horizontal line

**ORG**

360&1

Full-performance

Synchronext

Web2Roi

Test technique

**Web**

# **Consigne**

Ce test technique est à réaliser en une semaine maximum. Internet est autorisé tant que le code utilisé peut être justifié et expliqué. Les consignes sont volontairement vagues afin de regarder vos approches. La qualité du code est également prise en compte. Si vous restez bloqué ou que vous ne comprenez pas une consigne, il vaut mieux poser une question (via [mathieu.bousse@360and1.com](mailto:mathieu.bousse@360and1.com)) que ne pas avancer. Ne pas réussir toutes les questions ne signifie pas un refus de la candidature, donc pas de stress 😄.

A la fin du test, merci de commit vos changements dans une branche à votre nom. Ce fichier doit également être commit après sa modification.

[**Consigne 1**](#_14mpx6a8znb7)

[**Vue / NodeJS (Express): 2**](#_2jwe26qwryx)

[1. Fonctionnalité: Implémentée sur la page nommée "Gestionnaires", les fonctionnalités suivantes: 2](#_jzhaf4hkqqo9)

[2. Résolution de bug: La page "Clients" ne fonctionne pas correctement. Deux bugs ont été signalés, c'est à vous de les repérer et de les rectifier. 2](#_j5fsiv1yw4vv)

[**PHP - Questions / Réponse: 2**](#_2558gkly53n)

[**Bonus 3**](#_67ldys70zv78)

# 

# 

# 

# **Vue / NodeJS (Express):**

## **Fonctionnalité:** Implémenter sur la page nommée "Gestionnaires", les fonctionnalités suivantes:

* Afficher la liste des employés (seulement les employés avec le rôle de "Gestionnaire"), cela peut prendre la forme que vous souhaitez (tableau, card, etc…) ainsi que leurs statistiques associés (nombre total de ventes, chiffre d'affaires total, en utilisant les clients qui leur sont associés).
* Implémenter un filtre pour afficher les employés par nom.

## **Résolution de bug:** La page "Clients" ne fonctionne pas correctement. Deux bugs ont été signalés, c'est à vous de les repérer et de les rectifier.

# PHP - Questions / Réponses:

Amélioration de code

Dans ces cas de figures, réécrire le code d’un extrait de php afin de l’améliorer. Il peut s’agir de le simplifier, corriger une faille ou l’optimiser. Le code de l’extrait n°1 n’est volontairement pas testable.

1.

|  |
| --- |
| ?php  // Si c'est un fichier pour tester en local c'est ok  // C'est un controller, on pourrait utiliser un framework pour gagner en lisibilité/sécurité.  // J'ai pas fait de class, mais c'est une bonne idée de le faire pour encapsuler la logique.  // Avec ce que j'ai fait, ce n'est pas simple de changer le code  function getAndValidateIds() {      $ids = isset($\_GET['ids']) ? $\_GET['ids'] : [];      if (!is\_array($ids)) {          die("IDs invalide.");      }      // on vérifie que tous les éléments sont numériques      $areAllIdValid = array\_reduce($ids, function($carry, $item) {          return $carry && is\_numeric($item);      }, true);      if (!$areAllIdValid) {          die("IDs invalide.");      }      return $ids;  }  // $ids = $\_GET['ids']; // il faut valider ce qui est passé dans $\_GET['ids'], sinon cela peut casser le code ou ouvrir la porte à des injection sql.  $ids = getAndValidateIds();    // Exposer directement les identifiants de connexion à la base de données n'est pas une bonne idée.  // Je ne suis pas certain, mais je pense qu'il vaut mieux les récupérer autrement  try {     $mysqlClient = new PDO('mysql:host=localhost;dbname=test;charset=utf8', 'root', 'azerty');  } catch (Exception $e) {      // Je ne suis pas certain, mais en production, ce n'est pas une bonne idée d'exposer les erreurs de connexion à la base de données. Plutôt la logger et afficher un message d'erreur.     Logger.error('Erreur : ' . $e->getMessage());     die('Erreur : Un problème est survenue');  }  // $result = [];  // foreach ($ids as $id) {  //    $sqlQuery = 'SELECT id FROM test where id = ' . $id . ' LIMIT 1'; // Injection SQL possible  //    $resStatement = $mysqlClient->prepare($sqlQuery);  //    $resStatement->execute();  //    $recipes = $resStatement->fetch();  //    array\_push($result, $recipes['id']);  // }  // Je ne suis pas certain de la syntaxe, mais je crois qu'on peut simplifier de cette façon  $sqlQuery = $mysqlClient->prepare('SELECT id FROM test where id = ?', ['(' . implode(',', $ids) .')']); // je ne suis pas sûr de la syntaxe  $sqlQuery->execute();  $result = $sqlQuery->fetchAll(\PDO::FETCH\_ASSOC);  header("Content-Type: application/json");  http\_response\_code(200);  var\_dump($result); |

2.

|  |
| --- |
| ?php  $array = array("lemon", "orange", "banana", "apple");  var\_dump(array\_reverse($array)); |

# 

# **Bonus**

Si vous avez terminé les tâches précédentes nous vous invitons à:

* Relire vos réponses et les améliorer si possible.
* Le code de la page principale "Dashboard" (section ‘**Vue / NodeJS (Express)**’) n'est pas aussi optimal qu'il pourrait l'être et mérite d'être amélioré. À vous de rechercher dans le code de l'API et de l'application tout ce qui peut l’être.

Attention à ce que tout soit bien fonctionnel à la fin du test!