

RAPPORT DE STAGE

Gestion de l'envoi de notifications automatique

Cube Heartbeat

PECH Thibault

Du 08 Février au 05 Mars 2021

Tuteur en entreprise : Monsieur François SOULARD

Tuteur académique : Monsieur Michel ARNAUD

Entreprise d'accueil : Agence Cube, 99 boulevard d'Angleterre,
85000 La Roche-sur-Yon

Etablissement / Formation : Lycée Jean De Lattre de Tassigny,
165 Rue Hubert Cailler, 85000 La Roche-sur-Yon / BTS SNIR - Session 2021

SOMMAIRE

| | |
|--|----|
| I. MISE EN SITUATION..... | 2 |
| A. REMERCIEMENTS..... | 2 |
| B. INTRODUCTION..... | 3 |
| II. PRÉSENTATION DE L'ENTREPRISE..... | 4 |
| III. PRÉSENTATION DU SERVICE D'ACCUEIL..... | 5 |
| IV. ACTIVITÉS CONFIÉES AU STAGIAIRE..... | 6 |
| V. RÉFLEXIONS ET CONCLUSIONS..... | 17 |
| VI. ANNEXES..... | 18 |
| A. DOCUMENT TECHNIQUE EN ANGLAIS..... | 18 |
| B. 1 ^{er} ARTICLE DE PRESSE EN ANGLAIS..... | 19 |
| C. 2 nd ARTICLE DE PRESSE EN ANGLAIS..... | 23 |
| D. EVALUATION DE STAGE..... | 26 |
| E. ATTESTATION DE STAGE..... | 27 |

REMERCIEMENTS :

Je voudrais tout d'abord adresser toute ma gratitude au directeur et maître de stage, M. François SOULARD, pour sa confiance, sa disponibilité et surtout l'autonomie qu'il m'a offert pendant ce stage.

Je remercie également toute l'équipe de l'agence Cube, et tout particulièrement M. Alban POIRIER, M. Quentin DRAPPIER et M. Charles POULLAIN pour leur aide lors de divers problèmes rencontrés durant ces 4 semaines de stage.

Faire ce stage dans l'agence Cube a été un plaisir, j'ai pu apprendre beaucoup grâce à vous, notamment dans la méthode de travail.

Enfin, je tiens à remercier le corps enseignant, notamment M. Akim YOUNSI et M. Michel ARNAUD pour leur enseignement théorique et pratique, utilisés lors de ce stage.

INTRODUCTION :

Le BTS Systèmes Numériques, option Informatique et Réseaux requiert en première année un stage en entreprise de 6 semaines vers la fin de l'année scolaire.

La pandémie de la COVID-19 a malheureusement rendu ce stage impossible. Il a dû être reporté en seconde année pour une période de 4 semaines durant la période de Février - Mars.

Ces semaines de stages permettent d'exploiter nos connaissances acquises pendant notre formation dans le monde de l'entreprise.

Cela permet aussi de découvrir de nouvelles technologies qui ne sont pas abordées en cours.

Lors de la recherche de ce stage, je voulais trouver une entreprise qui était axée dans le développement web ou la maintenance de serveurs réseaux, des domaines que j'apprécie personnellement depuis mon plus jeune âge.

L'agence cube, spécialisée dans le développement d'applications mobiles et web a généreusement accepté ma candidature pour développer un site web pour vérifier le fonctionnement de leurs applications mobiles.

PRÉSENTATION DE L'ENTREPRISE :

L'Agence Cube, SARL située au 99 boulevard d'Angleterre à La Roche-sur-Yon est une entreprise qui propose le développement des sites web et applications mobiles sur mesure.



Elle est également éditrice de l'application Ma commune et moi, application proposant plus de proximité entre les concitoyens et de Moderato.social éditée en 2015, proposant la gestion des messages pour les relayer sur les réseaux sociaux.



MA COMMUNE
& MOI



Fondé en 2010 par M. François SOULARD & M. Alban POIRIER, l'entreprise recrute son premier employé en 2014.

Ils collaborent avec des entreprises françaises tel que le Crédit Mutuel, Colas ou encore Enedis pour élaborer des sites web et applications mobiles. Certaines associations (Quartiers, MFR), font également appel à leur aide pour concevoir des outils spécifiques à leurs demandes.

L'entreprise possède un effectif de 6 personnes :

- M. François SOULARD, occupant les postes de commercial et de l'administration.
- M. Alban POIRIER, occupant les postes de la gestion technique et de l'administration.
- Mme. Natalie GOSSELIN, occupant le poste de directrice commerciale.
- M. Charles POULLAIN, occupant le poste de développeur web.
- M. Quentin DRAPPIER, occupant le poste de développeur web.
- Mme. Marion BROCHARD, occupant le poste de commerciale.

Leur principal concurrent à la Roche sur Yon est l'entreprise FBO, fondée en 2011, éditeur d'applications web, de logiciels et de maintenance réseau.

PRÉSENTATION DU SERVICE D'ACCUEIL :

Après une semaine en télétravail, j'ai pu intégrer l'équipe de développement de l'Agence Cube pour continuer ma mission assignée, le développement d'un site web pour répondre à un problème d'envoi de notifications.

Suite à l'abandon du serveur d'émission de notifications développé par l'Agence Cube dû à des problèmes de maintenance et de mise à jour d'Android/iOS, ils ont voulu passer par un intermédiaire, Amazon Web Services pour gérer les notifications push.

Mais un problème est très vite arrivé : il était impossible de vérifier si une notification à été envoyée.

Dû à une limitation de Google, développeur d'Android et d'Apple, développeur de iOS, il est impossible de savoir si une notification à été réellement reçue. Amazon Web Services est donc incapable de renvoyer un message, qu'il soit de réussite ou d'erreur.

On arrive donc au sujet de ma mission : créer un site web permettant l'envoi automatique et la gestion des notifications avec une tâche Cron récurrente.

Avec l'aide des deux développeurs web de l'entreprise, Charles et Quentin, j'ai pu continuer le développement de ce site web dans un environnement professionnel.

Ils ont également pu à plusieurs reprises, m'aider dans le développement en m'aidant dans le débogage du site web.

Lors de ce stage, j'ai pu en apprendre un peu plus sur l'équipe de l'Agence Cube et notamment sur la qualification du personnel :

- M. Charles POULLAIN est titulaire d'un BAC + 5
- M. Quentin DRAPPIER est titulaire d'un BAC + 4
- M. François SOULARD est titulaire d'un BAC + 4
- Mme. Marion BROCHARD est titulaire d'un BAC + 3
- M. Alban POIRIER est titulaire d'un BAC + 2

ACTIVITÉS CONFIÉES AU STAGIAIRE :

Lors de ma première journée de stage, j'ai pu discuter avec M. POIRIER et M. SOULARD de la mission à effectuer durant le stage.

Comme expliquée ci-dessus, je devais créer un site web permettant l'envoi automatique et la gestion des notifications avec une tâche Cron récurrente.

L'idée était de pouvoir ajouter, modifier ou supprimer une application ou un téléphone depuis un site web administrateur.

L'envoi des notifications push est gérée par le serveur linux grâce à la commande cron.

Cette commande permet d'exécuter à une date donnée, dans notre cas, tous les jours à 12h, le lancement d'un script PHP permettant l'envoi d'une requête Amazon Resource Names (ARN) aux serveurs d'Amazon.

Le serveur s'occupera ensuite d'envoyer les notifications aux téléphones concernés.

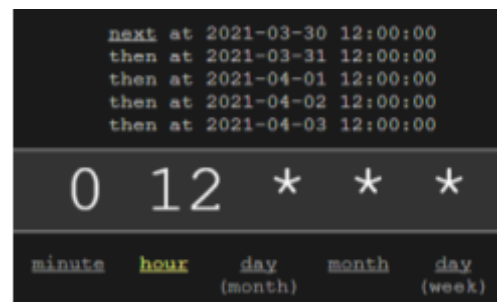


Figure 1. Requête CRON tous les jours à 12h
(https://crontab.guru/#0_12_*_*_*)

arn:partition:service:region:account-id:resource-id
arn:partition:service:region:account-id:resource-type/resource-id
arn:partition:service:region:account-id:resource-type:resource-id

Figure 2. Exemple de requêtes ARN (<https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html>)

Il n'y a pas eu de contrainte au niveau de la sécurité (connexion utilisateur / administrateur) car le code sera hébergé sur un serveur interne de l'entreprise et un fichier .htaccess permettra de faire une demande de connexion avant d'avoir la possibilité de l'utiliser.

J'ai donc commencé par faire la conception de l'affichage du site web. La première version me semblait être celle qui répondait le plus facilement à la mission confiée mais après discussion, cette version possédais beaucoup de défauts :

- Manque d'ergonomie pour l'utilisateur final
- Maintenance difficile
- Éléments de l'interface peu agréables pour l'utilisateur final



Figure 3. Conception du site web



Figure 4. Première version du site web

Le site avait trop d'informations sur la page pour qu'un utilisateur non initié comprenne du premier coup d'œil le fonctionnement du site web.

La maintenance allait être aussi compliquée si tout le code était inclus dans un seul fichier car il rendait la tâche du débogage impossible. Comment savoir si l'erreur venait de la base de données ou d'une erreur de saisie ?

Enfin, mon design utilisait des modals et des notifications. Ces outils, malgré leur utilité et l'effet agréable à l'œil posait un problème majeur : Comment faire en sorte que l'utilisateur comprenne qu'une action vient de se passer ?

L'agence cube a arrêté d'utiliser ces outils car leurs utilisateurs finaux ne comprenaient pas ce qui se passait à l'écran. Malgré l'amélioration des technologies web, pour un utilisateur lambda, un envoi ne se fait uniquement lors d'une actualisation de page web. Ils ne font pas toujours attention aux notifications sur l'écran.

Tous ces problèmes ont donc rendu inutilisable la première version du site web. Une seconde version a dû être développée, prenant en compte tous les changements cités ci-dessus pour une meilleure ergonomie.

- La seconde version possède une interface plus explicite avec une navbar permettant d'accéder aux actions précédemment accessibles dans le bouton +. On peut donc lister, ajouter, modifier ou supprimer une application ou un téléphone.



Figure 5. Seconde version du site web

- Le site web est maintenant sur plusieurs pages web, ce qui a également facilité le développement pour la suite.

Lors du développement je me suis servi de plusieurs outils différents :






| | |
|---|--|
|  | Visual Studio Code : Environnement de développement intégrée, qui m'a permis de concevoir le projet. |
|  | MAMP : Acronyme pour MacOS Apache MariaDB PHP, cet utilitaire tout-en-un a permis l'exécution de la page web avant le passage en production. |
|  | Git : Gestionnaire de version décentralisée permettant de garder un historique de modifications. |
|  | AdminLTE 3 : Modèle bootstrap utilisé lors du développement du site web. |
|  | Bootstrap : Librairie css et js permettant de créer des sites web responsive. |

Figure 6. Liste des outils de développements

AdminLTE permet d'adapter ses modèles pour avoir une interface adapté à ses besoins. Grâce à leurs documentation, il est possible d'ajouter des conteneurs spécifiques à AdminLTE.

Basée sur Bootstrap, il est également possible d'utiliser conjointement ces classes pour un contrôle précis des éléments de la page web.

Après les bases de la seconde version du site web fondée, j'ai commencé à créer les bases de données servant à stocker les informations utiles au bon fonctionnement du site web.



Figure 7. Prototype de la BDD

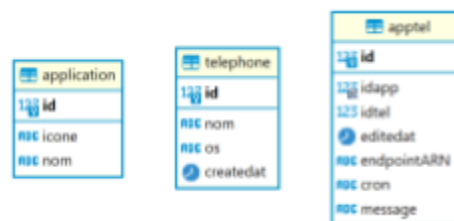


Figure 8. BDD Finale

Cette version à été modifiée plusieurs fois pendant le développement pour permettre.

J'ai dû par exemple ajouter des champs dans mes tables pour connaître la date à laquelle la modification a été effectuée ou encore, ajouter une colonne ARN pour pouvoir stocker ma requête Amazon.

On remarque également que la table logs a été remplacée par la table apptel. Ce changement a été fait vers la fin du développement pour permettre de faire un lien entre une application et un téléphone. Cette relation sera expliquée plus bas.

Pendant la conception, le site web a été structuré en trois onglets possédant chacune des actions quasiment identiques.

Dans les premières versions, il n'y avait que deux onglets: applications et téléphones. Ces sous onglets liste, ajouter, modifier et supprimer possèdent tous le même code : le plus gros changement est du côté backend. Le frontend possède quasiment le même code.

A la fin du stage, Charles m'a expliqué qu'il était possible d'éviter la répétition du code grâce au framework Vue.js. Cette technologie permet de modifier dans un seul fichier un lien de ma navbar pour ensuite l'inclure dans tous les autres fichiers. Grâce à cette technique, cela permet d'éviter de fouiller dans tous les fichiers les modifications à effectuer.

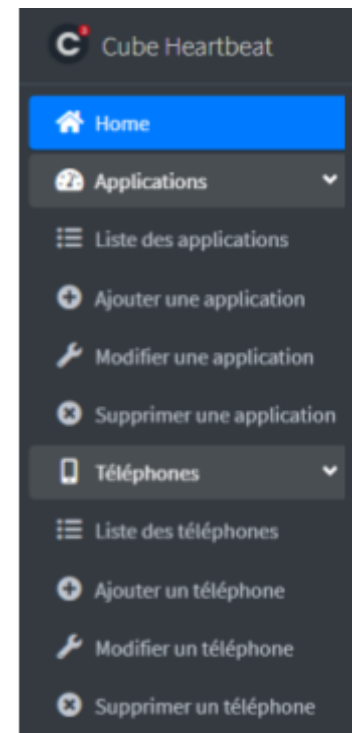


Figure 9. Navbar du site web

Pendant la conception, j'ai aussi eu besoin d'utiliser des scripts externes. J'ai donc utilisé le gestionnaire de dépendances Composer. Il permet de télécharger des bibliothèques externes pour l'implémenter dans le code PHP.

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"  
  
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'756890a4488ce9024fc62c56153228907f1545c228516cbf63f885e036d37e9a59  
d27d63f46af1d4d07ee0f76181c7d3') { echo 'Installer verified'; } else { echo  
'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"  
  
php composer-setup.php  
  
php -r "unlink('composer-setup.php');"
```

Figure 10. Installation de composer (<https://getcomposer.org/download/>)

```
{
  "require": {
    "aws/aws-sdk-php": "^3.173",
    "peppeocchi/php-cron-scheduler": "^3.1"
  }
}
```

Figure 11. Fichier composer.json

Après l'installation de Composer sur MAMP, j'ai pu télécharger le SDK Amazon permettant l'envoi de mon message push et la librairie permettant la gestion des tâches cron.

Charles m'a également donné l'accès à un script permettant d'envoyer une notification push directement à un téléphone. L'exécution de ce script sera décrite plus bas.

Le premier sous onglet affichage des données s'est avéré simple à mettre en œuvre. Il suffit de faire une simple requête SQL pour récupérer les données nécessaires.

```
if (!$conn->connect_errno) $res = $conn->query("SELECT * FROM
chb_telephones");
```

Figure 12. Requête SQL pour afficher la liste des téléphones (telephones/listetel.php)

La requête est décomposable en trois parties :

- La vérification de la connection à la BDD avec le if
- L'affectation de la requête SQL via la query
- La requête SQL entre guillemets

On exécute ensuite la requête suivante pour récupérer et afficher les données sur la page web.

```
$res->data_seek(0);
while ($row = $res->fetch_assoc()) {
    echo "<td>" . htmlspecialchars($row['nom']) . "</td>";
}
```

Figure 13. Extrait de la fonction de lecture des données (telephones/listetel.php)

Lors du développement, un problème est survenu : il était possible d'inclure des balises HTML dans les données.

La fonction htmlspecialchars a permis de régler ce problème.

Par défaut, sans cette fonction, si une donnée possédait le typage d'une balise HTML (exemple : <h1>, permet de transformer le texte en titre), le client web risquait d'afficher toutes sortes de résultats improbables voir illogique.

Ensuite, la partie ajouts des données à été la plus basique des quatres sous onglets car, il s'agissait de deux pages web. Une page HTML basique avec un formulaire demandant à l'utilisateur d'insérer les données pour le bon fonctionnement de la BDD et d'une page PHP lançant une requête SQL à la BDD.

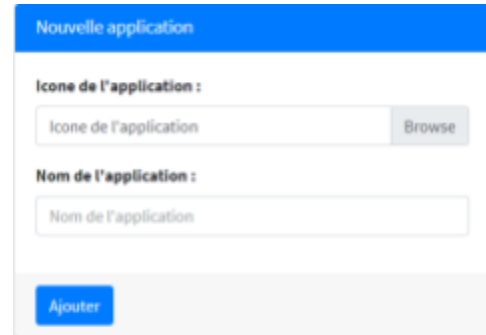


Figure 14. Formulaire application (applications/add/addtbdapp.php)

```
$sql = mysqli_prepare($conn, "INSERT INTO chb_telephones (nom, os)
VALUES (?, ?)");
mysqli_stmt_bind_param($sql, 'ss', $nom, $os);
```

Figure 15. Requête de la page téléphone préparé (telephones/add/addtelthink.php)

Encore une fois, l'entrée client à dû être sanétisée pour éviter l'insertion de commandes SQL vérolée (type OR 1 = 1).

Ce genre d'attaque (injection SQL) est très simple à mettre en œuvre et peut facilement dévoiler le contenu d'une BDD.

C'est pour cela que `mysqli_stmt_bind_param` s'avère utile : en indiquant le types de données à recevoir, cette fonction est capable d'insérer les valeurs sans avoir un risque d'exécution de code.

Le troisième sous onglet modifications est la plus complexe car elle demande d'exécuter plusieurs requêtes SQL à la fois. Pour décrire simplement, c'est une combinaison des requêtes de l'affichage des données et de la partie ajout des données. Le changement le plus flagrant entre l'ajout (Figure 15) et la modification vient de la requête à exécuter. Au lieu d'utiliser la requête INSERT, on utilisera le mot-clé UPDATE.

```
if (!$conn->connect_errno) $res = $conn->query("SELECT nom FROM
chb_telephones");
if($_POST['idtel']) {
    $_SESSION['idtel'] = $_POST['idtel'];
    header("Refresh: 0; url = editingtel.php"); }
```

Figure 16. Code PHP pour la sélection des téléphones (telephones/edit/edittel.php)

Ce code permet de passer la partie choix du téléphone (Figure 17) si la base de données comporte uniquement 1 téléphone.

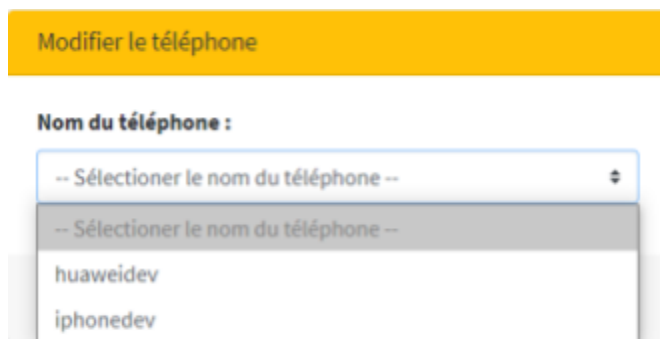


Figure 17. Modifier le téléphone (liste déroulante) (telephones/edit/edittel.php)

La liste déroulante affiche les données de la même manière qu'avec la fonction de la Figure 13.

La seule différence est qu'au lieu d'utiliser la balise <td>, on utilise la balise <option>.

La fenêtre modification de l'application / téléphone ne diffère pas vraiment du sous onglet ajouter : on retrouve le même formulaire. Seulement le côté backend change comme expliqué plus haut.

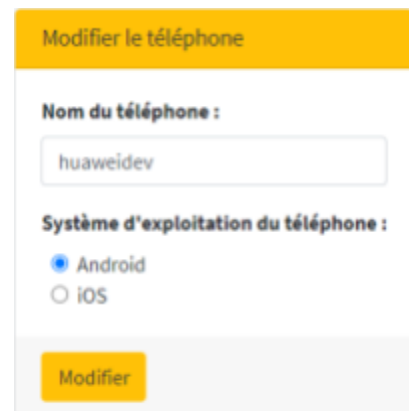


Figure 18. Modification du téléphone huaweidev (telephones/edit/edittel.php)

Enfin, la dernière sous partie suppression permet comme son nom l'indique, de supprimer un téléphone / application.

Encore une fois, on utilise un code semblable à la Figure 17 pour sélectionner son téléphone.

La seule différence visible dans le frontend est la couleur de la notification qui passe de l'orange au rouge.

Pour éviter une erreur, ce sous onglet est le seul à posséder un modal qui demande bien à l'utilisateur de confirmer la demande. On attire l'oeil de l'utilisateur en donnant une fenêtre rouge sur un fond noir transparent. Il est ensuite invité à cliquer sur le bouton vert de base. Cela permet d'éviter la suppression si on ne lit pas ce qui se passe à l'écran.

On comprend facilement que le bouton à cliquer est le rouge si l'utilisateur prend le temps de lire la notification.

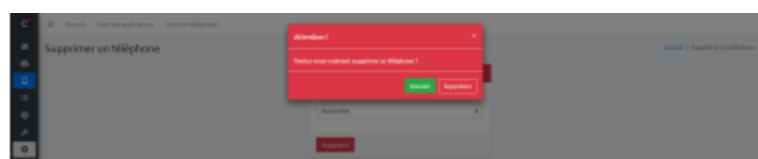


Figure 19. Notification de suppression (telephones/rm/rmtel.php)

Après l'exécution des sous onglets ajout, modification ou suppression, il est important de faire comprendre à l'utilisateur que sa demande à bien été traitée.

Comme expliqué ci dessus, l'Agence Cube n'utilise pas des notifications toast car l'utilisateur a de grandes chances de ne pas le voir.

Il est également préférable d'utiliser plusieurs pages web pour éviter des problèmes de compréhension lors du débogage d'une page comme indiqué plus haut.



Figure 20. Exemples de notifications push
<https://adminlte.io/themes/v3/pages/UI/models.html>

Après avoir réfléchi aux contraintes imposées, je suis parti sur un modèle de notification au texte dynamique.

Une première version dev des notifications utilisait une page de notification spécifique à l'action.

Cette technique s'est avérée très contre productif car elle rendait la maintenance ingérable lors de modifications dans la partie HTML statique de la page web (cf. Figure 9).

Les notifications de la version de production peut se résumer par ce diagramme de séquence :

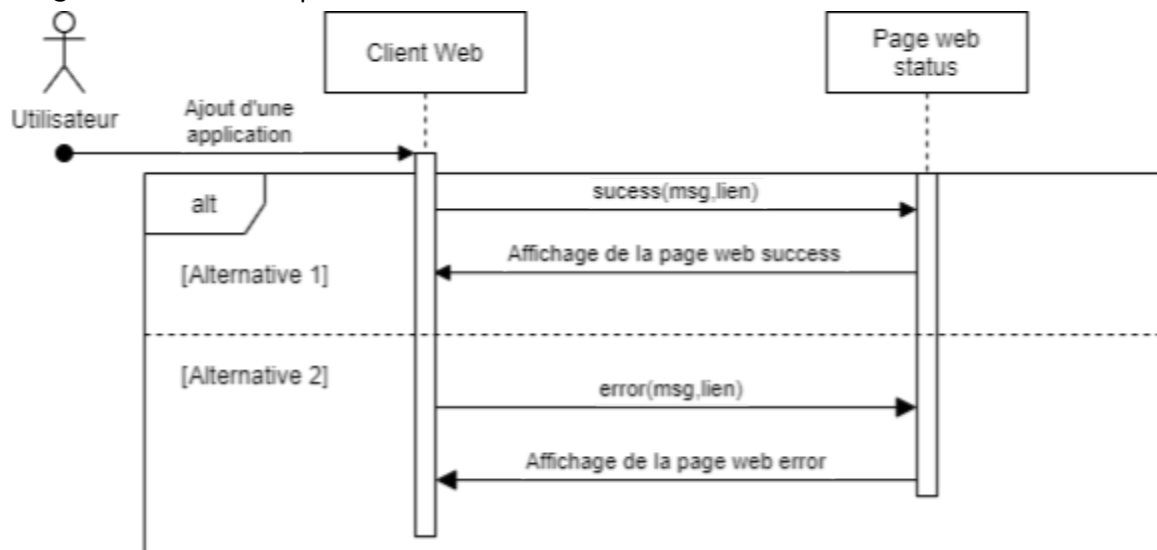


Figure 21. Diagramme de séquence

La page web demandant d'envoyer une notification exécute donc la fonction success ou error.

```
function success($msg, $link) {
    session_start();
    global $defaultsuccess, $defaultlink;

    if(empty($msg)) $_SESSION['success'] = $defaultsuccess;
    else $_SESSION['success'] = $msg;

    if(empty($link or $link == NULL)) $_SESSION['link'] = $defaultlink;
    else $_SESSION['link'] = $link;

    header("Refresh: 0; url = /cubeheartbeat/status/success.php"); }
```

Figure 22. Extrait de la fonction success (status/status.php)

Ce code backend récupère les données envoyées par la fonction success.

Ce script est inclus de base sur toutes les pages du site web.

Cela permet d'appeler les fonctions success et error sur n'importe quelle page.

Il est possible de ne pas renseigner les variables msg et link.

Dans ce cas là, ce sont les valeurs par défaut renseignées dans le code qui prendront le dessus.

Après traitement, la page renvoie le client sur la page success.php (Fig. 23).

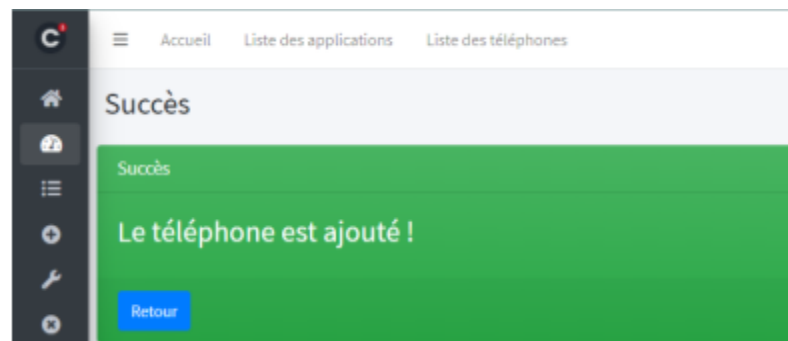


Figure 23. Exemple de notification de réussite

Toute la partie décrite ci-dessus m'a pris deux semaines et demie. On a donc pu faire évoluer le projet en ajoutant le lien entre le téléphone et l'application pour pouvoir utiliser la partie notification et les requêtes ARN.

La page d'accueil du site web montre donc la partie app-tel :



Figure 24. Page d'accueil du site, version production

Depuis cette interface, il est possible d'envoyer manuellement un message à un téléphone avec le bouton envoyer. Il est également possible d'aller sur la page pour modifier les liens entre les applications et les téléphones.

| Liste des téléphones | | | | |
|---|------------------|------------------------|---------------------|---|
| | Nom du téléphone | Système d'exploitation | Date de création | Paramètres additionnels |
|  | huaweidev | Android | 2021-02-24 09:16:33 |    |
|  | iphonedev | iOS | 2021-02-24 09:16:33 |    |

Figure 25. Liste des téléphone, version production

Dans la version finale, il est possible de modifier et de supprimer un téléphone sans passer par le menu déroulant.

C'est un problème que Alban et François avaient pointé.

Ils m'ont donné l'exemple d'une centaine de téléphones ou, il ce n'est pas vraiment efficace de chercher manuellement le nom du téléphone dans ce menu.

Pour revenir à l'app-tel, la partie lien et édition possèdent le même formulaire et quasiment les mêmes commandes.

La différence a déjà été évoquée plus haut (commentaires Figure 15).

Lier l'application au téléphone

Nom de l'application :

-- Sélectionner le nom de l'application --

Endpoint ARN :

Endpoint ARN

Message de notification :

Message de notification

Heures :

Minutes :

Jour :

Mois :

Jour de la semaine :

Planification CRON :

00
01
02
03
04
05
06
07
08
09
10
11

00
01
02
03
04
05
06
07
08
09
10
11

01
02
03
04
05
06
07
08
09
10
11
12

Janvier
Février
Mars
Avril
Mai
Juin
Juillet
Août
Septembre
Octobre
Novembre
Décembre

Lundi
Mardi
Mercredi
Jeudi
Vendredi
Samedi
Dimanche

0 * * * * *

Tout sélectionner

Tout sélectionner

Tout sélectionner

Tout sélectionner

Tout sélectionner

Lier

Figure 26. Lier une application au téléphone [appTel/addLinkappTel.php]

Sur ce formulaire, il est possible d'indiquer le nom de l'application, endpoint ARN, un message personnalisé pour la notification et la tâche CRON.

Pour la partie tâche CRON, j'utilise une librairie déjà existante, écrite par Thomas Ba.

J'ai remarqué lors de l'utilisation que cette librairie avait quelques bugs. J'ai donc dû modifier son code pour l'adapter à mes besoins, notamment le traduire en français.

```
for ($i = 0; $i <= 59; $i++) {  
    echo '<option selected="selected" value="' . $i . '">' . str_pad($i, 2, "0",  
    STR_PAD_LEFT) . '</option>'; }
```

Figure 27. Extrait du code modifié (<https://github.com/SRThibaultP/cron-expression-generator>)

Thomas avait listé tous les jours pendant un mois. Cette méthode marche mais pour éviter d'avoir un code trop surchargé, j'ai préféré utiliser une boucle for. Cela revient au même mais j'évite plus de 60 lignes de code "inutile". La partie suppression est comme les autres fois, identique à la Figure 19, que ce soit dans le frontend, ou le backend.

Pour finir, la partie cron : lors de l'installation du site web, il faut également exécuter cette commande dans un shell linux.

```
* * * * * php /var/www/html/notifications/sendnotifloopcron.php
```

Figure 28. Commande CRON exécutant le script toutes les minutes.

Ce fichier contient les commandes permettant au bon fonctionnement des tâches CRON.

Sans ce fichier, aucune notification automatique ne pourra être exécutée.

```
while ($entry = $list->fetch_assoc()) { $scheduler->call(  
    function ($arn, $msg) {  
        $dateenvoi = date("H:i:s");  
        PushNotificationHelper::setup();  
        if (empty($msg)) PushNotificationHelper::sendMessage("❤️ Heartbeat @ $dateenvoi", $arn);  
        else PushNotificationHelper::sendMessage($msg, $arn); },  
    [  
        $entry['endpointARN'],  
        $entry['message']  
    ],  
    'cronNotifier'  
)->at($entry['cron']);
```

Figure 29. Extrait du fichier de gestion des tâches CRON (notifications/sendnotifloopcron.php)

Ce fichier va récupérer la liste de la table app-tel, va regarder la tâche CRON assignée (non visible dans l'extrait) et va vérifier que la tâche soit exécutable à ce moment là.

S'il est exécutable, il va utiliser le script de Charles pour envoyer une requête ARN. Sinon il réessayera à la prochaine minute.

RÉFLEXIONS ET CONCLUSIONS :

Ce stage m'a permis d'acquérir de nouvelles connaissances dans les méthodes de travail dans le monde de l'entreprise.

Ce qui m'a marqué notamment sont :

- Le briefing à chaque début de journée, pour expliquer ce qu'on a fait le jour dernier et de ce que l'on compte faire aujourd'hui
- La méthodologie, tout ce qui est conception, maquettage du site web. Faire des maquettes, surtout en fin de projet, m'aurait évité de modifier le code plusieurs fois vers la fin du projet.

Ces deux points ont été ceux qui m'ont le plus marqué. Après ce stage, je me suis mis à l'appliquer pour notre projet de fin d'année. Notamment le briefing, cet atout important dans le travail de groupe.

J'ai également pu acquérir rapidement des connaissances sur la technologie bootstrap, une librairie que j'avais déjà entendue de nom mais jamais utilisée réellement.

Enfin, en discutant avec les deux développeurs de l'Agence Cube, j'ai pu mieux comprendre les attentes de l'employeur à l'employée. Charles a également pu me briefer sur ce que je pourrais améliorer pour pouvoir travailler en entreprise :

- Maîtriser le langage objet sur PHP
- Apprendre à utiliser VueJS pour éviter la duplication de ma page web statique
- Essayer d'exploiter les autres technologies web (ex. Laravel)

Ces quatre semaines ont vraiment été utiles pour approfondir mes connaissances en technologies web et pour découvrir les attentes dans le monde professionnel.

DOCUMENT TECHNIQUE EN ANGLAIS :

<https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html>

Amazon Resource Names (ARNs)

Amazon Resource Names (ARNs) uniquely identify AWS resources. We require an ARN when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

Contents

- [Format](#)
- [Resource ARNs](#)

Format

The following are the general formats for ARNs. The specific formats depend on the resource. To use an ARN, replace the *italicized* text with the resource-specific information. Be aware that the ARNs for some resources omit the Region, the account ID, or both the Region and the account ID.

```
arn:partition:service:region:account-id:resource-id
arn:partition:service:region:account-id:resource-type/resource-id
arn:partition:service:region:account-id:resource-type:resource-id
```

partition

The partition in which the resource is located. A *partition* is a group of AWS Regions. Each AWS account is scoped to one partition.

The following are the supported partitions:

- `aws` - AWS Regions
- `aws-cn` - China Regions
- `aws-us-gov` - AWS GovCloud (US) Regions

service

The service namespace that identifies the AWS product. For example, `s3` for Amazon S3 resources.

region

The Region. For example, `us-east-2` for US East (Ohio).

account-id

The ID of the AWS account that owns the resource, without the hyphens. For example, `123456789012`.

resource-id

The resource identifier. This part of the ARN can be the name or ID of the resource or a [resource path](#). For example, `user/Bob` for an IAM user or `instance/i-1234567890abcdef0` for an EC2 instance. Some resource identifiers include a parent resource (sub-resource-type/parent-resource/sub-resource) or a qualifier such as a version (resource-type:resource-name:qualifier).

Paths in ARNs

Resource ARNs can include a path. For example, in Amazon S3, the resource identifier is an object name that can include slashes (/) to form a path. Similarly, IAM user names and group names can include paths.

Paths can include a wildcard character, namely an asterisk (*). For example, if you are writing an IAM policy, you can specify all IAM users that have the path `product_1234` using a wildcard as follows:

```
arn:aws:iam::123456789012:user/Development/product_1234/*
```

Similarly, you can specify `user/*` to mean all users or `group/*` to mean all groups, as in the following examples:

```
"Resource": "arn:aws:iam::123456789012:user/*"
"Resource": "arn:aws:iam::123456789012:group/*"
```

You cannot use a wildcard to specify all users in the `Principal` element in a resource-based policy or a role trust policy. Groups are not supported as principals in any policy.

The following example shows ARNs for an Amazon S3 bucket in which the resource name includes a path:

```
arn:aws:s3:::my_corporate_bucket/*
arn:aws:s3:::my_corporate_bucket/Development/*
```

You cannot use a wildcard in the portion of the ARN that specifies the resource type, such as the term `user` in an IAM ARN.

The following is not allowed:

```
arn:aws:iam::123456789012:u*
```

1^{er} ARTICLE DE PRESSE EN ANGLAIS :

<https://www.bleepingcomputer.com/news/security/phps-git-server-hacked-to-add-backdoors-to-php-source-code>

Home > News > Security > PHP's Git server hacked to add backdoors to PHP source code

PHP's Git server hacked to add backdoors to PHP source code

By [Ax Sharma](#)

March 29, 2021 03:32 AM 1



In the latest software supply chain attack, the official PHP Git repository was hacked and the code base tampered with.

Yesterday, two malicious commits were pushed to the *php-src* Git repository maintained by the PHP team on their *git.php.net* server.

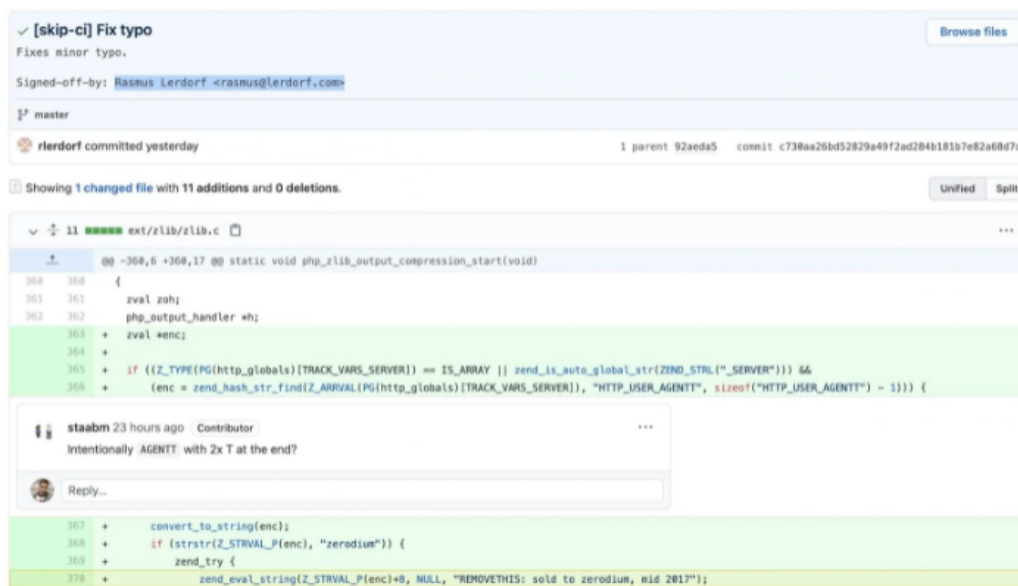
The threat actors had signed off on these commits as if these were made by known PHP developers and maintainers, Rasmus Lerdorf and Nikita Popov.

RCE backdoor planted on PHP Git server

In an attempt to compromise the PHP code base, two malicious commits were pushed to the official PHP Git repository yesterday.

The incident is alarming considering PHP remains the server-side programming language to power over [79% of the websites](#) on the Internet.

In the malicious commits [\[1, 2\]](#) seen by BleepingComputer, the attackers published a mysterious change upstream, "fix typo" under the pretense this was a minor typographical correction.



A malicious commit that attackers signed off on as Rasmus Lerdorf (illicitly) plants a remote code execution backdoor

However, taking a look at the added line 370 where `zend_eval_string` function is called, the code actually plants a backdoor for obtaining easy Remote Code Execution (RCE) on a website running this hijacked version of PHP.

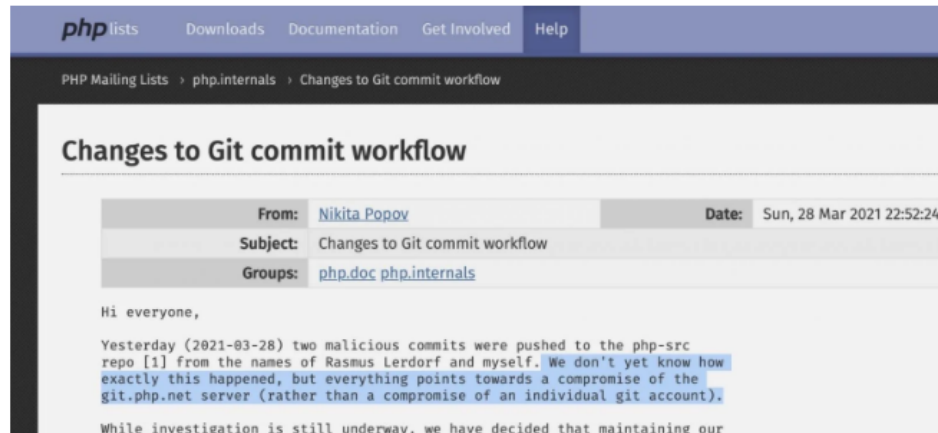
"This line executes PHP code from within the *useragent* HTTP header, if the string starts with 'zerodium'," responded PHP developer Jake Birchall to Michael Voříšek, who had first [pointed out](#) the anomaly.

In an email interview, PHP maintainer Nikita Popov told us:

"The first commit was found a couple hours after it was made, as part of routine post-commit code review. The changes were rather obviously malicious and reverted right away," Popov told BleepingComputer.

Additionally, the malicious commit was made in the name of PHP creator, [Rasmus Lerdorf](#).

But, that is hardly surprising as with source code version control systems like Git, it is possible to sign-off a commit as coming from anybody else [1, 2] locally and then upload the spoofed commit to the remote Git server, where it gives off the impression as if it had indeed been signed-off by the person named on it.



Official announcement (linked below) states the incident points towards a server compromise

Although a complete investigation of the incident is ongoing, according to PHP maintainers, this malicious activity stemmed from the compromised *git.php.net* server, rather than compromise of an individual's Git account.

PHP official code base migrated to GitHub

As a precaution following this incident, PHP maintainers have decided to migrate the official PHP source code repository to GitHub.

"While investigation is still underway, we have decided that maintaining our own git infrastructure is an unnecessary security risk, and that we will discontinue the *git.php.net* server."

"Instead, the repositories on GitHub, which were previously only mirrors, will become canonical," announced Popov.

With this change going forward Popov insists that any code changes be pushed directly to GitHub rather than the *git.php.net* server from this point on.

Those interested in contributing to the PHP project will now need to be added as a part of PHP organization on GitHub.

The instructions on that are provided in the same [security announcement](#).

For membership in the organization you would need to have two-factor authentication (2FA) enabled on your GitHub account.

"We're reviewing the repositories for any corruption beyond the two referenced commits," says Popov.

BleepingComputer reached out to both Popov and the PHP security team to find out the complete extent of this compromise, and if any code was distributed downstream before the malicious commits were caught.

"It may have been cloned/forked in the meantime, but the changes did not make it into any tags or release artifacts."

"The changes were on the development branch for PHP 8.1, which is due to release at the end of the year," Popov further told BleepingComputer.

The PHP team has confirmed to BleepingComputer that they plan on eventually decommissioning their git server in the upcoming days and moving to GitHub permanently.

Update 29-Mar-21 7:22 AM ET: Added response from PHP maintainer Nikita Popov.

Related Articles:

[Linux Foundation unveils Sigstore — a Let's Encrypt for code signing](#)

[New XcodeSpy malware targets iOS devs in supply-chain attack](#)

[Microsoft's Azure SDK site tricked into listing fake package](#)

[GitHub fixes bug causing users to log into other accounts](#)

[Gigaset Android phones infected by malware via hacked update server](#)

BACKDOOR

GITHUB

OPEN SOURCE

PHP

SOURCE CODE

SUPPLY-CHAIN ATTACK



AX SHARMA

Ax Sharma is a Security Researcher, Engineer, and Tech Columnist. His works and expert analyses have frequently been featured by leading media outlets like Fortune, The Register, TechRepublic, CIO, etc. Ax's expertise lies in vulnerability research, reverse engineering, software development, and web app security. He's an active community member of the OWASP Foundation and the British Association of Journalists (BAJ). Send any tips via email or Twitter DM.

2nd ARTICLE DE PRESSE EN ANGLAIS :

<https://www.bleepingcomputer.com/news/security/fbi-arrests-man-for-plan-to-kill-70-percent-of-internet-in-aws-bomb-attack>

Home > News > Security > FBI arrests man for plan to kill "70% of Internet" in AWS bomb attack

FBI arrests man for plan to kill "70% of Internet" in AWS bomb attack

By [Sergiu Gatlan](#)

April 9, 2021 04:29 PM 9



The FBI arrested a Texas man on Thursday for allegedly planning to "kill of about 70% of the internet" in a bomb attack targeting an Amazon Web Services (AWS) data center on Smith Switch Road in Ashburn, Virginia.

Seth Aaron Pendley, 28, was charged via criminal complaint on Friday morning for attempting to destroy a building using C-4 plastic explosives he tried to buy from an undercover FBI employee.

The FBI got wind of the suspect's plans in January when he revealed his plot on the MyMilitia website using a 'Dionysus' handle, a forum used by militia members and supporters to organize and communicate.

FBI agents found in late January from another source Pendley contacted using the Signal encrypted messaging app that he was planning to use C-4 plastic explosives to attack Amazon's data centers in an attempt to "kill off about 70% of the internet," according to the [criminal complaint](#) filed today.

18. According to screenshots of texts provided by CHS2 to FBI investigators, on February 19, 2021, **Pendley** showed CHS2 a handwritten list of addresses for AWS data centers, most of which were located in Virginia. On February 19, 2021, **Pendley** sent the following text to CHS2: "Oh yeah if I had cancer or something I would just drive a bomb into these servers lol." **Pendley** told CHS2 that **Pendley** hoped a successful attack could "kill off about 70% of the internet."

In February, Pendley shared with the source a hand-made map of Amazon's Virginia-based AWS data center "featuring proposed routes of ingress and egress at the facility."

The same source introduced the suspect to an undercover FBI employee posing as an explosives supplier at the end of March.

Pendley told the "supplier" that he was planning to bomb servers he believed were used by the FBI, CIA, and other US federal agencies. His plot also included taking down the "oligarchy" currently leading the United States.

23. **Pendley** further described his motive for the attack in words to the effect:

The main objective is to f*** up the Amazon servers. There's 24 buildings that all this data runs through in America. Three of them are right next to each other, and those 24 run 70 percent of the Internet. And the government, especially the higher ups, CIA, FBI, special sh**, they have like an 8 billion dollar a year contract with Amazon to run through their servers. So we f*** those servers, and it's gonna piss all the oligarchy off.

Undercover FBI employee hands suspect a dud

He met with the undercover FBI employee on April 8 to get what he believed were explosive devices. Instead, he received inert devices.

The suspect was arrested by FBI agents after placing them into his car, following a demonstration from the "explosives supplier" on how to arm and detonate them.

28. On April 8, 2021, **Pendley** again met with the UCE in Fort Worth, Texas. During the meeting, the UCE provided **Pendley** with devices the UCE explained were made of C-4 plastic explosives and detonation cord. The UCE showed **Pendley** how to arm and detonate the devices. **Pendley** took possession of the devices, and placed them in his car. At that time, **Pendley** was arrested by FBI investigators. In subsequent *Mirandized* statements, **Pendley** admitted to having orchestrated the plan to attack the AWS data centers. Investigators also executed a search warrant at **Pendley's** residence in Wichita Falls, in which they found hand-drawn maps, notes and flashcards relating to the planned attack; an AR receiver with a sawed-off barrel, along with the removed barrel piece; wigs and masks; a pistol which had been painted to look like a toy gun; paint cans; and a machete with the name "Dionysus" on the blade.

If he is found guilty of planning to blow up Amazon's Virginia data center, Pendley faces up to 20 years in federal prison.

"We are indebted to the concerned citizen who came forward to report the defendant's alarming online rhetoric," Acting US. Attorney Prerak Shah [said](#).

"In flagging his posts to the FBI, this individual may have saved the lives of a number of tech workers.

"We are also incredibly proud of our FBI partners, who ensured that the defendant was apprehended with an inert explosive device *before* he could inflict real harm."

Amazon shared the following statement with BleepingComputer about the planned attack.

"We would like to thank the FBI for their work in this investigation. We take the safety and security of our staff and customer data incredibly seriously, and constantly review various vectors for any potential threats. We will continue to retain this vigilance about our employees and customers." - An AWS Spokesperson.

Update 4/9/21 6:02 PM EST: Added statement from Amazon

Related Articles:

[Malware attack is preventing car inspections in eight US states](#)

[Brown University hit by cyberattack, some systems still offline](#)

[This bundle of 22 courses can help you build the perfect IT resume](#)

[US DOJ: Phishing attacks use vaccine surveys to steal personal info](#)

[CISA gives federal agencies 5 days to find hacked Exchange servers](#)

AMAZON

AWS

BOMB THREAT

INTERNET

USA



SERGIU GATLAN  

Sergiu Gatlan is a reporter who covered cybersecurity, technology, Apple, Google, and a few other topics at Softpedia for more than a decade. Email or Twitter DMs for tips.

| | |
|-----------------------------|--|
| | BTS SYSTEMES NUMERIQUES |
| | ENTREPRISE : <u>Agence CUBE</u> |
| | Période de stage du <u>8/02</u> au <u>5/03</u> |
| | Nom du STAGIAIRE : <u>Thibault PECH</u> |
| FICHE D'EVALUATION DE STAGE | |

Nom du TUTEUR, chargé de l'évaluation ci-dessous : François SOULARD

☞ Merci de cocher la case correspondant à votre appréciation

| | Très bien | Bien | Passable | Insuffisant |
|--|-------------------------------------|------|----------|-------------|
| Assiduité | <input checked="" type="checkbox"/> | | | |
| Ponctualité, respect des horaires | <input checked="" type="checkbox"/> | | | |
| Respect des règles internes (sécurité) | <input checked="" type="checkbox"/> | | | |
| Aptitude relationnelle et maîtrise de soi | | | | |
| - avec le tuteur | <input checked="" type="checkbox"/> | | | |
| - avec les collègues de travail | <input checked="" type="checkbox"/> | | | |
| Volonté, persévérance face aux difficultés | <input checked="" type="checkbox"/> | | | |
| Ouverture - Curiosité | <input checked="" type="checkbox"/> | | | |
| Autonomie (capacité à prendre des initiatives) | <input checked="" type="checkbox"/> | | | |
| Méthode, organisation | <input checked="" type="checkbox"/> | | | |
| Connaissances techniques | <input checked="" type="checkbox"/> | | | |
| Capacité à réaliser les tâches confiées | <input checked="" type="checkbox"/> | | | |
| Perception de l'évolution du stagiaire | <input checked="" type="checkbox"/> | | | |

Signature du Tuteur :



Cachet de l'Entreprise :

CUBE SARL au capital de 5000 €
RCS LA ROCHE SUR YON B 523 259 455 00021
99 boulevard d'Angleterre
85000 La Roche sur Yon
Tel 02 85 52 05 44 - www.cubecom.fr

