

<p>420-329-SF</p> <p>Programmation de gestion</p> <p>Été 2018</p> <p>Pierre-François Léon</p>	<p>Examen 2</p> <p>Création d'un prototype ASP.Net MVC</p> <p>Saisie de temps</p>
---	--

Remarques

- Les points associés à chaque question sont écrits dans la section « Grille d'évaluation ».
- Lisez tout l'énoncé avant de répondre.
- Le nom du projet doit être « Devoir2Recettes »
- Toute communication entre élèves est interdite pendant l'examen. Si vous devez sortir pour une urgence pendant l'examen, demandez d'abord la permission au professeur.
- Le total des points de cet examen est 110%.
- Vous devez remettre votre code sur LEA avant 18h35 (mettez votre projet dans un zip).
- L'archive ne doit pas contenir de fichiers binaires ou temporaires
- Tout retard de plus d'une minute sera pénalisé de 5%. Tout retard de plus de 5 minutes ne sera pas corrigé. Assurez-vous donc de faire une première remise 10 minutes avant la fin.

Description

Le département des ressources humaines vous mandate pour démarrer le prototypage d'une application web permettant de saisir les temps travaillés des employés de l'entreprise. Chaque lundi, les employés remettent une feuille où ils ont inscrit des heures qu'ils ont travaillé. Un employé peut avoir plusieurs entrées pour une même journée. Chaque feuille ne compte que les temps de travail d'une semaine. Des adjoints vont saisir les feuilles de temps de chaque employé.

L'application doit permettre de gérer des employés : ajout, modification, détails **mais pas de suppression**.

À partir de la page de la liste des employés, les adjoints doivent pouvoir accéder à la liste des temps saisis pour un employé. La liste des temps doit proposer un bouton « ajouter du temps de travail ».

Un temps de travail correspond au temps passé par un employé à une journée donnée (Date). Un employé peut avoir plusieurs entrées pour une même journée (généralement des clients différents ou un longue pause).

Ce temps de travail doit donc être supérieur à 0 et inférieur ou égale à 24 (Certaines journées de projet sont dures !).

Avec votre architecte logiciel, vous avez convenu d'utiliser les structures suivantes :

```
public class Employe
```

```

{
    public int EmployeId { get; set; }
    public string Nom { get; set; }
    public string Prenom { get; set; }
    public DateTime DateEntreeEnFonction { get; set; }
}

public class TempsDeTravail
{
    public int TempsDeTravailId { get; set; }
    public int EmployeId { get; set; }
    public DateTime Date { get; set; }
    public string Client { get; set; }
    public double TempsEnHeures { get; set; }
}

```

Tous les champs sont obligatoires (Vous devez donc ajouter les annotations de données).

Vous devez modifier le message d'erreur pour avoir un message cohérent en français (ErrorMessage).

Quand on saisit une date, celle-ci ne doit pas comporter l'heure (DataType).

Les URLs suivantes doivent être respectées (Route) :

Entité	Opération	URL	Sécurité
Employe	Liste	/employe	Authentifiée
	Création	/employe/Create	Authentifiée
	Modification	/employe/Edit/{employeId}	Authentifiée
	Détails	/employe/Details/{employeId}	Authentifiée
	Stats	/employe/Stats	Authentifiée
TempsDeTravail	Liste	/employe/{employeId}/tempstravail	Authentifiée
	Création	/employe/{employeId}/tempstravail/Create	Authentifiée
	Modification	/employe/{employeId}/tempstravail/Edit/{tempsDeTravailId}	Authentifiée
	Détails	/employe/{employeId}/tempstravail/Details/{tempsDeTravailId}	Authentifiée
	Suppression	/employe/{employeId}/tempstravail/Delete/{tempsDeTravailId}	Authentifiée

La première version de l'application doit valider que les utilisateurs qui manipulent les entités « employe » et « tempsdetravail » **doivent être authentifiés**. Vous ne devez pas modifier la sécurité des pages préexistantes. Elle ne doit pas tenir compte des rôles des différents usagés.

Votre directeur veut une **page de statistiques**.

Elle doit comporter les statistiques suivantes :

- Moyenne du nombre d'heures travaillées par employé depuis le début du programme

- Moyenne du nombre d'heures travaillées par employé la semaine passée (Aidez-vous du code fourni en annexe)

Bonus

Durant la saisie d'un temps de travail, vous devez valider que le total des temps de travail saisis pour un employé à la même date ne dépasse pas 24 heure (Votre entreprise est éthique, elle ne facture pas des clients en double !). (À effectuer dans les actions associées aux POST des ajouts/modifications d'un temps de travail).

Remise

Le projet doit être remis sur Léa à l'heure demandée dans une archive au format zip.

Grille d'évaluation

La grille est donnée à titre indicatif pourrait être modifiée.

	Point	%	Total
Cacher les liens non accessibles en mode non authentifié	5		
Actions non authentifiées non accessibles	5		
Ajout des structures	5		
Ajout des entités dans le contexte EF	5		
CRU des employés	15		
Suppression de la page de suppression d'un employé	5		
CRUD des TempsDeTravails	10		
Saisie de la date sans heure	5		
Passage de l'idEmploye	10		
Respect des URLs	10		
Champs obligatoires	10		
Messages adéquats	5		
Page de stats - vue	5		
Page de stats - moyenne heures travaillés par employé	5		
Page de stats - moyenne heures travaillés la semaine passée	5		
Les journées ne dépassent pas 24h	5		
	110		

Semaine passée

```

public static class DateUtil
{
    public static Periode ObtenirPeriodeSemaineCourante(DayOfWeek
jourDebutSemaine = DayOfWeek.Monday)
    {
        return ObtenirPeriodeSemaine(DateTime.Now.Date, jourDebutSemaine);
    }
}

```

```

    }

    public static Periode ObtenirPeriodeSemainePrecedente(DayOfWeek
jourDebutSemaine = DayOfWeek.Monday)
    {
        return ObtenirPeriodeSemaine(DateTime.Now.Date.AddDays(-7),
jourDebutSemaine);
    }

    public static Periode ObtenirPeriodeSemaine(DateTime date, DayOfWeek
jourDebutSemaine = DayOfWeek.Monday)
    {
        var decalageJours = (int)jourDebutSemaine;

        DateTime debutSemaine = date;
        debutSemaine = debutSemaine.AddDays(-(int)debutSemaine.AddDays(-
decalageJours).DayOfWeek);

        return new Periode { DateDebut = debutSemaine, DateFin =
debutSemaine.AddDays(7).AddTicks(-1) };
    }
}

public class Periode
{
    public DateTime DateDebut { get; set; }
    public DateTime DateFin { get; set; }
    public override bool Equals(object obj)
    {
        var cmp = obj as Periode;

        return cmp != null && this.DateDebut == cmp.DateDebut && this.DateFin
== cmp.DateFin;
    }

    public override int GetHashCode()
    {
        var hashCode = 819881995;
        hashCode = hashCode * -1521134295 + DateDebut.GetHashCode();
        hashCode = hashCode * -1521134295 + DateFin.GetHashCode();
        return hashCode;
    }
}

```