



Systeme d'information et Framework de nouvelles générations

NOUREDDINE ANNITRA

Introduction



Objectif de la présentation

Importance de choisir les bons outils et Frameworks

Architecture Monolithique

- Spring Framework
- Java EE / Jakarta EE
- Hibernate
- Struts
- Grails
- Spring Batch

Spring Framework

- Avantages :
 - Très flexible et modulaire
 - Large écosystème de projets et support Communautaire
- Inconvénients :
 - Courbe d'apprentissage élevée
 - Configuration initiale peut être complexe

Java EE / Jakarta EE

- Avantages :
 - Standardisé et mature
 - Bonne intégration avec les serveurs d'applications
- Inconvénients :
 - Peut être lourd et verbeux
 - Moins flexible comparé à d'autres frameworks modernes

Architecture Microservices

- Spring Boot
- Dropwizard
- Micronaut
- Vert.x
- Quarkus
- Helidon

Spring Boot

- Avantages :
 - Démarrage rapide avec une configuration minimale
 - Large écosystème de projets Spring
- Inconvénients :
 - Peut devenir complexe pour de grands projets
 - Consommation de mémoire peut être élevée

Micronaut

- Avantages :
 - Démarrage rapide et faible consommation de mémoire
 - Support natif pour GraalVM
- Inconvénients :
 - Moins mature que Spring Boot
 - Plus petite communauté

Architecture Serverless

- AWS Lambda avec Java
- Spring Cloud Function
- Micronaut
- Apache OpenWhisk

AWS Lambda avec Java

- Avantages :
 - Facilité d'intégration avec l'écosystème AWS
 - Scalabilité automatique
- Inconvénients :
 - Temps de démarrage des fonctions peut être élevé (cold start)
 - Dépendance à l'égard des services AWS

Spring Cloud Function

- Avantages :
 - Abstraction des fournisseurs serverless
 - Réutilisation du code Spring existant
- Inconvénients :
 - Peut ajouter une couche de complexité
 - Performances peuvent varier selon le fournisseur

Architecture Événementielle

Event-Driven Architecture

- Akka
- Kafka Streams
- Spring Cloud Stream

Akka

- Avantages :
 - Hautement concurrentiel et distribué
 - Supporte les modèles de programmation réactifs
- Inconvénients :
 - Courbe d'apprentissage raide
 - Complexité de la gestion des acteurs

Kafka Streams

- Avantages :
 - Intégré avec Apache Kafka pour le traitement de flux
 - Haute performance et scalabilité
- Inconvénients :
 - Complexité de mise en place
 - Dépendance à Kafka

Frameworks Front-End

- SPA : Angular, React, Vue.js, Svelte, Ember.js
- MPA : Thymeleaf, JSP, JSF, Freemarker, Handlebars.js

Angular

- Avantages :
 - Framework complet avec un écosystème riche
 - Maintenu par Google
- Inconvénients :
 - Courbe d'apprentissage élevée
 - Taille des bundles peut être importante

React

- Avantages :
 - Flexible et réutilisable grâce aux composants
 - Large communauté et support de Facebook
- Inconvénients :
 - Nécessite des bibliothèques supplémentaires pour un framework complet
 - Fréquence des mises à jour peut être difficile à suivre

Frameworks de Tests

- Tests Unitaires : JUnit, TestNG, Mockito, PowerMock, Spock
- Tests d'Intégration : Spring Test, Cucumber, Arquillian, DBUnit

JUnit

- Avantages :
 - Standard de facto pour les tests unitaires en Java
 - Large support et documentation
- Inconvénients :
 - Limité aux tests unitaires
 - Fonctionnalités avancées limitées

Mockito

- Avantages :
 - Facilité de création de mocks pour les tests
 - Bonne intégration avec Junit
- Inconvénients :
 - Peut devenir complexe pour des scénarios de tests avancés
 - Ne supporte pas le mocking des méthodes statiques sans PowerMock

Rapports et Statistiques

- JasperReports
- BIRT
- Pentaho
- Apache POI
- JFreeChart

Surveillance et Monitoring

- Splunk
- Prometheus
- Grafana
- ELK Stack (Elasticsearch, Logstash, Kibana)
- New Relic
- Datadog
- Zipkin

Outils CI/CD

- Jenkins
- GitLab CI/CD
- Travis CI
- CircleCI
- Bamboo
- TeamCity
- Azure DevOps
- AWS CodePipeline
- Argo CD

Conclusion



Merci de votre Attention!