

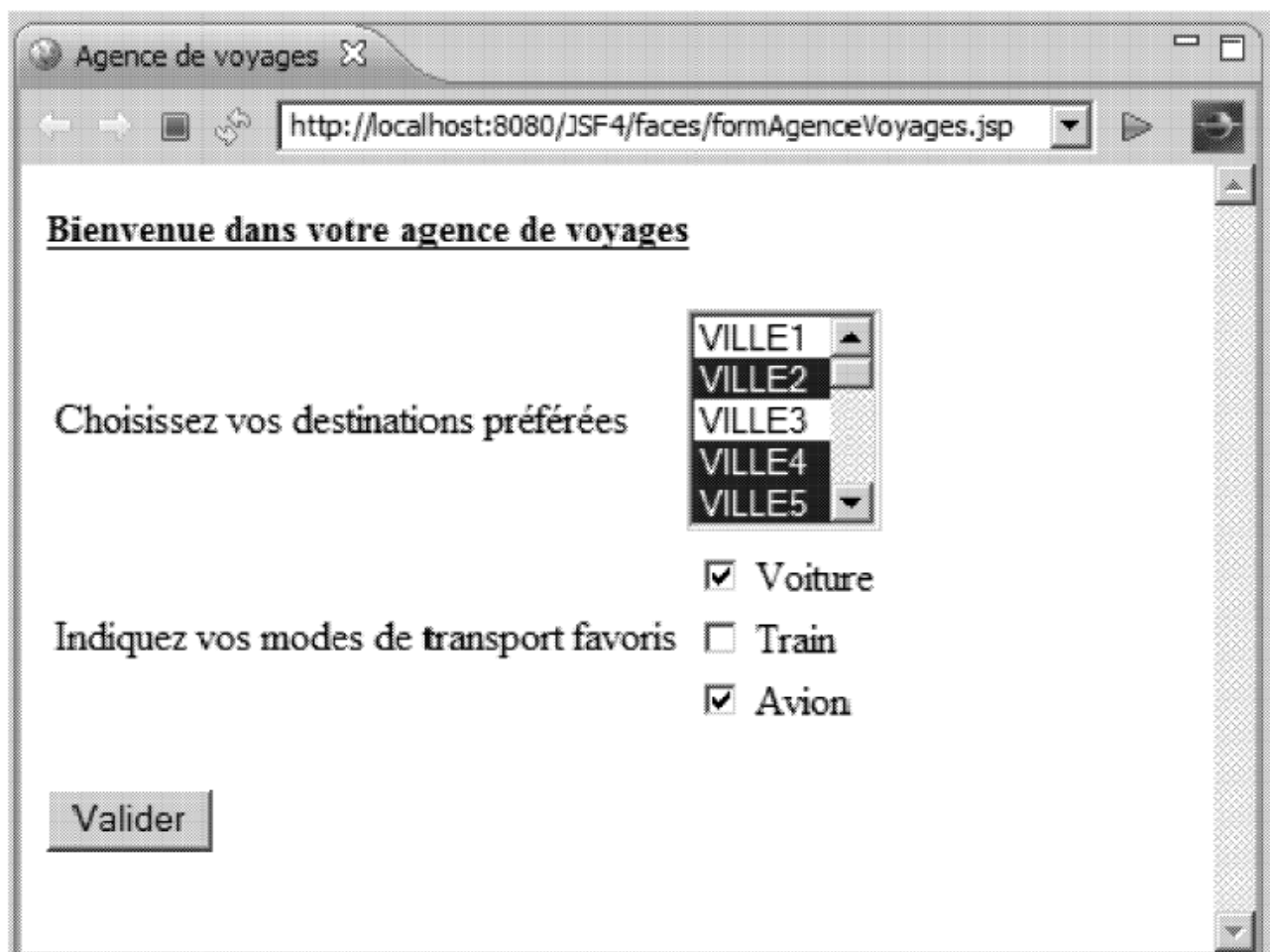
Exercice10 : les composants JSF 2.0

Etape1 :

Le but de cet exercice est de compléter les travaux qui ont déjà été abordés dans le cadre du troisième chapitre, concernant le traitement des formulaires de saisie.

Vous allez ici mettre en place un formulaire permettant à l'utilisateur de réaliser des choix multiples parmi des ensembles de valeurs prédéfinies, et vous développerez également tous les éléments qui assureront la prise en charge du formulaire en question.

La problématique de l'exercice concerne la conception de l'interface web d'une agence de voyages. Par le biais de cette interface, les internautes doivent pouvoir mentionner leurs destinations favorites et indiquer les modes de transports qu'ils préfèrent.



The screenshot shows a web browser window titled "Agence de voyages". The address bar displays "http://localhost:8080/JSF4/faces/formAgenceVoyages.jsp". The page content includes a heading "Bienvenue dans votre agence de voyages", a label "Choisissez vos destinations préférées" next to a list box containing "VILLE1", "VILLE2", "VILLE3", "VILLE4", and "VILLE5", a label "Indiquez vos modes de transport favoris" next to three checkboxes: "Voiture" (checked), "Train" (unchecked), and "Avion" (checked), and a "Valider" button.

Agence de voyages

http://localhost:8080/JSF4/faces/formAgenceVoyages.jsp

Bienvenue dans votre agence de voyages

Choisissez vos destinations préférées

VILLE1
VILLE2
VILLE3
VILLE4
VILLE5

Indiquez vos modes de transport favoris

☒ Voiture
☐ Train
☒ Avion

Valider

1. Créez un projet web dynamique nommé JSF4. Celui-ci sera utilisé pour l'ensemble des exercices de ce chapitre. Le projet en question devra permettre la prise en charge de la technologie JSF.
2. Concevez deux classes Java nommées `Ville` et `ModeDeTransport`, et rattachées à un package `jsf4`.

La première de ces classes sert à représenter le concept de ville : vous considérerez qu'une ville possède un identifiant numérique, un nom et un code postal numérique.

La classe `ModeDeTransport` sert à représenter le concept de mode de transport, qui se limite à deux propriétés : un identifiant numérique et un nom.

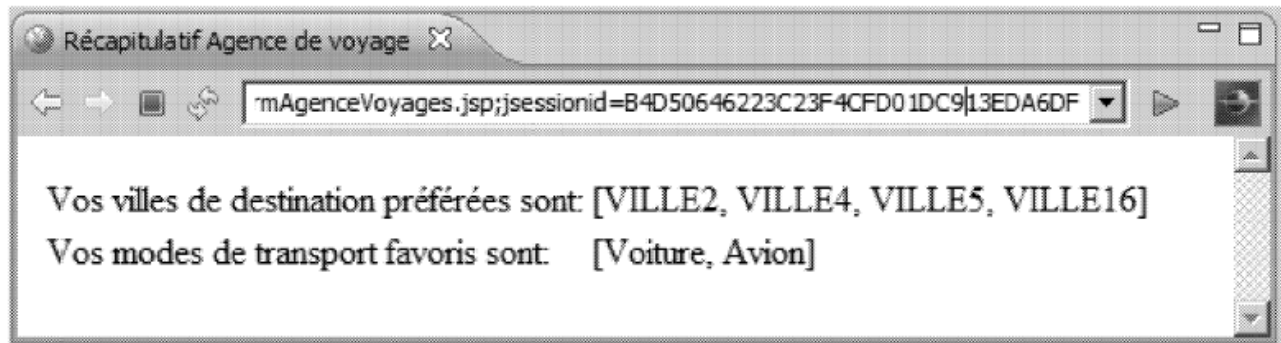
3. Développez une classe Java `BeanAgenceVoyages`, susceptible de pouvoir être exploitée par l'interface web par l'intermédiaire d'un bean managé. Dans un premier temps, cette classe doit permettre :
 - La gestion des villes proposées par l'agence de voyages.
 - La gestion des modes de transport proposés.
 - L'affichage des villes dans un composant JSF à choix multiples.
 - L'affichage des modes de transport dans un composant JSF à choix multiples.

Vous considérerez que cette classe se charge d'initialiser une liste de villes composée d'une vingtaine d'éléments. Les noms de villes sont générés de manière itérative par l'utilisation

d'un index numérique qui tient lieu d'identifiant pour la ville (VILLE1, VILLE2, ..., VILLE20). De même, le code postal est produit automatiquement de manière itérative à partir de l'identifiant de la ville (par exemple 75001, 75002, ..., 75020).

La liste des modes de transport, également initialisée par la classe BeanAgenceVoyages, se limite aux trois éléments représentés sur l'illustration précédente : Voiture, Train, et Avion. Chacun de ces éléments dispose bien entendu d'un identifiant numérique qui lui est propre.

4. Modifiez le fichier de configuration de l'application web pour déclarer un bean managé beanAgenceVoyages, de portée session, s'appuyant sur la classe BeanAgenceVoyages.
5. Créez la page web `formAgenceVoyage.jsp` permettant d'obtenir l'affichage correspondant à l'illustration précédente. La liste des destinations doit être présentée dans une zone de liste déroulante à choix multiples, alors que la liste des moyens de transport est fournie sous la forme d'un ensemble de cases à cocher.
6. Il faut maintenant permettre la mémorisation des informations saisies par l'utilisateur. Ce stockage de données doit se faire au sein même du bean managé beanAgenceVoyages. Modifiez en conséquence le code source de la classe BeanAgenceVoyages, en créant les propriétés et méthodes indispensables à la réalisation de cette opération.
7. Faites maintenant en sorte qu'un clic sur le bouton de soumission du formulaire provoque l'affichage des données saisies, dans une page récapitulative nommée `choixAgenceVoyages.jsp`. L'illustration suivante montre un exemple d'affichage possible.

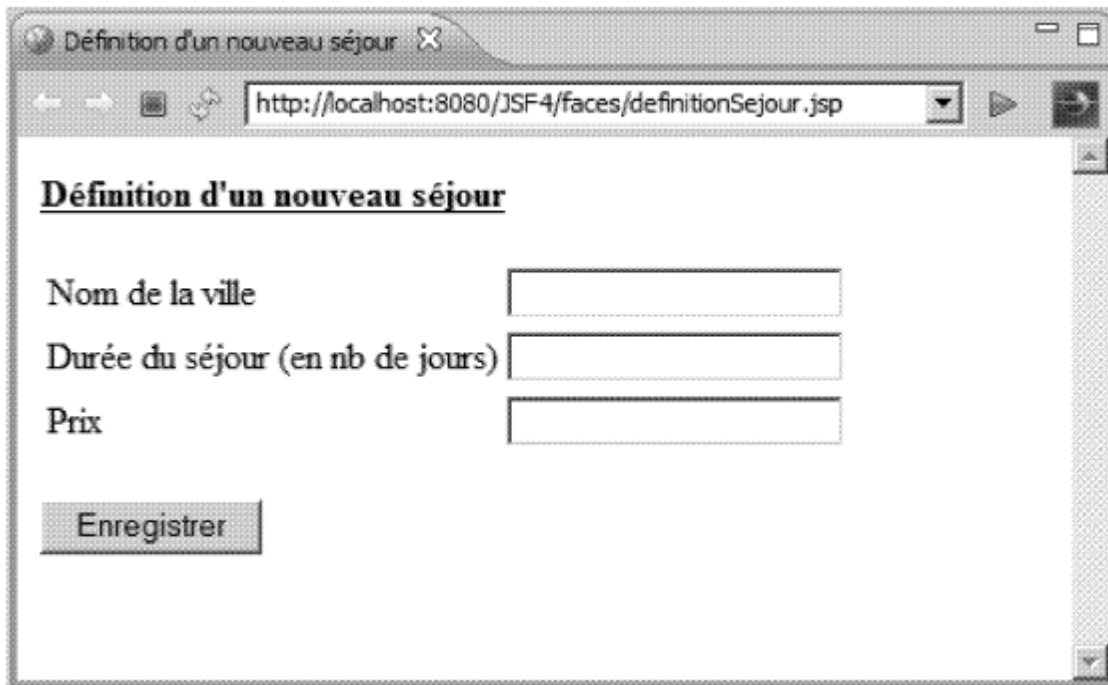


Etape2 :

Utiliser les trois types de validateurs standards proposés par l'implémentation JSF.

Le thème abordé ici est complémentaire à celui présenté dans l'exercice 4.1. Il s'agit de constituer une interface web permettant aux gestionnaires d'une agence de voyages de définir les caractéristiques d'un nouveau séjour. Ces caractéristiques, volontairement limitées, sont au nombre de trois : le nom de la ville correspondant à la destination, la durée du séjour, et le prix du séjour. Outre permettre l'affichage des champs de saisie, cette interface devra assurer les contrôles de validation adaptés, pour empêcher l'utilisateur de saisir des informations non conformes.

1. Développez une page `definitionSejour.jsp` présentant les différents champs de saisie requis. Pour que vous puissiez facilement tester la mise en œuvre des validateurs standards, ces champs devront tous être des zones de texte. Voici une illustration correspondant au résultat à obtenir.



Définition d'un nouveau séjour

Nom de la ville

Durée du séjour (en nb de jours)

Prix

2. En exécutant la page dans son état actuel, vous devriez constater qu'aucune action visible ne se produit lors de la validation du formulaire par un clic sur le bouton **Enregistrer**. Cette constatation est valable que vous saisissiez ou non des valeurs dans les champs ; que vous indiquiez des informations cohérentes ou non.

Améliorez la page pour que les contraintes suivantes soient opérationnelles :

- Obligation de fournir une donnée dans chacun des champs.
- Obligation d'indiquer un nom de ville dont le nombre de caractères est compris entre 2 et 30.
- Impossibilité de définir un séjour dont la durée est inférieure à 1 jour ou supérieure à 14 jours.
- Obligation de fournir un prix de séjour compris entre 50 et 200 euros.

3. Malgré les améliorations apportées, l'exécution de la page ne provoque manifestement aucune évolution visible en cas d'absence de saisie ou de fourniture de valeur inadaptée. En effet, bien que les différents contrôles soient mis en œuvre, aucun artifice n'a été prévu pour présenter à l'internaute les motifs d'un échec du processus de validation des données saisies.

Pour répondre à cette nécessité, associez à chaque composant de saisie un composant de type **Message**, et faites en sorte que celui-ci permette l'affichage d'un message d'erreur adapté, comme le montre l'illustration suivante :

Définition d'un nouveau séjour

http://localhost:8080/JSP4/faces/definitionSejour.jsp

Définition d'un nouveau séjour

Nom de la ville Merci d'indiquer un nom de ville.

Durée du séjour (en nb de jours) La durée du séjour doit être comprise entre 1 et 14 jours.

Prix Le prix doit être compris entre 50 et 200 euros.

Etape 3 :

Mettre en œuvre les convertisseurs standards pour vérifier le format des informations fournies dans un formulaire, et les rendre compatibles avec les propriétés d'un bean managé lié à ce formulaire.

