

Nô—Jaffré Kylian BTS SIO SLAM

1^{er} année : Mise à jour de la fonctionnalité de demande de rappel du site web de l'entreprise

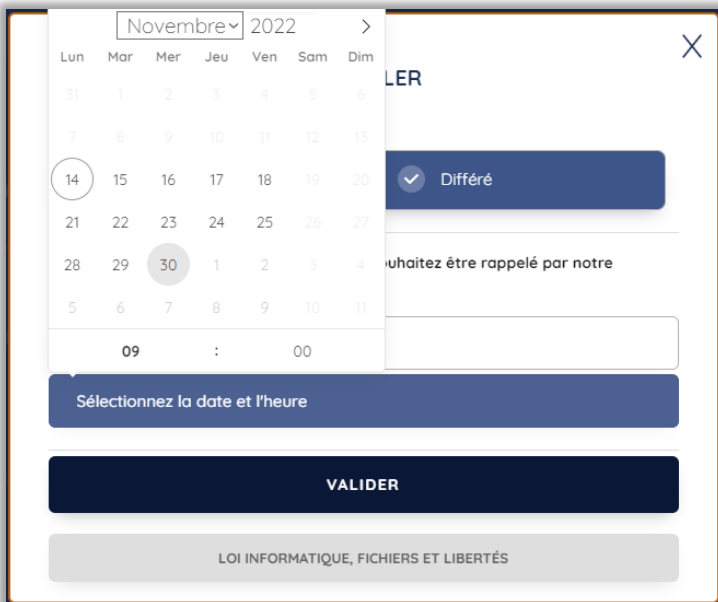
Sur le site web mascotte-assurances.fr se trouve un formulaire afin d'envoyer une demande pour être rappelé par un gestionnaire.

Tout d'abord, le formulaire comme vue par l'utilisateur ne permettait de voir clairement lorsque la demande était envoyée avec succès ce qui forcer parfois certains utilisateurs à envoyer un grand nombre de demande à la fois.

Enfin, le choix de la date et de l'heure de rappel ne subissait aucune vérification donc les utilisateur pouvaient demander d'être rappelés des jours et à des heures où l'entreprise était fermé.

J'ai tout d'abord changer utiliser la bibliothèque « Flatpickr » afin de donner au calendrier utilisé pour la sélection de la date un visuel en accord avec le cahier des charges du site web.

L'utilisation de « Flatpickr » me permet aussi d'intégrer le choix de l'horaire de rappel au calendrier ce qui me permettra de définir un format pour la date et l'heure que l'utilisateur n'aura pas besoin faire soi-même.



```
const config = {
  enableTime: true,
  allowInput: true,
  dateFormat: "d/m/Y H:i",
  time_24hr: true,
  minDate: "today",
  defaultHour: 9,
  minTime: "09:00",
  maxTime: "12:00",
  "locale": { "firstDayOfWeek": 1 },
  "disable": [DaysFerie],
  onValueUpdate: UpdateLimTime,
  onOpen: DisableInput,
  onClose: EnableInput,
  position: "above",
};
```

Il fallait maintenant ajouter certaines options afin que la sélection soit en accord avec les horaires d'ouvertures de l'entreprise :

- Enlever les weekends
- Enlever les jours fériés
- Rester entre 9h – 12h30 et 14h – 17h30

Enlever les weekends :

Flatpickr possède un attribut « disabled » empêche la sélection des jours de la semaine entrés en arguments, ici on utilise une fonction qui retournera la liste jours à désactiver, pour les weekends la bibliothèque date de Javascript les retourne avec la méthode `getDay()`

```
function JoursFerie(date) {  
    return (date.getDay() === 0  
        || date.getDay() === 6
```

```
const config = {  
    enableTime: true,  
    allowInput: true,  
    dateFormat: "d/m/Y H:i",  
    time_24hr: true,  
    minDate: "today",  
    defaultHour: 9 ,  
    minTime: "09:00",  
    maxTime: "12:00",  
    "locale": { "firstDayOfWeek": 1 },  
    "disable": [JoursFerie],
```

Enlever les jours fériés :

Les jours fériés ont été ajouté à la main dans une liste puis formater et retourner par la fonction `JoursFerie()` vue ci-dessus,

```
// le numéro de mois commence à 0 , ex : Janv = 0, Fev = 1  
ferie = ["1/0", "1/4", "8/4", "14/6", "15/7", "1/10", "11/10", "25/11"];
```

```
function JoursFerie(date) {  
    return (date.getDay() === 0  
        || date.getDay() === 6  
        || ferie.includes(date.getDate().toString().concat('/', date.getMonth().toString())))  
    );  
}
```

Certains jours fériés ont été exclus car trop compliqué à rajouter à la liste :

Calcul de la date de Pâques

Le calcul de la date de Pâques permet de déterminer le dimanche de Pâques et la date des fêtes associées comme l'Ascension et la Pentecôte. Pâques est le dimanche qui suit la première pleine lune du printemps, c'est-à-dire selon la définition établie par le Concile de Nicée en 325 : [Wikipédia](#)

Rester en 9h – 12h30 et 14h – 17h30 :

Flatpickr possède deux attribut qui permettent de choisir une heure minimum et maximum cependant cela ne permet pas de donner deux fourchettes horaires qui ne suivent pas.

J'ai donc utilisé un autre attribut qui permet d'effectuer une action lorsque que la valeur de l'heure est modifiée,

```
const config = {
  enableTime: true,
  allowInput: true,
  dateFormat: "d/m/Y H:i",
  time_24hr: true,
  minDate: "today",
  defaultHour: 9,
  minTime: "09:00",
  maxTime: "12:00",
  "locale": { "firstDayOfWeek": 1 },
  "disable": [JoursFerie],
  onValueUpdate: UpdateLimTime,
  onOpen: DisableInput,
  onClose: EnableInput,
  position: "above",
};
```

La fonction UpdateLimTime() permet à chaque modification de l'heure de vérifier si l'heure demandé est dans quelle fourchette horaire et de changer les heures minimum et maximum pour permettre la validation du formulaire.

```
function UpdateLimTime(selectedDates, updatedTime, obj) {
  var selectedDate = obj.latestSelectedDateObj;
  var selectedHour = selectedDate.getHours();
  // console.debug(obj);

  // Changement avec les flèches
  //=====
  if (selectedHour == 12 && lastArrow == "U" && lastHour == 12){
    obj.config.minTime = "14:00";
    obj.config.maxTime = "17:00";
    //On aurait pu faire plus facilement avec : arrowUp.click();
    var newDate = selectedDate;
    newDate.setHours(14);
    newDate.setMinutes(0);
    obj.setDate(newDate);
    lastHour = 14;

  }else if (selectedHour == 14 && lastArrow == "D" && lastHour == 14){
    obj.config.minTime = "09:00";
    obj.config.maxTime = "12:00";
    //On aurait pu faire plus facilement avec : arrowUp.click();
    var newDate = selectedDate;
    newDate.setHours(12);
    newDate.setMinutes(0);
    obj.setDate(newDate);
    lastHour = 12;

  }else{
    lastHour = selectedHour;
  }

  lastArrow = "";
}
```

```
datetime.hourElement.oninput = UpdateLimOnInput;
arrowUp.addEventListener("click", function(e){lastArrow = "U"});
arrowDown.addEventListener("click", function(e){lastArrow = "D"});
datetime.calendarContainer.style.zIndex = "99999999";
// console.log(datetime.calendarContainer);
```

Nô—Jaffré Kylian BTS SIO SLAM

1^{er} année : Mise à jour de la fonctionnalité de demande de rappel du site web de l'entreprise

Il fallait maintenant vérifier toutes les informations envoyées à notre serveur.

Numéro de téléphone :

Afin de vérifier le numéro de téléphone renseigné, j'ai utilisé une regex ,

```
$callnumber = "";
if (isset($_GET["callnumber"])) {

    $callnumber = $_GET["callnumber"];
    $callnumber = preg_replace('/\s+/', '', $callnumber); //enlève les espaces de $callnumber

    if (strlen($callnumber) == 11){$callnumber = '+'.$callnumber;}

    if (!preg_match('/^(?:\d{00}33|0)\s*[1-9](?:[\s.-]*\d{2}){4}$/ ', $callnumber)){ //vérifie le format du numéro de tel (fonctionne avec le +33 pour les psychopates)
        log_er('MauvaisFormatTel');
    }

}else{
    log_er('MauvaisFormatTel');
}
```

Date et heure :

```
$periode = "";

if (isset($_GET["periode"])) {

    $periode = $_GET["periode"];
    if (!preg_match('/^[0-9]{2}\D[0-9]{2}\D[0-9]{2,4}\D+[0-9]{1,2}\D[0-9]{2}$/ ', $periode)){
        log_er('MauvaisFormatPeriode');
    }

    $periode = preg_replace('/\s+\D*\s+/', ' ', $periode);
    $lperiode = explode(' ', $periode); //renvoie un array avec [0] = dd/mm/yyyy et [1] = hh:mm
    $date = preg_replace('/\D/', '/', $lperiode[0]);
    $horaire = preg_replace('/\D/', ':', $lperiode[1]);
    $heures = intval(substr($horaire, 0,2));
    $minutes = intval(substr($horaire, 3));

    if (!(in_array($heures, range(9,12)) or in_array($heures, range(14,17)))){log_er('MauvaiseHoraire');}

    if (!(in_array($minutes, range(0,59)))){log_er('MauvaiseHoraire');}

    if ($heures == 12 or $heures == 17){
        if ($minutes != 0){log_er('MauvaiseHoraire');}
    }

    $check = preg_split('/\D/', $date);
    $check = checkdate($check[1], $check[0], $check[2]);
    if (!$check) {log_er('MauvaisFormatPeriode');}

    $date = date_create_from_format('d/m/Y H:i', "$date $horaire");

    $ajd = new DateTime("now");

    if ($date < $ajd){log_er('MauvaiseHoraireBis');}
    if ($date->format('N') >= 6){log_er('MauvaisePeriode');}
    $ferie = ["01/01", "01/05", "08/05", "14/07", "15/08", "01/11", "11/11", "25/12"];

    if (in_array($date->format('d/m'), $ferie)){log_er('MauvaisePeriode');}

}

if ($periode != '' & $callnumber != ''){
    echo 'differe';
}else{
    echo 'immediat';
}
```

Nô—Jaffré Kylian BTS SIO SLAM

1^{er} année : Mise à jour de la fonctionnalité de demande de rappel du site web de l'entreprise

Gestion des erreurs :

Enfin j'ai rajouté des messages d'erreurs qui permettent à l'utilisateur de se corriger plus facilement et d'être sûr de l'envoi de sa demande,

```
// Poor Phone sanity check
if (raw_phonenumber.length > 0) {

    phone_status = true;

    // Poor period sanity check

    if (raw_period.length > 0) {period_status = true;}

    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            // console.debug(this.responseText.trim());
            var reponse = this.responseText.trim();
            var erreur = '';
            // console.log(reponse);
            switch (reponse) {
                case 'MauvaiseHoraire':
                    erreur = 'Nous sommes joignables de 9h à 12h et de 14h à 17h.';
                    break;
                case 'MauvaisePeriode':
                    erreur = 'Nous ne sommes pas disponibles les week-ends et jours fériés.';
                    break;
                case 'MauvaisFormatPeriode':
                    erreur = 'Le format de la date rentrée n'est pas valide, essayez de cette façon : jj/mm/aaaa à hh:mm .';
                    break;
                case 'MauvaiseHoraireBis':
                    erreur = 'Le créneau souhaité est déjà passé.';
                    break;
                case 'MauvaisFormatTel':
                    erreur = 'Le format de votre numéro de téléphone n'est pas valide.';
                    break;
                case 'NoSendedMail':
                    erreur = 'Une erreur interne est survenue merci de réessayer plus tard ou de nous appeler au 04.94.09.79.70.';
                    break;
            }

            if (erreur != '') {
                document.getElementById('ErFormRappel').style="color:red";
                document.getElementById('ErFormRappel').innerHTML = erreur;
                return;
            }

            document.getElementById('ErFormRappel').style="color:green";
            if (reponse == 'differe'){
                document.getElementById('ErFormRappel').innerHTML = 'Merci, un conseiller va traiter votre demande sur le créneau souhaité.';
            }
            if (reponse == 'immédiat'){
                document.getElementById('ErFormRappel').innerHTML = 'Merci de patienter un instant, un conseiller va traiter votre demande dans les plus brefs délais.';
            }
        }
    };
}
```