



RAPPORT DE STAGE

Stagiaire du pôle tourisme

BENATOUT Stéphane

Du 08 Janvier au 16 Février 2018

Tuteur en entreprise : Monsieur RUDELLE Nicolas (Directeur de Proxim IT Tourisme)

Maître de stage : Monsieur STEF Romain (Chef de projet)

Professeurs référents : Madame BROUGEOIS Agnès et Monsieur DEPPELEY Benoit

Établissement / Formation : Lycée Suzanne Valadon 39 rue François Perrin / BTS SIO

Entreprise d'accueil : Proxim IT 24 avenue du Président Wilson 87700 Aix sur Vienne

Table des matières

Remerciements	3
I.) Présentation de l'entreprise	3
II.) Présentation des besoins	4
III.) Description des travaux réalisés	4
IV.) Liste des logiciels et technologies utilisés	5
V.) Documentation technique	5
VI.) Conclusion	10
Figure 1 : création du tableau des informations.....	5
Figure 2 : fonction qui parcourt le tableau d'informations et affiche les paramètres.....	6
Figure 3 : modal	6
Figure 4 : fonction qui donne le type de la variable	7
Figure 5 : fonction de génération de formulaire	7
Figure 6 : fonction qui test les tableaux	8
Figure 7 : appel des fonctions.....	8
Figure 8 : fonction de test du formulaire partie 1	9
Figure 9 : fonction de test du formulaire partie 2	9
Figure 10 : traitement des informations pour l'API	10

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué au succès de mon stage et m'ont soutenu dans la rédaction de ce rapport.

Tout d'abord, je souhaite remercier chaleureusement mon maître de stage, M. Romain Stef, chef de projet au service Tourisme de Proxim IT, qui m'a permis d'obtenir ce stage et de découvrir de près le monde de l'entreprise. Cette expérience m'a motivé à poursuivre mon BTS et à envisager la poursuite de mes études après l'obtention de ce diplôme. Je le remercie pour son accueil, le temps passé ensemble, et le partage quotidien de son expertise. Grâce à la confiance qu'il m'a accordée, j'ai pu m'accomplir pleinement dans chacune de mes missions.

Je tiens également à exprimer ma reconnaissance envers mon tuteur de stage, M. Nicolas Rudelle, directeur du service Tourisme de Proxim IT, qui a accepté de m'accueillir au sein de son service. Je le remercie d'avoir interrompu ses congés spécialement pour signer ma convention de stage, témoignant ainsi de son engagement et de son soutien.

Je remercie chaleureusement toute l'équipe du service Tourisme pour leur accueil et leur esprit d'équipe. Leur soutien m'a permis d'apprendre et de comprendre de nombreuses choses essentielles.

Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage : ma famille, mes amis et l'équipe Tourisme.

I.) Présentation de l'entreprise.

J'ai effectué mon stage au sein de la société Proxim IT, créée en 2005, qui offre un large éventail de services dans le domaine des technologies numériques. L'entreprise propose des activités variées telles que l'infogérance, l'édition de logiciels et de progiciels, la création de sites internet, d'applications mobiles et web, le design graphique, la communication et le marketing, ainsi que le e-tourisme. Son expertise repose sur une démarche pragmatique, portée par un esprit créatif et une capacité d'adaptation constante aux évolutions de son environnement.

Proxim IT est organisée en quatre grands services :

- **Infogérance** : supervision réseau et hébergement,
- **SSII** : développement de logiciels et de progiciels,
- **Agency** : agence web, développement d'applications mobiles, design global et stratégie,
- **Tourisme** : logiciels de gestion touristique et opérations de promotion du territoire.

J'ai intégré le service Tourisme de l'entreprise, sous la tutelle de M. Nicolas RUDELLE, directeur de Proxim IT Tourisme, avec M. Romain STEF, chef de projet, comme maître de stage.

L'équipe est composée de Mme Élisabeth COMPÈRE, directrice produit, Mme Ludivine CHARRON, développeuse, et Mme Elodie GRENIER, chef de projet et consultante éditoriale, Monsieur Romain STEF, chef de projet et développeur, Monsieur Nicolas RUDELLE, directeur du service Tourisme.

II.) Présentation des besoins.

L'entreprise m'a confié une mission de développement web visant spécifiquement à améliorer une solution existante. Cette interface avait pour objectif d'optimiser les phases de recettage d'une API, en renforçant son efficacité et en facilitant le processus de validation.

III.) Description des travaux réalisés.

Dans un premier temps, j'ai effectué des recherches sur certaines technologies, telles que REST et les tâches CRON, utilisées dans une application web dédiée au test d'une API. Ensuite, j'ai configuré mon poste de travail en installant Windows 10 Professionnel sur l'ordinateur fourni par l'entreprise. J'ai également réalisé une rétro-ingénierie du code de la première version de la solution existante. Cela m'a permis de réfléchir à des améliorations

potentielles, que j'ai ensuite présentées au chef de projet pour approbation. Une fois la solution retenue, j'ai entamé son développement.

IV.) Liste des logiciels et technologies utilisés

J'ai utilisé le logiciel Sublime Text comme interface de développement, Microsoft Word pour toutes les rédactions, et Google Chrome comme navigateur web.

Les technologies employées pour l'optimisation des tests de l'API incluent la bibliothèque JavaScript jQuery, ainsi que Bootstrap, AJAX, JSON et PHP.

V.) Documentation technique

J'ai commencé par organiser toutes les informations dans un tableau. Ce tableau stocke les paramètres de base des routes qui nécessitent des paramètres pour fonctionner. Une route correspond à une fonction, une méthode ou une classe permettant d'effectuer des requêtes vers une API.

Une API (Application Programming Interface, ou Interface de Programmation Applicative) est un ensemble standardisé de méthodes, de classes et de fonctions, qui sert de façade à un logiciel afin de fournir des services à d'autres logiciels.

```
$delete_contact_remove_campaigns = array('nom' => 'delete_contact_remove_campaigns.php', 'data' => array('campaigns' => array ()));
$delete_contact_remove_criteria = array('nom' => 'delete_contact_remove_criteria.php', 'data' => array('criteria' => array ()));
$delete_remove_item = array('nom' => 'delete_remove_item.php', 'data' => array());
$delete_remove_reservation = array('nom' => 'delete_remove_reservation.php', 'data' => array());
array_push($routes, $post_contact_new_adress);
array_push($routes, $post_contact_new_campaigns);
array_push($routes, $post_contact_new_contact);
array_push($routes, $post_contact_new_criteria);
array_push($routes, $put_contact_edit_adress);
```

Figure 1 : création du tableau des informations

J'ai créé une fonction qui parcourt le tableau contenant les informations et affiche les paramètres nécessaires pour tester la route concernée.

```

function findRoute($routes){
    if(isset($_GET["route"])) {
        $myAskedRoute = $_GET["route"];

        foreach($routes as $param){

            if($param['nom'] == $myAskedRoute){
                echo json_encode($param['data']);
            }
        }
    }
}

```

Figure 2 : fonction qui parcourt le tableau d'informations et affiche les paramètres

Création d'une modal, une petite fenêtre pop-up personnalisable, qui est une fonctionnalité fournie par Bootstrap.

```

<div class="modal fade" id="<?php echo $compteur; ?>" tabindex="-1"
    role="dialog" aria-labelledby="exampleModallLabel" aria-hidden="
    true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModallLabel">Formulaire
                des données</h5>
                <button type="button" class="close" data-dismiss="modal"
                aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body" id="n<?php echo $compteur; ?>">

            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
                data-dismiss="modal">annuler</button>
                <button type="button" class="btn btn-primary" data-dismiss=
                "modal" onclick="verifForm(this, '<?php echo $inc.'
                ', '#result-".$compteur ?>')">Continuer</button>
            </div>
        </div>
    </div>
</div>

```

Figure 3 : modal

J'ai ensuite créé une fonction permettant de déterminer le type de variable à entrer pour tester la route appelée.

```

function typeVar($var){
    if(is_array($var)){
        return "array";
    }

    if(preg_match('/^\d{4}-\d{2}-\d{2}$/', $var)){
        return "date";
    }

    if(preg_match('/\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}/', $var)){
        return "date & heure";
    }

    if(is_bool($var)){
        return "boolean";
    }

    if(is_float($var)||is_int($var)||is_numeric($var)){
        return "numeric";
    }

    if(is_string($var)){
        return "string";
    }
}

```

Figure 4 : fonction qui donne le type de la variable

J'ai développé une fonction qui génère un formulaire.

```

function genForm($key, $params, $vartype){
    $form = "<tr><td>".$key."</td>";

    if($vartype == "string"){
        $form .= "<td><input type='text' name='".$key."' value='".$params."/></td></tr>";
    }

    if($vartype == "numeric"){
        $form .= "<td><input type='number' name='".$key."' value='".$params."/></td></tr>";
    }

    if($vartype == "boolean"){
        $form .= "<td><input type='checkbox' name='".$key."' value='".$params."/> Oui</td></tr>";
    }

    if($vartype == "date"){
        $form .= "<td><input type='date' name='".$key."' value='".$params."/></td></tr>";
    }

    if($vartype == "date & heure"){
        $date1 = $params;
        $date2 = explode(' ', $date1);
        $date = $date2[0]. 'T' . $date2[1];

        $form .= "<td><input type='datetime-local' name = '".$key."' step='1' value='".$date."/></td></tr>";
    }

    return $form;
}

```

Figure 5 : fonction de génération de formulaire

J'ai développé une fonction qui parcourt le tableau de paramètres pour identifier la présence éventuelle d'un tableau imbriqué. Si un tableau est

détecté, la fonction se rappelle elle-même (récursion) ; sinon, elle appelle la fonction qui génère le formulaire.

```
function testArray($tab, $form = null) {
    foreach ($tab as $key => $params) {
        $vartype = typevar($params);

        if($vartype == "array"){
            $form .= testArray($params);
        }

        else {
            $form .= genForm($key, $params, $vartype);
        }
    }

    return $form;
}
?>

<?php
$json = $_GET['data'];

if($json != ""){
    echo "<form class='form1'><table>";
    $data = json_decode($json, true);
    echo testArray($data);
    echo "</table></form>";
}
?>
```

Figure 6 : fonction qui test les tableaux

J'ai créé une fonction qui appelle toutes ces autres fonctions sans quitter la page actuelle, afin de ne pas perdre mes données. Pour cela, j'ai utilisé des appels AJAX.

```
<script>
function generationForm(url2,div,version){
    $.ajax({
        url: version+"/parameters.php?route="+url2
    }).done(function(donnees){
        $.ajax({
            url: version+"/genForm.php?data="+donnees,
            context: document.body
        }).done(function(form){
            document.getElementById(div).innerHTML=form;
        })
    });
}
</script>
```

Figure 7 : appel des fonctions

J'ai ensuite vérifié si mon formulaire existait, stocké mes données dans une variable, puis envoyé ces données à un fichier chargé de les transmettre à la base de données.


```

function verifForm($this,route,div){
    var donnees;
    var form = $($this).parents('.modal').find('.form1');

    if(form.find('input').length != 0){
        donnees = {};
    }

    form.find('input').each(function() {
        var name = $(this).attr('name');

        if($(this).attr('type') == 'checkbox'){

            if($(this).is(':checked')){
                donnees[name] = true;
            }

            else{
                donnees[name] = false;
            }
        }

        else{
            var value = $(this).val();
            donnees[name] = value;
        }
    });
    donnees = JSON.stringify(donnees);
}

```

Figure 8 : fonction de test du formulaire partie 1

```

    if(donnees !== undefined){
        $.ajax({
            url: route+"?data="+donnees,
            context: document.body
        }).done(function(result){
            $(div).append(result);
        });
    }

    else{
        $.ajax({
            url: route,
            context: document.body
        }).done(function(result){
            $(div).append(result);
        });
    }
}

```

Figure 9 : fonction de test du formulaire partie 2

Il faut ensuite traduire les données converties au format JSON en PHP afin de pouvoir les traiter conformément aux exigences de l'API.

```
$data = json_decode($_GET['data'], true);

$date1 = $data['reservation_datetime'];
$date2 = explode('T', $date1);
$date = $date2[0].' '.$date2[1];

$data['reservation_datetime'] = $date;
$data['status'] = intval($data['status']);
$data['budget'] = floatval($data['budget']);

$dataJson = json_encode($data);
```

Figure 10 : traitement des informations pour l'API

VI.) Conclusion

Pour résumer le projet, il m'a permis de découvrir de nouvelles technologies comme AJAX et JSON. J'ai également appris de nouvelles fonctions en PHP et en jQuery. Pour développer cette application, j'ai utilisé Sublime Text comme éditeur et WAMP comme serveur local.

Les plus grandes difficultés que j'ai rencontrées ont été liées à la rétro-ingénierie, car je ne connaissais pas la plupart des fonctions utilisées dans le code existant. La prise en main des appels AJAX a aussi été complexe en raison de la syntaxe qui m'était inconnue. Un autre défi majeur a été de parcourir le formulaire : bien qu'il ait été simple de localiser le formulaire, il a fallu trouver tous les inputs, les stocker dans un tableau, les convertir en JSON, puis les envoyer à la base de données.

En revanche, les fonctions permettant de générer les formulaires m'ont semblé relativement simples à réaliser. Il suffisait de récupérer le type de variable à enregistrer, d'afficher l'input correspondant, puis de le remplir avec la variable récupérée.