

Langages et Environnements Évolués

Sommaire des cours

25 octobre 2018

Il n'y a pas de poly pour ce cours (le contenu change trop souvent pour rendre un poly très utile). Voici tout de même le plan prévu.

Souvenez-vous : ce cours n'a pas pour vocation de faire de vous des expert dans aucune de ces technologies, mais de vous présenter quelques uns de leurs aspects, de vous permettre de les évaluer, et surtout de vous offrir un premier contact avec des outils professionnels.

1 Cours 1 : Introduction

1.1 Modalités du cours

1.2 Préoccupations du développeur

1.2.1 Frontend

1.2.2 Backend

1.2.3 Persistance

1.2.4 Déploiement

1.2.5 Organisationnel

1.2.6 Débug

2 Cours 2 : Déploiement

2.1 Intermède : Gestionnaire de version (Git)

2.1.1 Modèle de Gestionnaire de version

2.1.2 Intérêt

2.2 Introduction au déploiement

Déploiement d'une application distribuée : casse tête irrésoluble.

2.3 Serveur d'application

Programme conçu en une pièce puis découpé et dispatché par le serveur sur plusieurs machine.

2.3.1 serveur d'application = serveur JEE

2.3.2 Servlets et JSP

2.3.3 EJBs

2.4 Docker

Micriservices conçus indépendamment avec connection distante par défaut.

2.4.1 Virtualisation légère

2.4.2 Images et conteneurs

2.4.3 Réseau docker

3 Cours 3 : Persistance

3.1 Types de persistance

Conserver les données : un traid-of dépendant du cas d'utilisation (criticité, taille, temporalité, efficacité, ressources).

3.1.1 Bdd relationnelles

3.1.2 Bases de données non-relationnelles

3.1.3 Stockage de fichiers

3.1.4 Mémoire dynamique

3.2 Persistance et Déploiement

Solutions de persistance en environnement distribué.

3.2.1 Serveur d'application

3.2.2 Docker

3.3 ORMs

Utiliser un intermédiaire pour alléger l'interface backend-Bdd : cas des frameworks de mapping objet-relationnel

3.3.1 Principe

3.3.2 Cycle de vie d'une entité, paresse

3.3.3 Langages de requête d'ORM

3.3.4 Liaisons $n:1$ et $n:n$

3.3.5 Héritage

4 Cours 4 : Organisation du code

4.1 Intégrer du code communautaire

4.1.1 Bibliothèques

4.1.2 Génération de code

4.1.3 Inversion de contrôle

4.1.4 Boots

4.2 Builder et IDE

4.3 Spring

4.3.1 Injection de dépendance

4.3.2 Programmation pas aspect (AOP)

4.3.3 Spring MVC

(seulement survolé en cours mais revu en TP)

4.4 Architectures de programmes

4.4.1 Architecture MVC

4.4.2 Architecture n-tier

4.4.3 L'architecture JEE "classique" est une archi n-tier