtrer des mots-clés dans le nom des événements. Les congés et les RTT étant posés via Tempus* pour les SSR fonctionnaires, les événements rendant un SSR indisponible peuvent donc être filtrés, en partie, avec le mot-clé "Tempus". D'autres mots-clés seront également utilisés, et le module intégrera des options pour ajouter d'autres mots-clés si nécessaire.

Pour pouvoir consulter les calendriers en ligne, la seule solution est d'utiliser le lien de consultation fourni lorsque l'on publie son calendrier Outlook.

La solution retenue pour récupérer les calendriers Outlook est donc l'utilisation de lien d'abonnement fourni par la publication des calendriers. Ce choix a semblé être le plus judicieux car le lien d'abonnement est dans tous les cas fourni car pour consulter les calendriers en ligne, il était obligatoire de publier les calendriers Outlook. Cette solution a semblé également être la plus simple à mettre en place.

IV. Méthodologie

Liste des livrables :

- Le module Xoops
- La documentation du module, expliquant comment l'utiliser et donnant des règles à suivre lors de l'ajout d'événement dans les calendriers ([voir Annexes](#))

Ressources :

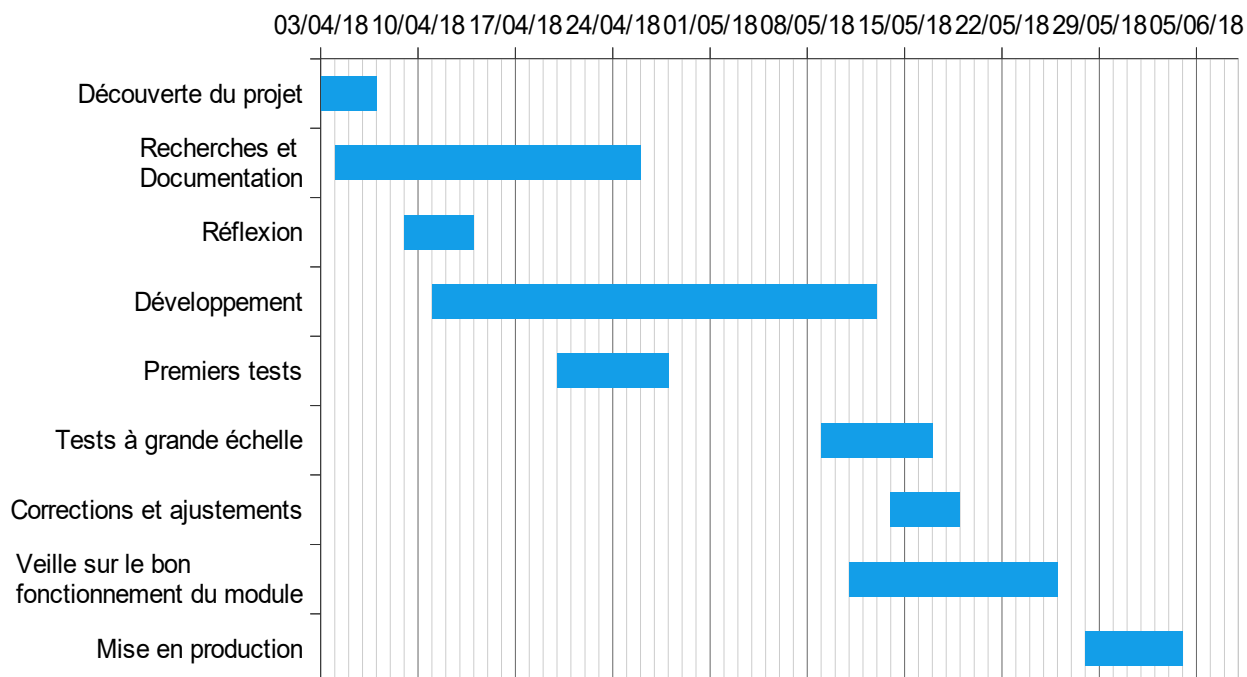
- Humaines : l'équipe informatique (notamment mon maître de stage, ainsi que Michel SCHUTZ, responsable de l'équipe informatique territoriale, et Nicolas GRIMMER, SSR à Sarrebourg et membre du groupe de projet CMS)
- Matérielles : installation dans le bureau de mon maître de stage, mise à disposition d'un ordinateur portable, d'une station d'accueil, d'un second écran, des périphériques (micro-casque, clavier, souris) et d'un serveur virtuel de test

Logiciels utilisés :

- Visual Studio Code
- Mozilla Firefox, Internet Explorer
- Outil de connexion Bureau à distance
- Skype Entreprise, Outlook
- Suite LibreOffice, Suite Microsoft Office

Budget prévisionnel :

- Ordinateur portable (prêt) : 700 € HT
- Station d'accueil pour l'ordinateur (prêt) : 180 € HT
- Écran (prêt) : 300 € HT
- Périphériques (prêt) : 40 €
- Frais de réunion (repas et déplacement) : 400 €

Planning prévisionnel :

V. Mise en œuvre

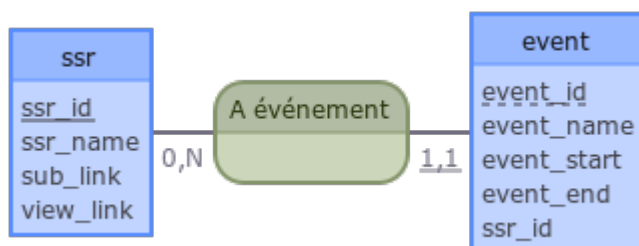
Pour réaliser ce projet, j'ai commencé par me documenter sur Xoops, un système de gestion de contenu conçu avec une approche orientée objet, ainsi que sur la création d'un module.

Avant de commencer à coder, j'ai installé XAMPP* et Xoops sur un serveur virtuel de test mis à disposition par mon maître de stage. J'ai également réfléchi à la structure des tables nécessaires au module et testé la publication du calendrier Outlook avec le compte LDAP fourni par l'ONF pour mon stage.

A. Début de la réalisation du module

Le début de la réalisation a été fait en parallèle du suivi d'un tutoriel pour créer un module Xoops.

La première étape a été de créer les fichiers qui lui sont indispensables (xoops_version.php, index.php) ainsi que le fichier .sql permettant de construire les tables dans la base de données.



Tables du module au début du projet

J'ai ensuite créé le formulaire sur la page dédiée au module pour ajouter des SSR à la table ssr. La création du formulaire fut faite deux fois : la première via des fonctions propres à Xoops pour développer la partie fonctionnelle, et la seconde en utilisant un template Smarty pour travailler

l'apparence et adopter une structure MVC (Modèle, Vue, Contrôleur).

Je me suis donc documenté sur Smarty, qui est un moteur de template PHP permettant de séparer la structure HTML du code. Le moteur se charge de compiler les deux fichiers en un seul script PHP, puis de l'exécuter. Ce système est très utilisé par Xoops, notamment pour les modules et les gabarits des thèmes.

Après avoir vérifié que les informations saisies dans le formulaire étaient correctement enregistrées dans la base de données, j'ai mis en place l'affichage des SSR enregistrés dans la table, sous la forme d'un tableau. J'ai passé les enregistrements dans une variable Smarty, et utilisé une boucle pour créer une ligne dans le tableau pour chaque enregistrement passé dans la variable. J'ai ensuite ajouté des liens permettant la suppression ou la modification des enregistrements.

[illegible]

Formulaire et tableau affichés dans la page dédiée au module

Les actions ont été récupérées via la variable superglobale `$_GET`, et j'ai ajouté un second formulaire dans le template, qui s'affiche uniquement lorsque l'on choisit de modifier un enregistrement existant.

B. La page d'administration

La partie gestion des SSR étant réservée aux administrateurs du site, j'ai migré le formulaire et le tableau dans la section d'administration, en faisant les modifications adéquates dans le fichier `xoops_version.php` :

- Définir `hasAdmin` à 1, afin de dire que le module dispose d'une page d'administration
- Définir les chemins vers les fichiers `index.php` et `menu.php` de la partie administration
- Définir `hasMain` à 0, car le module n'a plus besoin d'une page dédiée

Ajouter un SSR

Nom du SSR

Lien d'abonnement

Lien de consultation

Nom du SSR	Lien d'abonnement	Lien de consultation	Actions
GRIMMER Nicolas	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.ics	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.html	Modifier Supprimer
DIDIOT Philippe	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.ics	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.html	Modifier Supprimer
GAUTHEROT Philippe	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.ics	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.html	Modifier Supprimer
ABRAHAM Cyril	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.ics	http://xoops.cool.fr/ressources/calendar/255749489150/calendar.html	Modifier Supprimer

Formulaire et tableau après la migration vers la page d'administration

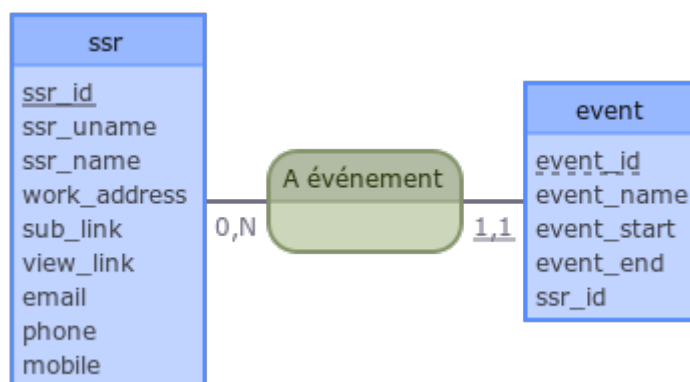
Une fois le bloc, le téléchargement des calendriers, la mise à jour de la table event et le tri correctement mis en place, le module répondait parfaitement à la demande. Les administrateurs peuvent ajouter, modifier ou supprimer un SSR. Les utilisateurs peuvent voir la disponibilité des SSR en "temps réel".

Afin de rendre le module plus complet, mon maître de stage m'a suggéré d'ajouter des informations comme l'adresse mail et les numéros de téléphone des SSR dans le bloc.

Dans l'optique de permettre aux administrateurs de saisir le moins de données possible, leur récupération s'est faite via l'annuaire LDAP* de l'ONF. J'ai donc commencé par me documenter sur le fonctionnement de LDAP, qui est une norme pour les systèmes d'annuaires utilisant une structure arborescente et qui est basée sur le protocole du même nom.

En m'inspirant de la classe d'authentification via LDAP intégrée à Xoops, j'ai codé une classe Ldap permettant de se connecter au serveur de l'annuaire et implémentant une fonction getUserData() retournant un tableau contenant le nom, le prénom, l'adresse du lieu de travail, l'adresse mail, le numéro de téléphone fixe et le numéro de portable d'un SSR à partir de son identifiant (pdidiot pour mon maître de stage par exemple). Une fois la classe fonctionnelle, j'ai modifié le reste du module pour gérer ces nouvelles données, non prévues au cahier des charges.

J'ai tout d'abord ajouté des champs pour l'identifiant, l'adresse, le mail, le téléphone et le portable dans la table ssr. Puis j'ai modifié les formulaires de la page d'administration ainsi que le tableau pour afficher les nouvelles données. Maintenant ce n'est plus le nom et le prénom qu'il faut fournir en plus des deux liens de publications, mais seulement l'identifiant du SSR.



Tables du module après ajout des champs pour accueillir les données LDAP

Coté traitement, après obtention de l'identifiant et des liens de publication depuis la variable superglobale \$_POST, j'ai utilisé la fonction getUserData() de la classe Ldap pour récupérer les données depuis l'annuaire. Si le tableau obtenu est vide, un message prévenant que l'identifiant n'est pas connu dans l'annuaire LDAP est affiché. Sinon, je récupère les données du tableau et je les ajoute à la requête SQL, avant de l'exécuter. Il en est de même pour le formulaire de modification, ainsi, lorsque l'on valide ce formulaire, les données provenant du serveur LDAP sont actualisées, ce qui est utile si jamais un SSR change de lieu de travail ou de numéro de portable par exemple.

Ajouter un SSR

Identifiant du SSR

Lien d'abonnement

Lien de consultation

ID du SSR	Nom du SSR	Adresse de travail	Lien d'ab...	Lien de c...	Email	Téléphone	Portable	Actions
cabraham	ABRAHAM Cyril	5 Rue Girardet 54000 Nancy	http://so...	http://so...	cyril.abrah...			Modifier Supprimer
pdidiot	DIDIOT Philippe	60 Boulevard Poincaré 55000 Bar-Le-Duc	http://so...	http://so...	philippe.didi...			Modifier Supprimer
pg70812	GAUTHEROT Philippe	Maison Forestières De Blanche Fo 52200 Langres	http://so...	http://so...	philippe.ga...			Modifier Supprimer
ngripper	GRIMMER Nicolas	24 Route De Phalsbourg 57400 Sarrebourg	http://so...	http://so...	nicolas.gri...			Modifier Supprimer

Page d'administration après les modifications pour utiliser LDAP

Après avoir testé le module avec plus de SSR, il s'est révélé que pour certains, le champ "onf-adresse-pro-complete" (champ contenant l'adresse de travail) était vide. J'ai donc ajouté "postaladdress" dans les champs à récupérer depuis l'annuaire LDAP. J'ai ensuite modifié le code PHP qui gère les formulaires d'ajout et de modification pour qu'il utilise la valeur de "postaladdress" si jamais "onf-adresse-pro-complete" est vide. J'ai également mis en place la normalisation des adresses (une majuscule au début de chaque mot, même lorsqu'il y a un trait d'union), ainsi que la suppression des éléments superflus lorsque "postaladdress" est utilisé (Cedex, Bureaux De La Dt Lorraine, etc) afin que les adresses soient uniformes.

Afin d'éviter d'attendre la prochaine actualisation de la table event pour charger les événements d'un SSR récemment ajouté, j'ai mis en place une fonction similaire à l'ancienne version de `ssr_loadCalendar` ([cf. D. §5](#)) dans le fichier `index.php` de la page d'administration du module.

C. Le bloc du module

Le bloc est la partie du module visible par tous les utilisateurs. Ce dernier peut être configuré pour s'afficher sur différentes pages, et à différents emplacements. Le code d'un bloc est au minimum constitué d'une fonction retournant une variable s'appelant obligatoirement `block` et d'un template Smarty. On doit préciser le fichier du bloc, le nom de la fonction et le template dans le fichier `xoops_version.php`.

Pour notre module, la fonction s'appelle `ssr_block()` et elle devra afficher la liste des SSR, ainsi que leur disponibilité au moment de la consultation du site.

Pour commencer, la fonction affiche seulement la liste sans prendre en compte les calendriers. Elle récupère donc le nom et le lien de consultation des SSR enregistrés, triés par nom, dans un tableau, qui est passé dans la variable `block` pour être utilisé dans le template. Les variables Smarty pouvant être appelées n'importe où dans le template, on peut appeler la variable contenant le lien de consultation directement dans le `href` de la balise `<a>` pour créer le lien vers le calendrier en ligne de chaque SSR.

Une fois le filtre mis en place (cf. E. §3), j'ai utilisé les variables Smarty pour définir la couleur du texte directement dans le style de la balise `<a>`. Grâce à la condition `if` utilisable dans le template, "(Disponible)", "(Indisponible)" et "(Occupé)" s'affichent en fonction de la valeur du statut.



ABRAHAM Cyril (Disponible)	ABRAHAM Cyril (Disponible)
DIDIOT Philippe (Disponible)	DIDIOT Philippe (Indisponible)
GAUTHEROT Philippe (Disponible)	GAUTHEROT Philippe (Indisponible)
GRIMMER Nicolas (Disponible)	GRIMMER Nicolas (Disponible)

Bloc avant et après la mise en place des filtres

Lors de l'ajout des données provenant de l'annuaire LDAP (cf. B. §4), j'ai simplement ajouté l'adresse, le mail et les numéros de téléphones au tableau qui est passé dans la variable `block`. Dans le template, j'ai ajouté l'appel aux nouvelles variables, en mettant l'adresse mail dans un lien "mailto:" et les numéros de téléphones dans des liens "tel:". Ainsi, les utilisateurs pourront directement contacter un SSR (le lien "mailto:", par exemple, permet d'ouvrir directement le client Outlook avec une fenêtre pour écrire un nouveau mail) depuis le site.

Options*	
Adresse de travail	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Adresse email	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Téléphone	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Portable	<input checked="" type="radio"/> Oui <input type="radio"/> Non

Partie Options dans les paramètres du bloc

Lorsque l'on met en place un bloc, on a la possibilité d'ajouter des options, qui seront affichées dans la page des paramètres du bloc. J'en ai donc ajouté pour afficher ou non chacun de ces nouveaux éléments.

Compte tenu du nombre de SSR, toujours afficher les détails allait poser problème (la hauteur nécessaire serait beaucoup trop grande). J'ai donc mis en place un script jQuery qui permet d'afficher les informations d'un SSR seulement lors du survole sur son nom. J'ai aussi ajouté une option pour activer ou désactiver cette fonctionnalité. Et pour améliorer l'apparence, j'ai mis en place des conditions dans le template Smarty pour ne pas afficher d'espace vide en cas de données manquantes (numéro de portable par exemple).

Je me suis ensuite concentré sur les options du bloc, en migrant tout d'abord les mots-clés permettant de filtrer l'indisponibilité d'un SSR, ainsi que les codes RVB des statuts, qui étaient jusque-là écrits en dur dans le code. J'ai ensuite ajouté des options pour définir la fréquence de mise à jour de la

ABRAHAM Cyril
 (Disponible)

DIDIOT Philippe
 (Indisponible)
 60 Boulevard Poincaré
 55000 Bar-Le-Duc
 03 83 33 33 33
 06 83 33 33 33

GAUTHEROT Philippe
 (Indisponible)

GRIMMER Nicolas
 (Occupé)

Bloc après l'ajout des données LDAP et du script

CUISENIER Alain
 Disponible

DANCOURT Thomas
 Congés

DIDIOT Philippe
 Absence

FISCHER Francois
 Congés

GAUTHEROT Philippe
 Congés Fractionnement

GRIMMER Nicolas
 CA - Sur site

Bloc après modifications et affichage de la localisation

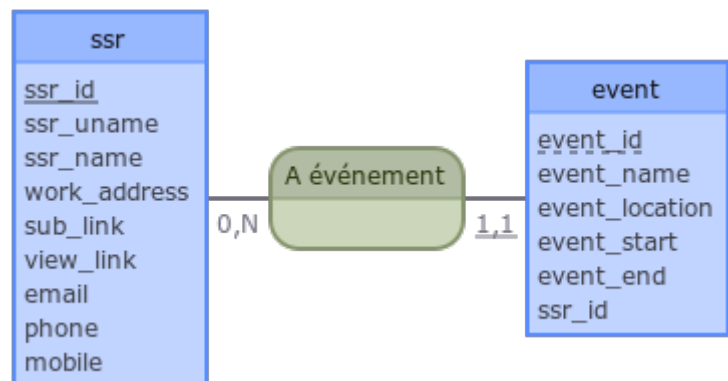
table event ([cf. D. §6](#)), et pour forcer la mise à jour de la table à chaque chargement. J'ai restructuré cette partie afin qu'elle soit plus compréhensible et mieux organisée, puis j'ai ajouté une option pour définir des mots-clés laissant un SSR disponible (l'accueil d'un stagiaire par exemple).

Après une remarque de Michel SCHUTZ, j'ai modifié le code du bloc pour qu'il affiche le nom de l'événement, plutôt que "(Indisponible)" ou "(Occupé)", afin que cela soit plus clair pour les utilisateurs.

Concernant la localisation des événements, j'avais, dans un premier temps, ajouté cette dernière dans le nom des événements (lorsqu'elle était indiquée), afin de filtrer plus d'éléments. En effet, un des SSR (Eric PRUVOST) ajoutait ses stagiaires à son calendrier, et renseignait qu'il s'agissait de stagiaires dans le champ Emplacement dans Outlook.

Par la suite, j'ai ajouté un champ pour la localisation dans la table event, car la localisation sera finalement utilisée pour indiquer si un SSR est sur site ou non. Cette fonctionnalité devait être à l'origine mise en place en récupérant le statut Skype des SSR, mais des modifications sur les serveurs Skype étaient nécessaires, et ni les SSR, ni moi, n'avions les habilitations pour les faire. Puis j'ai ajouté une option pour saisir les mots-clés permettant de filtrer les emplacements correspondant à "Sur site".

Après avoir mis en place l'enregistrement des événements récurrents (cf. D. §7), j'ai fait face à un problème. En effet, il arrivait parfois qu'un SSR ait plusieurs événements rendant Indisponible simultanément, et par exemple, même si un SSR était en Congés, ce n'était pas celui-ci qui était affiché dans le bloc. J'ai donc ajouté une option pour saisir les mots-clés prioritaires



Tables du module après ajout de la localisation

(Tempus, Congés, Férié, etc), afin qu'un autre

Informations	
Toujours afficher les informations	<input type="radio"/> Oui <input checked="" type="radio"/> Non
Adresse de travail	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Adresse email	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Téléphone	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Portable	<input checked="" type="radio"/> Oui <input type="radio"/> Non
Libellé événement privé	<input type="text" value="Rendez-vous privé"/>
Mise à jour des événements	
Fréquence de mise à jour	<input type="text" value="2"/> heure(s)
À chaque chargement de page	<input type="radio"/> Oui <input checked="" type="radio"/> Non
Mots-clés (séparés d'une virgule)	
Disponible	<input type="text" value="Stagiaire"/>
Indisponible	<input type="text" value="Tempus,RTT,Congés,Férié,Réunion,Rendez-vous,RDV,CFA,CESi,Télétravail"/>
Indisponible (Prioritaire) <i>Ex: congés et réunion en même temps</i>	<input type="text" value="Tempus,RTT,Congés,Férié"/>
Sur site	<input type="text" value="Sur site,Bureau"/>
Couleurs	
Disponible	<input type="text" value=""/>
Occupé	<input type="text" value=""/>
Indisponible	<input type="text" value=""/>

Options après la restructuration

événement Indisponible “n’écrase pas” le contenu de la variable \$status si jamais un événement Indisponible prioritaire a été trouvé avant. Puis j’ai également ajouté une option pour saisir le libellé à donner aux rendez-vous privés.

Juste avant la mise en production, j’ai présenté le module à quelques SSR. Ils m’ont alors demandé d’afficher les SSR en fonction de leur statut. Ce changement a impliqué beaucoup de modification dans le code. En effet, le statut n’étant pas enregistré dans la base de donnée, un "ORDER BY status" n’est pas possible. J’ai donc créé la fonction `ssr_getSSRs()` qui retourne un tableau lui-même composé de trois tableaux, un pour chaque statut. La fonction va récupérer la liste des SSR et les ajouter à un des tableaux en fonction du statut obtenu avec la fonction `ssr_getStatus()`. Le template du bloc a également été changé, il est maintenant composé de trois boucles `foreach` (une pour chaque tableau). La liste des SSR s’affiche donc avec les SSR disponibles en premier, puis les SSR occupés et enfin les SSR indisponibles.

PRUVOST Eric
Disponible

SCHUTZ Michel
Disponible

GAUTHEROT Philippe
Inter IT - Chaumont - Sur site

TRICOT Gaylor
Déplacement Aube-Marne/Ardennes - Sur site

DANCOURT Thomas
Congés

GRIMMER Nicolas
RTT

Bloc après la mise en place du tri en fonction du statut

```
// Sélectionne l'id, le nom et le lien de consultation des SSR triés par nom
$query = $xoopsDB->queryF('SELECT ssr_id, ssr_name, work_address, view_link, email, phone, mobile FROM '
    . $xoopsDB->prefix("ssr_disponibility_ssr") . " ORDER BY ssr_name;");

while( $row = $xoopsDB->fetchArray( $query ) ) {
    $ssr = array();
    $ssr['ssr_id'] = $row['ssr_id'];
    $ssr['ssr_name'] = $row['ssr_name'];
    $ssr['work_address'] = $row['work_address'];
    $ssr['view_link'] = $row['view_link'];
    $ssr['email'] = $row['email'];
    $ssr['phone'] = $row['phone'];
    $ssr['mobile'] = $row['mobile'];
    $ssr['status'] = ssr_getStatus( $row['ssr_id'] , $options );

    // Place le SSR dans un des tableaux en fonction de son statut
    if( $ssr['status']['status'] == 'available' ) {
        $available[] = $ssr;
    } else if( $ssr['status']['status'] == 'busy' ) {
        $busy[] = $ssr;
    } else if( $ssr['status']['status'] == 'unavailable' ) {
        $unavailable[] = $ssr;
    }
}
```

Code du tri en fonction du statut dans la fonction `ssr_getSSRs()`

D. Téléchargement des calendriers et mise à jour de la table event

Pour que le module fonctionne, il est nécessaire de télécharger les calendriers des SSR et d'enregistrer les événements dans la table event.

J'ai donc mis en place le téléchargement des calendriers via le lien d'abonnement. J'ai tout d'abord créé une fonction `ssr_loadCalendar()`, qui prend comme paramètre l'id du ssr concerné, et qui est appelée pour chaque SSR. Cette dernière commence par utiliser la fonction `fopen()`, avec le lien d'abonnement en paramètre, et enregistre le fichier récupéré dans un fichier de nom `ssr_id.ics` grâce à la fonction `file_put_contents()`. Avec la classe PHP `iCalEasyReader`, trouvée sur phpclasses.org, les fichiers `.ics` sont ensuite exportés dans des tableaux PHP, dans lesquels sont récupérés le nom, la date de début et la date de fin des événements. J'ai modifié légèrement la classe car cette dernière utilisait des syntaxes non supportées par la version PHP installée sur le serveur.

Avant de mettre les données dans la table, j'ai normalisé les dates des événements pour les mettre dans des champs `DATETIME`. J'ai également protégé la requête des erreurs dues à la possible présence de caractères spéciaux dans le nom des événements, avec la fonction `addslashes()`.

```
// Normalisation des dates et heures
// Début
$dtstart = $lines['VEVENT'][$i]['DTSTART']['value'];
// Pour les événements définis sur une journée entière
if( isset( $lines['VEVENT'][$i]['DTSTART']['type'] ) ) {
    $event_start = substr( $dtstart, 0, 4 )."-".substr( $dtstart, 4, 2 )."-".substr( $dtstart, 6, 2 )." 00:00:00";
} elseif( isset( $lines['VEVENT'][$i]['DTSTART']['TZID'] ) ) { // Pour les événements définis avec une heure de début
    $event_start = substr( $dtstart, 0, 4 )."-".substr( $dtstart, 4, 2 )."-".substr( $dtstart, 6, 2 );
    $event_start .= " ";
    $event_start .= substr( $dtstart, 9, 2 ).":".substr( $dtstart, 11, 2 ).":".substr( $dtstart, 13, 2 );
}
```

Code de la normalisation de la date de début de l'événement dans la fonction `ssr_loadCalendars()`

Après avoir effectué des tests avec beaucoup de calendriers, il s'est révélé que le chargement des pages était assez long, car le téléchargement des fichiers `.ics` et l'enregistrement des événements dans la base se faisait à chaque fois. J'ai donc mis en place un cookie de session pour que la mise à jour des événements ne se fasse que lors de la connexion d'une personne. Ainsi, vu que les personnes se connectent à différents moments de la journée, la table des événements sera actualisée régulièrement, au cas où des SSR ajoutent des événements pour le jour même.

La mise en place de ce dernier m'a forcé à modifier la fonction `ssr_loadCalendar()` pour qu'elle fasse la mise à jour pour tous les SSR d'elle même, sans avoir à être appelée pour chaque SSR. Je l'ai donc renommé

ssr_loadCalendars() pour la différencier de l'ancienne version, qui est toujours utilisée dans la page d'administration du module. Puis j'ai ajouté la récupération de la localisation ([cf. C. §10](#)) dans les tableaux PHP provenant de la classe iCalEasyReader.

J'ai par la suite remplacé le cookie de session par un cookie normal, qui expire tous les soirs. Puis j'ai finalement abandonné les cookies, qui obligeaient la table à s'actualiser trop régulièrement, et qui gênaient nettement les utilisateurs. Maintenant, lorsque la table est mise à jour, cela enregistre la date et l'heure (obtenu avec la fonction time()) dans un fichier. Et lors de chaque chargement de page, cela compare ce qui est enregistré avec la date et l'heure actuelle. Si la table n'a pas été mise à jour depuis plus longtemps que la fréquence définie dans les paramètres du bloc, cela la met à jour et enregistre la nouvelle date et heure dans le fichier.

```
// Mise à jour de la table event
if( $options[6] == 'true' ) { // Si la mise à jour à chaque chargement est activée
    // Charge tous les calendriers
    ssr_loadCalendars();

    $block['updated'] = 'Mise à jour forcée';
} else {
    $last_update = fopen( './modules/ssr_disponibility/blocks/last_update' , 'r+' );
    if( !$last_update ) {
        $last_update = fopen( './modules/ssr_disponibility/blocks/last_update' , 'a+' ); // Si le fichier n'existe pas
    }
    if( $last_update ) {
        $date = fgets( $last_update );
        // Si le fichier est vide ou si cela fait plus de n heures que la table event n'a pas été mäj
        if( $date === false || time() - $date > 60*60*$options[5] ) {
            fseek( $last_update , 0 );
            fputs( $last_update , time() );

            // Charge tous les calendriers
            ssr_loadCalendars();

            $block['updated'] = 'Mise à jour';
        }
        fclose( $last_update );
    }
}
```

Code chargé de mettre à jour la table event dans la fonction ssr_block()

J'ai ensuite remarqué que seule la première itération d'un événement récursif était enregistrée (dû à la manière dont ce type d'événement est enregistré dans un fichier .ics). J'ai donc modifié la fonction ssr_loadCalendars() afin que cela enregistre toutes les itérations des événements récursifs, en récupérant la fréquence (jour, semaine, etc), l'intervalle (2 jours, 1 semaine, etc) et la date de fin de la répétition. Puis en faisant une boucle qui ajoute l'événement à la requête, avance la date de début et de fin en fonction de la fréquence et de l'intervalle, et cela tant que la date de début est inférieure à la date de fin de la récursivité.

E. Classement des événements

Une fois l'enregistrement des événements fonctionnel, j'ai mis en place la fonction `ssr_getStatus()`, qui classe les événements pour connaître le statut des SSR. La fonction récupère nom des événements ayant pour `ssr_id` l'id du SSR concerné, et ayant une date de début inférieure à la date actuel et une date de fin supérieure. Ainsi, si la requête renvoie un tableau vide, le SSR est disponible, et si le tableau contient au moins un enregistrement, l'utilisation de mots-clés va permettre de déterminer si le SSR est occupé ou indisponible. Le plus simple est de définir des mots-clés pour les événements indisponibles, donc si le nom de l'événement contient un de ces mots-clés, le SSR est indisponible, sinon il est occupé.

Cette fonction retourne un tableau contenant un statut et un code RVB, ces derniers sont définis différemment suivant dans la disponibilité du SSR. Le code RVB est utilisé dans le template Smarty directement dans le style de la balise `<p>` pour définir la couleur de nom du SSR. Le statut, quant à lui, est utilisé pour afficher "Disponible", "Occupé" ou "Indisponible" à côté du nom.

```
do {
    if( ( !$row || strpos( $row['event_name'] , explode( ',' , $options[7] ) ) )
    && ( $status['status'] != 'busy' || $status['status'] != 'unavailable' ) ) {
        // Si il n'y a pas d'événement ou si il y a un événement filtré Disponible
        $status['color'] = $options[11]; // Vert
        $status['status'] = 'available';
    } elseif( strpos( $row['event_name'] , explode( ',' , $options[8] ) ) && !$prio ) {
        // Si indisponible
        $status['color'] = $options[13]; // Rouge
        $status['status'] = 'unavailable';
        $status['name'] = preg_replace( '#\(.*)#s' , '' , $row['event_name'] );
    } elseif( $status['status'] != 'unavailable' ) {
        // Si occupé
        $status['color'] = $options[12]; // Orange
        $status['status'] = 'busy';
        $status['name'] = preg_replace( '#\(.*)#s' , '' , $row['event_name'] );
    }

    if( $status['status'] == 'busy' ) {
        if( $row['event_location'] == '' || strpos( $row['event_location'] , explode( ',' , $options[10] ) ) ) {
            $status['location'] = 'Sur site';
        } else {
            $status['location'] = $row['event_location'];
        }
    }

    if( strpos( $row['event_name'] , explode( ',' , $options[9] ) ) ) $prio = true;
} while( $row = $xoopsDB->fetchArray( $query ) );
```

Code chargé de filtrer les événements dans la fonction `ssr_getStatus()`

Par la suite, j'ai migré les mots-clés dans les paramètres du bloc ([cf. C. §8](#)), et ajouté le nom de l'événement au tableau retourné par la fonction, car il sera affiché à la place de "Occupé" ou "Indisponible". J'ai également ajouté un filtre pour les événements laissant un SSR disponible et un second pour définir un booléen à true si

le nom de l'événement contient un des mots-clés de la liste Indisponible prioritaire ([cf. C. §12](#)) afin de ne pas écraser ce dernier si un événement Indisponible est filtré après.

Au final, la fonction `ssr_getStatus()` retourne un tableau contenant 4 éléments : la couleur, le statut, le nom de l'événement et la localisation. Cette dernière est par défaut définie à "Sur site" si elle n'était pas définie dans l'événement du calendrier Outlook.

F. Mise en production

Le développement et les tests du module étaient jusque-là fait sur le serveur de test xiis06-300. Pour finaliser le projet j'ai installé le module sur le serveur de production xiis06-000.

La mise en production m'a confronté à un problème assez important. En effet, le bloc ne s'affichait pas sur la page d'accueil lorsque l'on se connectait à un compte Webmestre. Le problème était lié à un autre bloc qui était affiché uniquement sur la page d'accueil des Webmestres. Ce dernier était fourni par le module Webalizer et permettait de connaître la fréquentation du site.

Malgré différents tests, le bloc de mon module ne s'affichait toujours pas. Le bloc de Webalizer étant responsable de ce bug, Nicolas GRIMMER et Cyril ABRAHAM ont décidé de le retirer de la page d'accueil des Webmestres et de l'afficher sur une page dédiée et réservée aux Webmestres.

Le mercredi 06 juin, cette mise en production a donné lieu à une réunion de restitution du projet avec une partie des SSR de l'ONF Grand-Est, dans laquelle j'ai présenté en détail le contenu, le fonctionnement et l'utilisation du module.

VI. Conclusion

Les dix semaines de stage au sein de l'ONF m'ont permis d'acquérir des connaissances sur Xoops, sur le fonctionnement du format de calendrier iCal et sur l'utilisation des outils de Microsoft comme Outlook. Ce stage a également renforcé mes compétences en PHP, JavaScript et autres langages du web.

Sur le plan de la mise en application des connaissances, j'ai beaucoup appris. En effet, avoir un projet avec un but précis et une utilisation concrète m'a permis d'exploiter au maximum mes connaissances et d'acquérir des méthodes d'organisation et d'expérimentation plus poussées qu'auparavant.

Bien que nous n'ayons jamais utilisé Xoops durant notre formation, j'ai réussi à en comprendre le fonctionnement assez facilement. Le monde de l'informatique étant si vaste, rares auraient été les chances d'étudier les CMS avec Xoops, celui-ci étant bien moins utilisé que WordPress ou Drupal. Mes connaissances sur Drupal, acquises au cours de ma formation, m'ont tout de même été fort utile.

Ce stage m'a permis de découvrir une vie professionnelle bien différente de ce à quoi je m'attendais. En effet, bien que rigueur, efficacité et assiduité soit toujours de mise, j'y ai découvert une ambiance et un rythme de travail plus détendu que je l'imaginais.

Étant de nature assez timide, j'ai tout de même réussi à communiquer et à discuter assez facilement avec les membres de l'équipe informatique, qui se sont montrés très accueillant. Mon stage m'a permis d'apprendre à mieux communiquer dans un milieu professionnel, aussi bien à l'oral, qu'à l'écrit (dans les mails notamment).

Mon stage m'a permis de valider mon choix de poursuivre mes études dans l'informatique, car je me suis rendu compte que l'informatique est très présente quel que soit le domaine d'activité d'une entreprise. Et bien que je souhaite poursuivre mes études dans de l'informatique plus poussée que celle rencontrée dans l'entreprise, je suis très heureux d'avoir effectué mon stage au sein de l'ONF, entouré d'une équipe très accueillante et, je l'espère, satisfaite de mon travail durant ces dix semaines.



SSR Disponibilité

Développé par Nicolas Carbonnier

Table des matières

I. Présentation du module.....	1
II. Utilisation du module.....	1
A. Page d'administration.....	1
B. Les paramètres du modules.....	2
III. Outlook.....	3
A. Obtenir les liens de publication.....	3
B. Règles pour la définitions des événements Outlook.....	4

I. Présentation du module

Le module SSR Disponibilité permet d'afficher la liste des SSR ainsi que leur disponibilité en temps réel (via leur Calendrier Outlook).

La page d'administration du module permet d'ajouter, de modifier ou de supprimer un SSR. L'ajout (et la modification) se fait via un formulaire dans lequel il faut fournir l'identifiant LDAP du SSR que l'on souhaite ajouter, ainsi que les liens de publications de son calendrier Outlook.

Le bloc du module affiche la liste des SSR, avec une couleur définie en fonction de leur statut (Disponible, Occupé, Indisponible), ainsi que l'événement (et la localisation de ce dernier) si ils en ont un au moment de la consultation du site. Le clique sur un nom ouvre le calendrier du SSR concerné dans un nouvel onglet et le survole permet d'afficher l'adresse du lieu de travail, l'adresse mail, le numéro de téléphone et de portable (le mail et les numéros de téléphone sont également cliquables).

Les paramètres du bloc contiennent des options pour régler les éléments à afficher dans ce dernier, la mise à jour des événements et les couleurs des statuts ainsi que pour définir les mots clés pour les filtres.

II. Utilisation du module

Après avoir installé le module, comme un module Xoops classique, plusieurs éléments sont à paramétrer pour que le bloc soit fonctionnel.

A. Page d'administration

La page d'administration permet de gérer les SSR présents dans la base de données de module. C'est donc ici que l'on doit ajouter les SSR une fois le module installé.

Ajouter un SSR

Identifiant du SSR:

Lien d'abonnement:

Lien de consultation:

ID du SSR	Nom du SSR	Adresse de travail	Lien d'ab...	Lien de c...	Email	Téléphone	Portable	Actions
cabraham	ABRAHAM Cyril	5 Rue Girardet 54000 Nancy	http://so...	http://so...	cyril.abrah...	<input type="text"/>	<input type="text"/>	Modifier Supprimer
pdidiot	DIDIOT Philippe	60 Boulevard Poincaré 55000 Bar-Le-Duc	http://so...	http://so...	philippe.ddi...	<input type="text"/>	<input type="text"/>	Modifier Supprimer
pg70812	GAUTHEROT Philippe	Maison Forestière De Blanche Fo 52200 Langres	http://so...	http://so...	philippe.ga...	<input type="text"/>	<input type="text"/>	Modifier Supprimer
ngrimmer	GRIMMER Nicolas	24 Route De Phalsbourg 57400 Sarrebourg	http://so...	http://so...	nicolas.gri...	<input type="text"/>	<input type="text"/>	Modifier Supprimer

La page se présente donc comme ceci. Pour chaque SSR présent dans la base de données, on a plusieurs informations d'affichées, dont les plus importantes (et celles à fournir) sont l'identifiant et les deux liens de publications Outlook ([cf III. A.](#)). Les autres données sont directement récupérées depuis le serveur LDAP (la méthode d'authentification du site Xoops doit bien entendu être définie et paramétrée sur Annuaire Standard LDAP).

Les actions permettent d'effectuer ses dernières (Modifier et valider directement le formulaire permet de mettre à jour le SSR si les données du serveur LDAP ont changé).

B. Les paramètres du modules

Une fois les SSR ajoutés, il faut rendre visible le bloc. Ce dernier est fait pour être placé sur un des cotés de la page (de préférence à droite). Quant aux pages dans lesquelles le bloc est visible et pour quels groupes il l'est, c'est au choix des administrateurs.

Informations

Toujours afficher les informations: ☐ Oui ☒ Non

Adresse de travail: ☒ Oui ☐ Non

Adresse email: ☒ Oui ☐ Non

Téléphone: ☒ Oui ☐ Non

Portable: ☒ Oui ☐ Non

Libellé événement privé:

Mise à jour des événements

Fréquence de mise à jour: heure(s)

À chaque chargement de page: ☐ Oui ☒ Non

Mots-clés (séparés d'une virgule)

Disponible:

Indisponible:

Indisponible (Prioritaire):

Ex: congés et réunion en même temps

Sur site:

Couleurs

Disponible:

Occupé:

Indisponible:

Voici les options propres au bloc de ce modules :

- Informations :
 - Toujours afficher les informations : si Oui, cela désactive le script chargé de faire apparaître les informations au survol (ces dernières seront toujours affichées)
 - Les autres options permettent d'afficher ou non chaque éléments
 - Libellé événement privé : permet de définir le nom à donner aux événements définis privés sur Outlook
- Mise à jour des événements :
 - Fréquence de mise à jour : durée minimum entre deux actualisation des événements dans la base de données
 - à chaque chargement de page : La mise à jour se fait à chaque chargement (très lourd, à utiliser uniquement en cas de débogage)
- Mots-clés
 - Disponible : mots-clés définissant un SSR disponible, l'événement ne sera pas affiché sous le nom du SSR dans le bloc, qui sera affiché Disponible
 - Indisponible : mots-clés définissant un SSR indisponible
 - Indisponible (Prioritaire) : si plusieurs événements filtrés indisponibles ont lieu en même temps pour un SSR, ce sera celui contenant un mot-clé prioritaire qui sera affiché dans le bloc
 - Sur site : mots-clés pour lesquels la localisation dans le bloc indiquera "Sur site" (lorsque la localisation est vide, l'événement est par défaut sur site, [cf. III. B.](#))
- Couleurs : couleurs utilisées pour les différents statuts

III. Outlook

A. Obtenir les liens de publication

- Sur Outlook, aller dans Calendriers
- Mes calendriers > clique-droit sur "Calendrier"
- Partager > Publier ce calendrier... (ou Configurer ce calendrier publié...)

OU

- Aller sur Citrix (ou mesApplis sur le bureau)
- Lancer Outlook 2013
- Sélectionner le calendrier (en bas à gauche)
- Cliquer sur Publier en ligne (en haut)

Une fois connecté avec ses identifiants, configurer la publication comme suit :

Publication de calendrier - Calendrier

[En savoir plus sur la publication de calendrier](#)

Détails de la publication :

Détails limités

Publier mon calendrier :

3 jours avant aujourd'hui

30 jours après aujourd'hui

Niveau d'accès :

☒ Restreint

☐ Public

Les liens vers votre calendrier publié seront affichés après avoir cliqué sur Commencer la publication.

Puis cliquer sur Commencer la publication (ou Enregistrer). Les deux liens sont ensuite affichés.

B. Règles pour la définitions des événements Outlook

Afin que le module fonctionne de la manière la plus optimale, il est important de définir des règles à suivre lors de la définitions des événements dans les calendriers Outlook. Pour commencer, il est important de respecter les règles de grammaire et d'orthographe, aussi bien pour le confort de lecture dans le bloc que pour filtrer correctement les mots-clés (les accents n'ont pas d'incidence).

Le contenu entre parenthèses dans le nom des événements n'est pas affiché dans le bloc mais est utilisé lors du filtrage. Cela est utile pour détailler un peu plus un événement (pour soi) sans que cela prenne de la place dans le bloc.

Concernant l'emplacement de l'événement, celui-ci est considéré "Sur site" par défaut. Lorsqu'un événement a lieu sur le site dans lequel le SSR travaille habituellement, il est inutile de préciser l'emplacement (un filtre est tout de même mis en place pour filtrer les emplacements correspondant à "Sur site"). Pour tous les événements se localisant ailleurs que sur le lieu de travail, il est obligatoire d'indiquer l'emplacement.

L'emplacement est important pour savoir si un SSR peut être disponible, bien que occupé. Il sera affiché à côté du nom de l'événement dans le bloc, lorsque le statut relevé pour le SSR est "Occupé" (il est inutile de préciser l'emplacement lors d'un événement "Indisponible").

B. Glossaire

Annuaire LDAP : Norme pour les systèmes d'annuaire avec un modèle de données fonctionnant avec le protocole d'interrogation et de modification LDAP

API REST : API utilisant l'architecture REST (Representational State Transfer), une architecture offrant réactivité et fiabilité

Cubage : Estimation du volume du bois récolté

Datacenter : site physique regroupant les équipements constituant le système d'information d'une entreprise

DHCP : protocole réseau permettant d'attribuer automatiquement une adresse IP et un masque de sous-réseau à une machine

DNS : service informatique chargé de traduire les noms de domaine en adresse IP

Résineux : essences d'arbres produisant de la résine (Épicéa, Pin, Sapin, etc)

SCCM : System Center Configuration Manager, logiciel de gestion de système proposé par Microsoft, permettant la gestion de grands parcs d'ordinateur, la télédistribution d'applications, la gestion de correctifs, l'automatisation de tâches, etc

Tempus : Outil de gestion des absences du personnel ONF fonctionnaire ou contractuel de droit public

Triages : étendue de forêt confiée à un garde forestier

XAMPP : Ensemble de logiciels permettant de mettre en place un serveur Web, similaire à WAMP

C. Bibliographie

ONF – Rapport d'activité 2017

http://www.onf.fr/onf/++oid++63fa/@@display_media.html (Dernière consultation : 17/03/2018)

ONF – Connaître l'ONF

<http://www.onf.fr/onf/@@index.html> (Dernière consultation : 17/05/2018)

ONF – Produits et prestations de service

http://www.onf.fr/produits_prestations/@@index.html (Dernière consultation : 17/05/2018)

Wikipédia – PEFC

https://fr.wikipedia.org/wiki/Programme_de_reconnaissance_des_certifications_forestieres (Dernière consultation : 17/05/2018)

Xoops France – Mise en ligne de la version 2.4.5 (Tutoriel d'installation)

<http://www.frxoops.org/modules/news/article.php?storyid=1780> (Dernière consultation : 17/05/2018)

GitBook – Building XOOPS Module (Tutorial by Kaotik)

<https://legacy.gitbook.com/book/xoops/kaotik-module-tutorial/details> (Dernière consultation : 17/05/2018)

PHPClasses – PHP ICS Calendar Reader

<https://www.phpclasses.org/package/9277-PHP-Parse-iCalendar-ics-files-to-extract-event-details.html> (Dernière consultation : 17/05/2018)