

# Intergiciels et Services

Emmanuel Coquery

# Organisation

2 intervenants :

- Lionel Médini
- Emmanuel Coquery

Page de l'UE :

```
http://liris.cnrs.fr/~ecoquery/dokuwiki/doku.php?id=
enseignement:tiw1-is:start
```

# Sommaire

- Révision Java (aujourd'hui)
- Introduction aux composants, *IoC*
- Frameworks client (Angular)
- Contexte, annuaire, *pooling* d'instance
- Introduction aux services
- RMI
- Services Web
- Architecture des serveurs d'application
- EJB, Spring

# Sommaire

- Implémentation de services
- Intercepteurs, composition, bus de services
- Processus métier (bas niveau), *Workflows*
- Microservices
- Performance
- Bonnes pratiques

# Environnement de développement intégré

Connaître son IDE :

- Raccourcis clavier
- Structuration des projets
- Affichage/intégration de la documentation
- Intégration outils de build, serveurs
- Affichage des logs
- Génération de code
- ...

Il est souvent rentable de passer 5/10 min à comprendre comment bien faire une chose dans son IDE.

# Environnement de développement intégré

Savoir se passer de son IDE :

- Pour compiler, exécuter les tests, ...
- Pour déployer
- Savoir se débrouiller en ligne de commande

## Bibliothèques & *frameworks*

Log :

- L'utilisation de `System.out/.err` est **interdite**
- Utiliser logging, SLF4J, Log4J, ...

Test :

- Tester son code
- via un framework de test (JUnit, TestNG, etc)
- regarder ce qui existe pour tester dans les cas plus complexes (e.g. test d'intégration) :
  - Spring Test, Arquillian
  - SOAPUI, Selenium

# Qualité de code

Faire attention à la qualité du code :

- formattage
- commentaire
- simplicité de compréhension

Auditer le code

- Automatiquement (e.g. sonarqube)
- Entre vous