

Demande initiale

Votre client vous demande de lui livrer ces fonctionnalités pour le 9 février 2012, avant 17h30. La date de livraison n'est pas négociable.

L'application à développer est un logiciel qui effectuera la validation du temps travaillé par des employés dans le respect des règles de l'entreprise.

Dans l'organisation du client, chaque employé doit remplir une feuille de temps hebdomadaire. Les employés doivent suivre plusieurs règles par rapport à leur temps travaillé qui sont dictées par l'entreprise. En fait, il y a tellement de règles à valider que les responsables de l'entreprise n'arrivent plus à s'y retrouver. Ils ont besoin d'un logiciel qui leur permettra de valider les règles de l'entreprise et qui générera les messages d'erreurs appropriés si une feuille de temps viole les règles.

Le logiciel ne possèdera pas d'interface utilisateur car il est destiné à être invoqué à partir d'une application web (le système d'approbation des feuilles de temps). Le contrat ne consiste donc qu'au développement du "back-end" de l'application.

Fonctionnalités

Le fichier d'entrée aura l'air de ceci :

```
{
  "numero_employe": 3884,
  "numero_semaine": 4,
  "annee": 2012,
  "lundi": [
    {
      "code_de_projet": 200,
      "minutes": 90
    },
    {
      "code_de_projet": 125,
      "minutes": 45
    },
    {
      "code_de_projet": 990,
      "minutes": 180
    }
  ],
  "mardi": [
    {
      "code_de_projet": 125,
      "minutes": 420
    }
  ],
  "mercredi": [
    {
      "code_de_projet": 125,
      "minutes": 420
    }
  ],
  "jeudi": [
    {
```

```

        "code_de_projet": 125,
        "minutes": 420
    }
],
"vendredi": [],
"samedi": [],
"dimanche": [
    {
        "code_de_projet": 990,
        "minutes": 30
    }
]
}

```

Le fichier de résultat généré par le logiciel devra ressembler à ceci :

```

[
    "L'employé n'a pas travaillé le nombre d'heures minimal.",
    "L'employé a dépassé le nombre d'heures de télétravail permis."
]

```

Le programme devra prendre ce fichier comme argument lors de l'exécution du logiciel dans une console. Le fichier où devra être placé le résultat devra également être spécifié à la console.

Exemple :

```
java TimeSheetValidator inputfile.json messages_result.json
```

Voici ce que vous devez savoir en premier lieu :

- Les codes de projet supérieurs à 900 sont des codes de télétravail.
- Si l'employé ne travaille pas durant une journée, elle est présente dans la feuille de temps mais vide, comme vendredi et samedi dans l'exemple précédent.
- Il est possible qu'un employé travaille la fin de semaine.
- Un employé dont le numéro est inférieur à 1000 est un employé de l'administration.
- Un employé dont le numéro est supérieur ou égale à 1000 est un employé normal.

Voici les règles que vous devez vérifier :

- Les employés de l'administration doivent travailler au moins 35 heures au bureau par semaine (excluant le télétravail).
- Les employés normaux doivent travailler au moins 40 heures au bureau par semaine (excluant le télétravail).
- Aucun employé n'a le droit de passer plus de 45 heures au bureau.
- Les employés de l'administration ne doivent pas faire plus de 10 heures de télétravail par semaine.
- Les employés normaux peuvent faire autant de télétravail qu'ils le souhaitent.
- Les employés normaux doivent faire un minimum de 6 heures au bureau pour les jours ouvrables (lundi au vendredi). Les employés doivent travailler même les journées de congé férié.
- Les employés de l'administration doivent faire un minimum de 5 heures au bureau pour les jours ouvrables (lundi au vendredi). Ils doivent également travailler les journées de congé férié.

Contraintes technologiques

Voici les contraintes que vous devez respecter :

- Le logiciel doit être développé avec le langage de programmation Java (JDK7).
- Il est impératif d'utiliser l'environnement de développement intégré NetBeans 7.01 ou plus.
- Les fichiers d'entrées doivent être des documents JSON.
- Les sources doivent être entreposées dans un dépôt GIT sous github.