



zenika  
ARCHITECTURE INFORMATIQUE

www.zenika.com

Repartez avec le livre



Durée  
2 jours

Répartition  
30% théorie  
70% pratique

Pré-requis  
Connaissance de Eclipse  
RCP

Public  
Architecte  
Développeur  
Chef de projet

Tarif (déjeuners inclus)  
1200 € (HT)  
Tarif (1 mois avant)  
1100 € (HT)

Lieu  
Paris 9ème

Sessions 2009  
24 au 25 septembre  
29 au 30 octobre  
10 au 11 décembre

Intra-entreprise sur  
demande

Inscription et  
renseignements  
+33(0)1.45.26.19.15  
training@zenika.com  
www.zenika.com

# Eclipse RAP pour développeur RCP

## Développement d'applications riches avec Eclipse RAP pour développeur Eclipse RCP

### Objectifs

- Créer une application web 2.0 reposant sur la plate-forme Eclipse RAP
- Connaître et comprendre les différences entre RAP et RCP
- Savoir porter une application RCP existante en RAP ...
- ... et continuer à les faire évoluer conjointement

### Contenu

#### Présentation

- Positionnement d'Eclipse RAP dans la mouvance RIA (Rich Internet Application)
- Comparaison par rapport à GWT, Flex, Wicket, etc...
- Principe du Single Sourcing entre RAP et RCP

#### Mon premier projet RAP

- Créer une application RAP, les points d'entrées
- Créer un premier écran Hello World
- Exécuter et déboguer une application RAP grâce au Jetty embarqué
- Gérer les configurations de lancements

#### Différences entre RAP et RCP

- Environnement Mono vs Multi-utilisateurs
- Plug-ins et dépendances
- Points d'extensions (Help, PhaseListener...)
- API (GC, MouseMove events...)
- Gestion des ressources
- Fonctionnalités aux comportements différents

#### Single Sourcing d'une application RAP et RCP

- Bénéficier du code d'une application RCP existante
- Factoriser le maximum de code commun entre applications
- Mise en place d'un socle de plug-ins permettant le Single Sourcing

#### RWT : les composants graphiques

- Similarité et différences par rapport à SWT
  - Utilisation des widgets basiques
  - Agencement des composants à l'aide du système de Layouts : FillLayout, GridLayout, ...
  - Gestion des événements graphiques (events) par Listeners (ex: ActionListener)
- Définir ses propres composants graphiques
  - JavaScript coté client
  - Adapteur Java coté serveur
  - Synchronisation des états
- Gestion des thèmes CSS

#### Gestion des différents contextes applicatifs

- Comprendre les différents niveaux disponibles
- Accéder aux contextes par la classe RWT
- Création de singleton grâce à SessionSingleton

#### Packager et livrer l'application

- Présentation du mécanisme de Feature
- Points d'entrées et thèmes de l'application
- Packaging war de l'application RAP à l'aide du PDE
- Test de charge d'une application RAP

#### Avancé

- RAP Incubator (ex : RAP Theme Editor)
- Améliorer l'aspect graphique avec Eclipse UI Forms
- Bénéficier de l'expérience de la communauté RAP
- Intégration BIRT