

Commentaire

La formation s'appuie par défaut sur l'IDE Eclipse et le serveur d'applications Apache Tomcat. La formation peut être réalisée avec un autre IDE (NetBeans, IBM WSAD/RAD) et un autre serveur d'applications (JBoss, GlassFish, WebSphere, WebLogic).

Réf : T462-020

Durée : 5 jours

Public

Développeurs Java SE.

Vous serez capable de

Concevoir une application Web basée sur Java EE.
Utiliser un environnement de développement pour concevoir et déboguer une application Web Java EE respectant le modèle MVC.
Packager, déployer et configurer une application Web Java EE sur un serveur d'applications.

Pré-requis

Avoir impérativement suivi le cours « T443-010 - Programmation Java (Java SE) - niveau 1 », ou avoir les connaissances équivalentes.
Connaissance des technologies de l'Internet et du langage HTML.

CONTENU PEDAGOGIQUE

Architecture de la plate-forme Java EE

- De l'architecture client/serveur à l'architecture Web
 - Les composantes d'une architecture Web
- Le modèle de composants Java EE
 - Servlet, JavaServer Pages (JSP), Enterprise JavaBeans (EJB)
- La plate-forme de services Java EE
 - Accès aux ressources et intégration dans le système d'information
 - JDBC, JNDI, JTA, JCA, JavaMail, ...
- Structure des modules et applications Java EE
 - Packaging et déploiement
- Les serveurs d'applications Java EE
 - Rôles et principes de fonctionnements
 - Les conteneurs du serveur
 - L'offre du marché

Conception d'applications Web Java EE

- Présentation du modèle de conception MVC
 - Attribution des rôles aux composants Java EE
- Répartition des composants applicatifs dans les différentes couches logicielles
 - Notion de couplage et d'interdépendance

Utiliser un IDE pour le développement Web Java EE

- Installer et configurer un IDE pour le développement Java EE
 - Configurer une instance de serveur d'applications pour tester l'exécution des composants Web
 - Déboguer les composants Web

Développement de la couche d'accès aux données

- Présentation de l'API JDBC (Java DataBase Connectivity)
- Notion de pilote/fournisseur JDBC
 - Présentation des différents types de pilotes JDBC
 - Utiliser et intégrer un pilote JDBC dans son projet Java sous Eclipse
- Optimiser l'accès aux bases de données avec les pools de connexions et l'objet `javax.sql.DataSource`
 - Configurer la ressource JDBC au niveau du serveur d'applications
 - Utiliser l'API JNDI pour obtenir une référence sur le pool de connexion
 - Récupérer une connexion JDBC disponible dans le pool
- Utilisation des classes du package JDBC standard : `java.sql`
 - Exécuter des requêtes SQL au travers des différents types d'interfaces (Requêtes simples, pré compilées, procédures stockées)
 - Exploiter les résultats
- Méthodologie pour l'écriture des programmes d'accès aux bases de données
 - Gestion efficace des exceptions

- Libération des ressources
- Les transactions
 - Utiliser les transactions JDBC
 - Présentation du concept de transactions distribuées

Développement de Servlets

- Présentation de l'API Servlet
 - Les classes et interfaces des packages `javax.servlet` et `javax.servlet.http`
- Les méthodes du cycle de vie d'une Servlet : `init()`, `destroy()`
- Les méthodes de traitements d'une Servlet : `doGet()`, `doPost()`, ...
- Manipulation des objets représentant les requêtes et les réponses HTTP
- Récupération des paramètres des requêtes HTTP
- Maintenir le contexte utilisateur avec les cookies et les sessions HTTP
 - Problème du suivi de session sans cookie : mise en place de la réécriture d'URL et encodage des URL
- Bonnes pratiques d'utilisation des contextes de stockage de données (application, session, requête, page)
- Création et utilisation de composants `JavaBean` et mise en œuvre du pattern Value Object pour le transfert de données métiers en les différentes couches d'une application Java EE
- Interaction avec d'autres ressources Web
 - L'inclusion de ressources avec la méthode `include()`
 - La redirection serveur avec la méthode `forward()`
 - La redirection client avec la méthode `sendRedirect()`

Développement de JSP

- Structure et syntaxe d'une JSP (JavaServer Pages)
 - Les directives
 - Les scriptlets et les expressions
 - Les tags actions
 - Les objets implicites
- Bonnes pratiques de conception des JSP
- Récupération de données de composants `JavaBean` dans les JSP
 - Manipulation des contextes de stockage de données (application, session, requête, page)
- Inconvénient de la coexistence HTML/Java dans les JSP et bonne pratique dans la mise en œuvre de la bibliothèque de balises JSTL (Java Standard Tag Library)
 - Les balises et le langage d'expression

Développement de classes de filtres

- Principe de fonctionnement et cas d'utilisations
- Création de classes de Filtres

Développement de classes listeners

- Principes de fonctionnement et cas d'utilisations
- Création de classes listeners sur les contextes de session et d'application

Configuration et paramétrage d'une application Web Java EE

- Le descripteur de déploiement : le fichier web.xml
- Déclaration de la page d'accueil par défaut
- Déclaration et paramétrage des :
 - Servlets
 - JSP
 - filtres
 - listeners
- Déclaration de la page de gestion des erreurs par défaut
- Déclaration de la durée de vie des sessions HTTP
- Déclaration des ressources sécurisées et des rôles applicatifs

Packaging et déploiement des applications Web Java EE

- Création de l'archive Web Java EE pour le déploiement
 - Utiliser l'IDE ou Apache Ant pour le packaging en fichier WAR
- Utilisation de la console d'administration du serveur d'applications pour configurer le déploiement de l'application Web Java EE
- Déployer l'application Web Java EE à chaud et/ou à froid