

J2EE et MVC

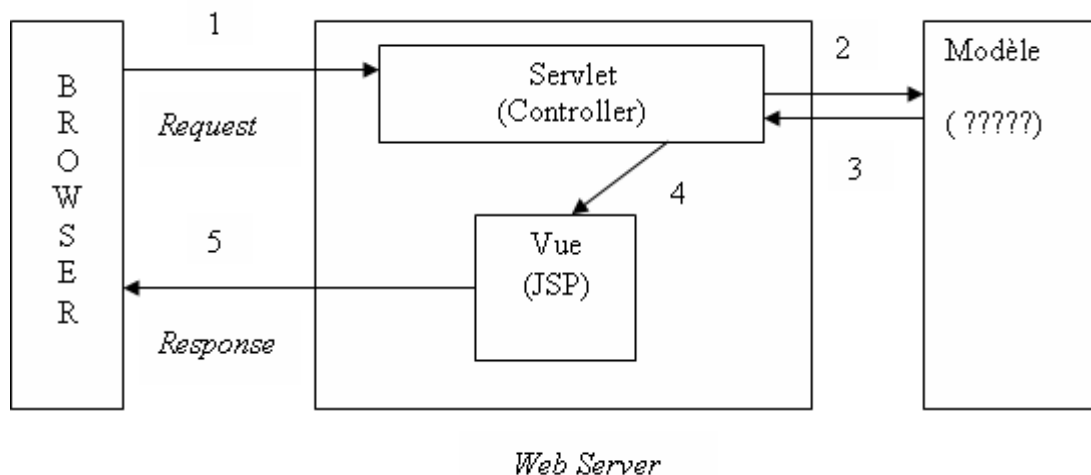
Nous avons vu deux techniques différentes mais équivalentes de génération de pages html : les servlets et les JSP. Même si il est possible d'appeler directement une page JSP ceci est contraire au Design Pattern (modèle de conception) MVC2 – actuellement fortement préconisé – qui impose de mettre en œuvre l'architecture Modèle – Vue – Contrôleur.

MVC

Le modèle est constitué de l'implémentation des données et des règles métiers qui les dirigent (dans l'exemple de la bibliothèque vu précédemment en ALG2, le package businesslogic joue ce rôle).

Le contrôleur (notion abordée dans la première partie de l'année en ALG2) a pour rôle de traiter les requêtes de l'utilisateur : c'est lui qui juge de la validité de la requête et détermine quelles fonctionnalités du modèle sont concernées et quelles sont les vues qui sont à fournir en réponse. Le contrôleur appelle la fonctionnalité du modèle qu'il juge devoir mettre en œuvre et transmet à la vue choisie les données nécessaires (généralement, sous J2ee, sous forme de JavaBeans).

Schéma de MVC:



Contrôleur : gère la validité des requêtes, détermine les fonctionnalités 'métier' à appeler et les vues à afficher. C'est lui qui gère la cinématique de l'application. Un tel contrôleur sera implémenté en JEE au travers d'une servlet.

Modèle : gère la logique métier

Vue : gère la logique de présentation. Une telle vue est généralement implémentée en JEE par une JSP.

Le modèle peut être implémenté de différentes manières et notamment sur un ou plusieurs serveurs (locaux ou distants) qui eux-mêmes peuvent accéder à un ou plusieurs serveurs (locaux ou distants) de données persistantes.

MVC2

MVC s'étant révélé fastidieux à mettre en oeuvre (pour chaque type de requête on devait définir un nouveau contrôleur) MVC2 fut préconisé : un contrôleur (une servlet) se charge de la redirection d'un ensemble de requêtes. Un tel contrôleur est souvent appelé FrontController).

Les Frameworks de mise en oeuvre de MVC2

La bonne conception, la validité, la maintenabilité des applications Web constituant actuellement un enjeu majeur, de nombreux outils ou frameworks existent dans le monde JEE. Les plus utilisés sont Struts et Java Server Faces (JSF).

EXEMPLE DE FONCTIONNEMENT

On peut mettre en oeuvre MVC2 de diverses manières, nous allons en exemplifier une. Imaginons que nous ayons un ensemble de requêtes gérées par le même FrontController accessible au travers de l'url «/FC». Les requêtes devront fournir au contrôleur un paramètre spécifiant l'action qu'il doit gérer: appelons ce paramètre « action ».

Le contrôleur devra d'abord déterminer l'action requise et transférer le traitement à sa méthode privée adéquate qui devra ajouter au request les attributs nécessités par les pages à afficher (au travers d'accès au modèle) et donner l'url de la page appelée (éventuellement une page d'erreur pour certaines exceptions). Il est à remarquer que les pages appelées, ne pouvant pas être directement référencées par le browser, seront créées dans un sous-répertoire de WEB-INF.

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    String page=""; //variable qui reprendra l'url de la page à appeler
    String action=request.getParameter("action");
    if (action==null){
        page="WEB-INF/Jeux/Accueil.jsp";
    } else {
        if(action.equals("action1")){
            page=gereAction1(request,response);
        }else{
            if(action.equals("action2")){
                page=gereAction2(request,response);
            }
            ...
            else{
                page=gereActionNonDefinie(request,response);
            }
        }
    }
    //appel de la page à afficher
    request.getRequestDispatcher(page).forward(request, response);
}

private String gereAction1(HttpServletRequest request, HttpServletResponse response){
    String page="WEB-INF/.../jspAction1.jsp";
    ...
    ...

    try{
        ...
    }catch (...){
        ...
        page="WEB-INF/.../errorXX.jsp";
    }

    return page;
}

...
```

En travaillant ainsi nous respectons MVC et donc découplons la logique de présentation du modèle.

EXERCICE

Mettons ceci en œuvre au travers de eVente.

Toutes les pages de l'application devront offrir un menu permettant d'accéder à la liste des promotions en cours, la liste des marques distribuées et au catalogue du site.

La page d'accueil de l'application devra présenter le site en plus de présenter le menu.

Liste des **promotions** en cours: présentation des produits (avec photo, marque, prix normal et promotion, catégorie, rayon, ...).

Marques distribuées: liste, ordonnée alphabétiquement, des marques.

Catalogue: Présente un menu (à gauche) reprenant la liste des catégories et une comboBox reprenant la liste des rayons de la catégorie sélectionnée (si aucune sélection n'est faite, la première catégorie (au sens alphabétique) est sélectionnée (idem pour les rayons). La liste des produits (actuellement disponibles) de la catégorie et du rayon sont affichés par ordre croissant de prix.

Toutes les requêtes, sauf l'accès à l'accueil seront gérées par le même contrôleur.

La logique de présentation sera gérée par des pages jsp, la présentation elle-même par css.