Le moteur de modèle hbs

1. Objectifs

- Être capable d'utiliser le moteur de modèle hbs dans l'application Node.js
- Connaître les boucles et les instructions conditionnelles dans hbs

2. Présentation

- **Handlebars.js** est un moteur de modèle JavaScript gratuit et open source qui nous permet d'injecter des valeurs de propriété JavaScript directement dans notre code HTML. Fondamentalement, il fonctionne comme un compilateur, ce qui signifie qu'il prend une entrée, puis la compile et la transforme en résultat.
- **Handlebars.js** est un moteur de création de modèles similaire au module jade de **node.js**, mais plus puissant et simple à utiliser. Il garantit un minimum de modèles et est un moteur sans logique qui maintient la vue et le code séparés.
- Il peut être utilisé avec express comme module **hbs**, disponible via **npm. HandleBars** peut être utilisé pour rendre des pages Web côté client à partir de données côté serveur.

3. Installation et Syntaxe de hbs

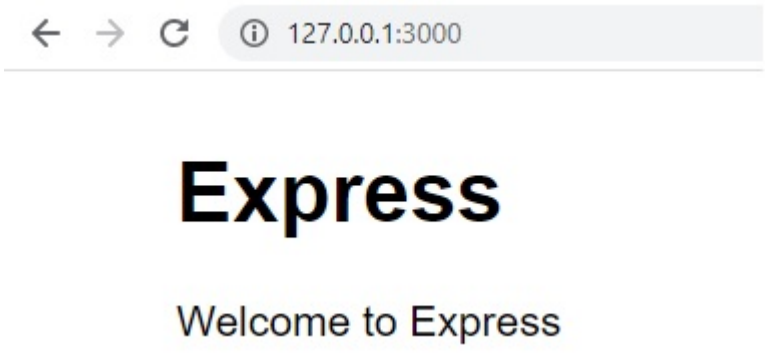
1. **Installation**
   - Installez hbs dans votre projet en utilisant la commande suivante :express test_hbs --hbs
2. Syntaxe de hbs

4. Utilisation

- L'utilisation de **hbs** comme moteur d'affichage par défaut ne nécessite qu'une seule ligne de code dans la configuration de votre application. Cela rendra les fichiers `.hbs` lorsqu'il `res.render` est appelé.
- `app.set('view engine', 'hbs');`
- Une fois la commande précédente à été exécuté avec succès , exécutez la commande `npm start` pour démarrer le serveur.
- Si vous avez installés `nodemon`, exécutez la commande : `nodemon`

apcpedagogie.com

- Si vous accédez maintenant à partir de votre navigateur `localhost:3000`, votre serveur est en cours d'exécution et vous pouvez voir la page d' accueil express . Cela signifie que tout fonctionne et est installé correctement.



- Si vous vérifiez le dossier **views** qui a été installé, vous pouvez voir les trois fichiers `.hbs` créés.

- **layout.hbs:**

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    <link rel='stylesheet' href='/stylesheets/style.css'
/>
  </head>
  <body>
    {{{body}}}
  </body>
</html>
```

  - Le fichier **layout.hbs** contient le squelette **HTML** de base et les autres fichiers de vue utiliseront ce squelette de base, sauf indication contraire. Ils rendront leur contenu spécifique dans le `{{{body}}}`. C'est l'endroit où les autres vues entreront et rendront leur contenu.

- **index.hbs:**

apcpedagogie.com

```
<h1>{{title}}</h1>
<p>Welcome to {{title}}</p>
```

- Cette page obtient sa sortie de titre du fichier **index.js**. `{{ }}` est utilisé pour afficher une variable passée du **nodejs - express** à la vue à restituer à l 'écran.
- Tout ce contenu est finalement rendu dans le fichier `layout.hbs`.
- Si vous remplacez le titre "Express" par "apcpedagogie.com" dans le fichier **index.js** et que vous redémarrez le serveur. votre localhost: 3000 ressemble maintenant à ceci:

5. **error.hbs:**

6. **routes/index.js:**

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'apcpedagogie.com' });
});
module.exports = router;
```

7. La fonction de rendu de ce fichier indique de renvoyer une réponse au navigateur contenant le code HTML du fichier d'index et **hbs** est ajouté car le guidon est configuré comme moteur de visualisation par défaut.
8. Mais avant d'envoyer la réponse, il remplace également toutes les variables à l'intérieur de la syntaxe `{{ }}` par les données appropriées passées comme deuxième argument dans cette fonction de rendu, ce n'est en fait rien d'autre qu'un objet javascript.
   Ainsi, la valeur de la variable **title** ici remplacera la variable **title** dans le fichier **index.hbs**.

9. ## Les Boucles et conditions

   - L'instruction `#if` utilise une expression booléenne dans laquelle, si l'expression booléenne est vraie, alors le bloc de code à l'intérieur de l'**instruction if** sera exécuté; si l'expression booléenne est fausse, alors le bloc `else` sera exécuté.

   - **Syntaxe**

apcpedagogie.com

```
{{#if property-name}}
   //statement
{{else}}
   //statement
{{/if}}
```

10. **Exemple**

    - L'exemple donné ci-dessous montre l'utilisation de l'assistant conditionnel **if**.

```
{{#if check}}
   //true block of statement
   <h3> boolean value is {{check}}</h3>
   {{else}}
   //false block of statement
   <h3>boolean value is {{check}}</h3>
{{/if}}
```

11. ## Activité

    - Utilisez Bootstrap 5 pour ajouter une barre de navigation à toutes les pages de votre projet.
    - **Travail à faire**
      - Incluez le CDN Bootstrap 5 de leur site Web officiel et l'ajouter à votre **layout.hbs**.
      - Créez un dossier appelé "**partials**" à l'intérieur du dossier **views**
      - Créez dans le dossier crée précédemment un nouveau fichier appelé **header.hbs**.
      - Dans le fichier **layout.hbs**, incluez le code suivant dans le fichier **>body<**: `{{> header}}`

Sommaire du cours Node.js

Source : https://apcpedagogie.com/le-moteur-de-modele-hbs · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com