

Présentation de l'entreprise

Présentation :

Mustinfo est une entreprise conceptrice de logiciel dédié au vétérinaire pour les aider dans la gestion liée a leur métier. Ils ont conçu le progiciel Bourgelat qui permet au vétérinaire la gestion de rendez-vous, stock, fiche client, animaux en pension. Elle est liée à l'entreprise Medimust qui a développer le même progiciel mais pour médecin.

Contact :

- - Téléphone : 02 31 06 22 00
- - Adresse : 7 Rue Michel Cabieu, 14000 Caen
- - Site : [Bourgelat](#)

Rapport de Stage année n°2 (03/01 --- 10/07)

Semaine n°1 (03/01 --- 06/01)

Jour 1 : Mardi 03 Janvier

Programme du jour :

- Installation du matériel de travail (c'est à dire ordinateur(Mac), installation d'un OS à jour etc....)
- Revoir les lignes de code de l'année précédente, c'est à dire ce qui a été entrepris lors des deux derniers stages (Le premier projet basé sur un récapitulatif d'appels téléphonique sous php avec un protocole client-serveur et le second un récap appels beaucoup plus élaboré en se servant d'un cti, avec manipulation objet à l'aide d'ajax et node.js)
- Début du nouveau projet : récap' appel pour l'autre boite : c'est à l'identique de ce qu'il y été fait lors du premier projet sauf qu'il faut remplacer toutes les méthodes obsolète de l'API d'OVH par les plus récentes, donc un travail de recherche et d'auto-formation par rapport au nouvelles méthodes.

Poste maître

Statistiques

Nombre d'appels	Pris	Manqués	Attente Moyenne	Durée moyenne
91 -> 75% pris	68	23	108s	04:09m.

Technique entrant en cours

Code Client	Nom	Etat
Ajouter le code client	(0670584884) (Azad)	Repondu depuis 59:26 m., attente 00:41 m. 🔊 🔇

Personne	Appels entrants	Appels sortants	Status	Depuis	Durée moyenne Entrant	Durée moyenne Sortant	Durée total
	13	9			00:07:39	00:05:44	02:31:05
	6	2			00:05:06	00:04:15	00:39:08
	24	3			00:07:02	00:02:43	02:56:59
	14	6			00:01:47	00:05:22	00:57:18
	6				00:01:32	00:00:00	00:09:15
	2				00:00:59	00:00:00	00:01:59
	4				00:00:38	00:00:00	00:02:34
	16				00:04:20	00:00:00	01:09:26
	17				00:03:42	00:00:00	01:03:00
	5	6			00:13:29	00:05:11	01:38:36
	6				00:07:43	00:00:00	00:46:23
	9	9			00:01:34	00:01:16	00:25:43
	9	1			00:12:33	00:00:12	01:53:13
	84	53			00:11:36	00:06:46	22:14:29

Jour 2 : Mercredi 04 Janvier

Programme du jour :

- On continue sur le projet de duplication du récap appel pour l'autre boîte.
- Réussir à obtenir les numéros des clients via ovh api.
- Plusieurs tests établis avec l'API OVH au préalable pour voir si on réussi à obtenir le résultat voulu avant de le retranscrire en PHP.

Jour 3 : Jeudi 05 Janvier

Programme du jour :

- Obtention des données permettant l'affichage "live" des appels via Ovh api.
- Découverte de "cURL" sur terminal Mac OS, une interface qui passe par le terminal de ligne de commande pour récupérer une ressource présente dans un réseau informatique (source wikipedia).

```
curl -XPOST -H "X-Ovh-Application: B9NNydt4gXslpbNe" -H "Content-type: application/json" \
https://eu.api.ovh.com/1.0/auth/credential -d '{
  "accessRules": [
    {
      "method": "GET",
      "path": "/"
    }
  ],
  "redirection": "http://appels.mustinfo.com/romain/"
}'
```

Jour 4 : Vendredi 06 Janvier

Programme du jour :

- Continuation de la veille donc succession de test et résolution de bug (erreur de syntaxe, mauvaise méthode OVH).

Semaine n°2 (09/01 --- 13/01)

Jour 1 : Lundi 09 Janvier

Programme du jour :

- Migration du projet d'un serveur FTP vers un serveur SSH/SFTP.
- Tentative de finalisation du projet (échec suite à la succession de bug et de non résultat).

Jour 2 : Mardi 10 Janvier

Programme du jour :

- Finalisation du premier projet (réussite : c'est à dire que le résultat demandé par le tuteur a été obtenu, on obtient un récapitulatif d'appels fonctionnels).

Jour 3,4 et 5 : Mercredi 11, Jeudi 12, Vendredi 13

- #Jour3# Pour la suite du projet 1 j'ai eu besoin d'aller chercher des données dans une BDD sauf que gros soucis de sql avec les pages de codes qui étaient à la base faite pour une autre BDD, j'ai du refaire et adapter les requêtes SQL à la nouvelle BDD.
- #Jour4 et 5# Soucis avec la BDD c'est à dire des incohérence avec ce qui été demander (table manquante : normal ou pas etc...), donc étude complète de la BDD avec le tuteur.

Semaine n°3 (16/01 --- 20/01)

Jour 1 : Lundi 16 Janvier

- Tentative de résolution soucis BDD (réussite : j'utilisais une BDD non à jour et n'ayant pas de droit d'accès avec l'application web du projet 1).

Jour 3 : Mercredi 18 Janvier

- Début du second projet (Tuteur absent pendant la matinée, donc discussion seulement en début d'après-midi) .

Jour 4 et 5 : Jeudi 19 et Vendredi 20 Janvier

Absolument rien : attente de matériel pour continuer le second projet (nouvelle machine fonctionnant sous windows et Mise à niveau des connaissances langage informatique et autoformation pour la suite du projet (PHP, javascript, Angular et début de lecture d'un livre sur Symfony2).

Semaine n°4 (23/01 --- 27/01)

Toute la semaine entière en attendant le matériel pour le projet 2, j'ai créé un tableau (à la demande de mon tuteur) reprenant l'intégralité des appels manqués en intégrant une alerte lorsqu'un client est raté (l'appel) trop souvent sous PHP, donc manipulation une fois encore d'objet, avec l'intégration des méthodes de l'api d'OVH.

```

SidAppels = array();
$detailAppels = array();
$detailAppel = array();
$date = date('Y-m-d') . 'T' . date('H:i:s', strtotime('-120 minutes'));
//print_r ($detailAppels);
//$date=date('2015-05-20');
foreach ($listeTelephoneGroupe as $cle => $groupeNumero) {
  foreach ($groupeNumero as $numero) {
    $detailAppels[$cle][$numero] = $API->call('GET', '/telephony/ . $cle . '/service/ . $numero . '/voiceConsumption/?creationDatetime.from=' . $date); // Affichage détail appel ( l'appel
    foreach ($detailAppels[$cle][$numero] as $cle2 => $idAppels) {

      $detailAppels[$cle][$numero][$idAppels] = $API->call('GET', '/telephony/ . $cle . '/service/ . $numero . '/voiceConsumption/ . $idAppels); // Affiche détail de l'appel téléphonique
      //print_r($detailAppels[$cle][$numero][$idAppels]);
      if ($detailAppels[$cle][$numero][$idAppels]->duration < 20){
        unset($detailAppels[$cle][$numero][$cle2]); // Supprime SI les appels dure moins de 20 secondes.
      }
      $detailAppels[$cle][$numero][$idAppels]->number = $numero;
      //if ($detailAppels[$cle][$numero][$idAppels]->duration == 0)
      // {
      //   unset($detailAppels[$cle][$numero][$cle2]);
      $detailAppel[] = $detailAppels[$cle][$numero][$idAppels];
      //print_r ($detailAppel);
      // }
    }
  }
}

```

Semaine n°5 (30/01 --- 04/02)

Jour 1 : Lundi 30 Janvier

Je continue l'autoformation sur symfony2 (qui semble assez complexe au premier abord). Pour ça je me sers d'un livre fournis par mon tuteur ("Symfony2" de Alexandre Bacco) qui tire ces informations du Site du zéro. Je me renseigne sur d'autre site aussi pour avoir plusieurs explications différentes (plus ou moins formel) tels que wikipédia pour l'information brut et humaine du framework ou de OpenClassRoom pour l'information plus "informaticien".

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/etudiants/2016/romain.p/stage2>

Last update: **2019/08/31 14:21**

