Plugin Eclipse - Scout

26 novembre 2012

Julien VERNEAU

Sommaire

- Introduction
- Présentation
- Construire une application
- Exemple d’utilisation
- Conclusion

Sommaire

- Introduction
  - Historique
  - Objectifs du projet Scout
- Présentation
- Construire une application
- Exemple d'utilisation
- Conclusion

Historique

- La fondation Eclipse existe depuis 2003
- Fondée par IBM
- Depuis 2010, Eclipse Scout est un projet de la fondation Eclipse

Objectif du projet Scout

- Permettre un développement rapide et bien formé

- Définition du projet
  - « Eclipse Scout est un Framework moderne et ouvert qui permet de réaliser facilement des applications métiers orientés services »

Sommaire

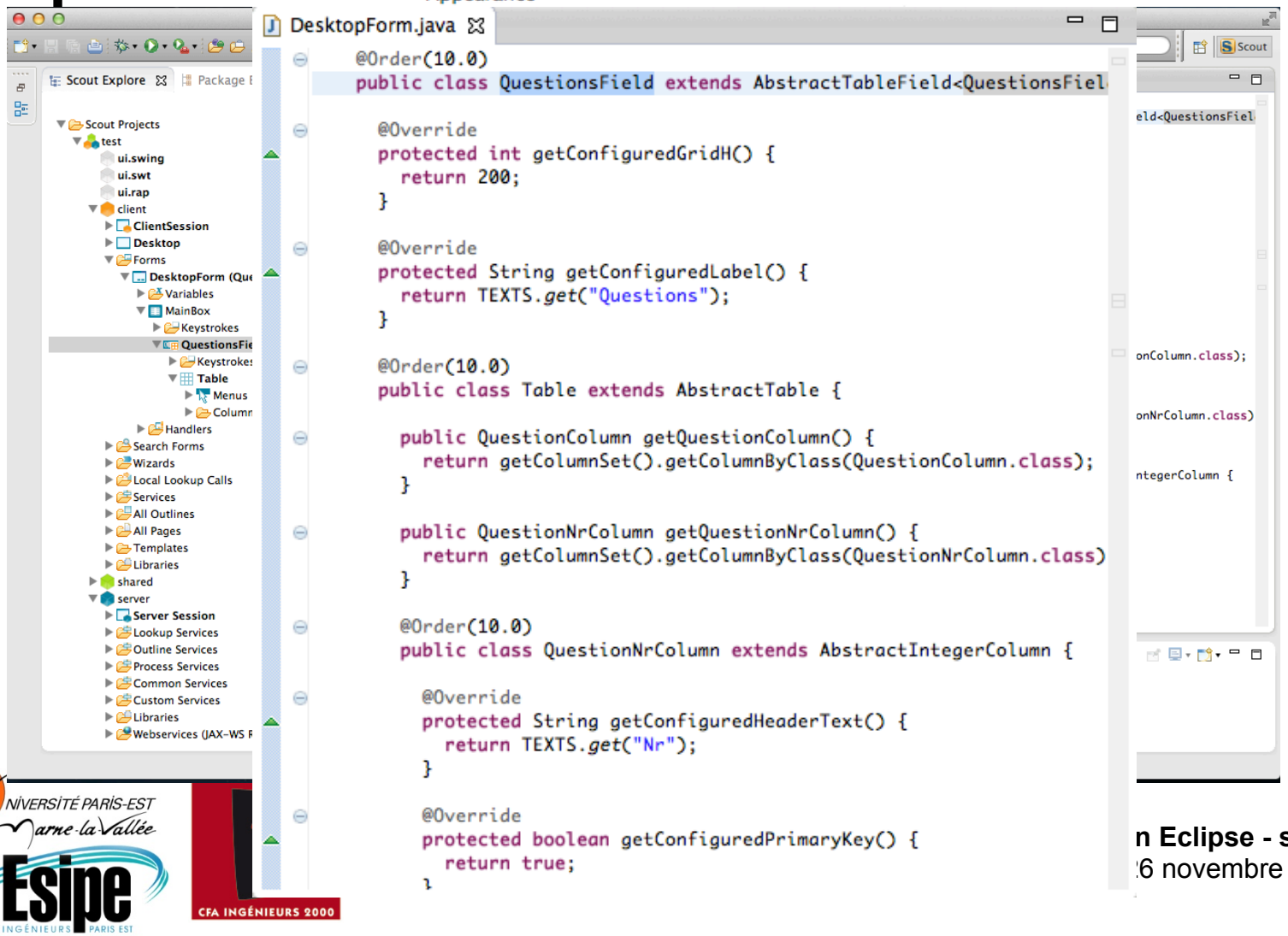
- Introduction
- Présentation
  - La partie SDK
  - La partie runtime
  - Architecture client / serveur
  - Applications réalisées avec Scout
- Construire une application
- Exemple d'utilisation
- Conclusion

La partie SDK

- Production d'un code lisible
- Développeur concentré sur les données
- Mise à disposition de vues et d'assistants
  - Application sous forme arborescente
  - Zone de propriété d'éléments

La partie SDK

Appearance



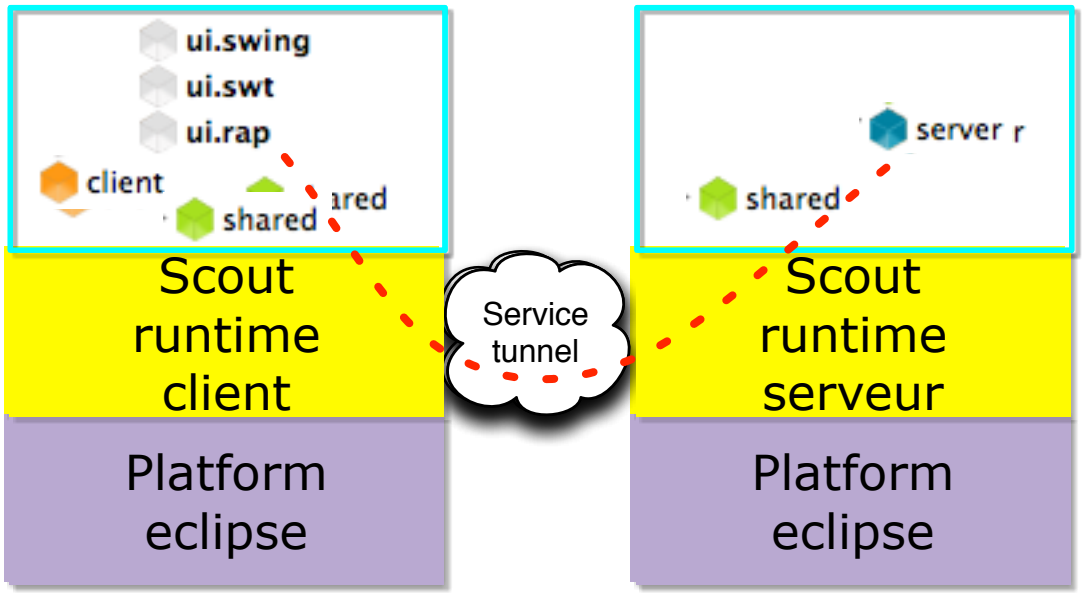
La partie runtime

- Dans chaque application on retrouve
  - Une partie client
  - Une partie serveur
- Plusieurs bibliothèques graphiques
  - Swing
  - SWT
  - Web avec RAP

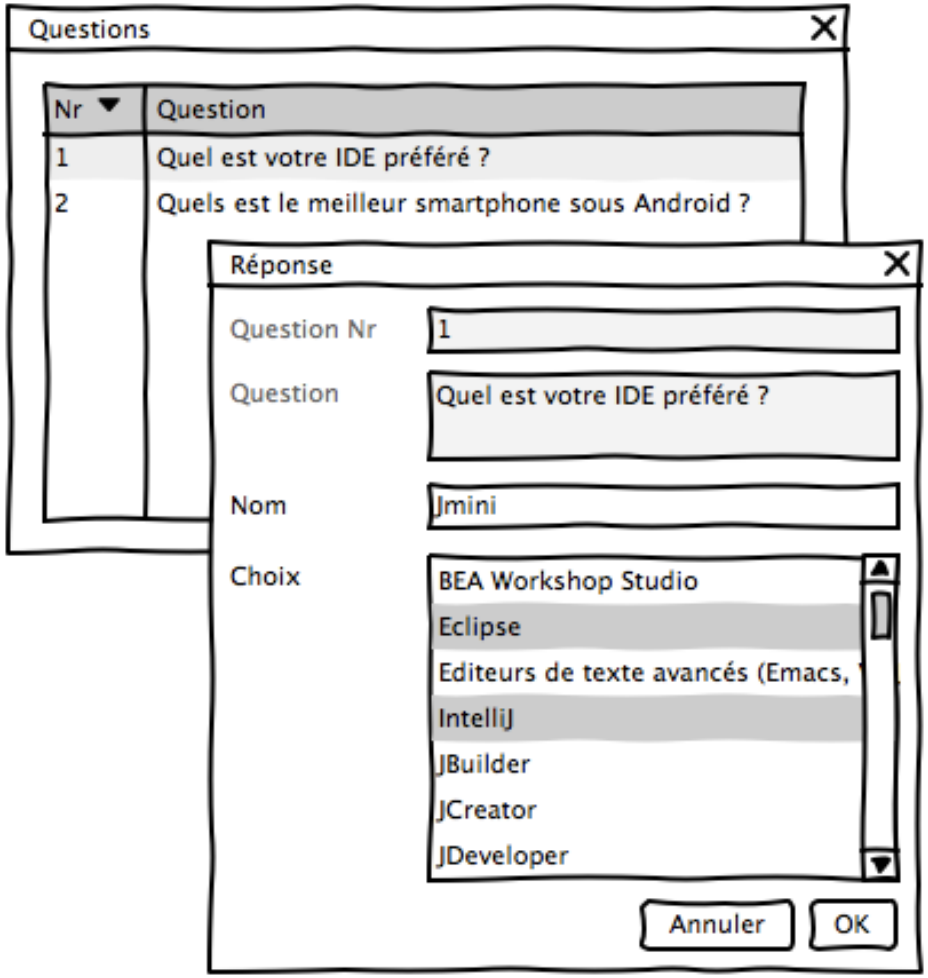
La partie runtime

- Eclipse Scout génère le rendu graphique
- Le développeur ne s'occupe que du contenu et de la présentation

Architecture Client / Serveur



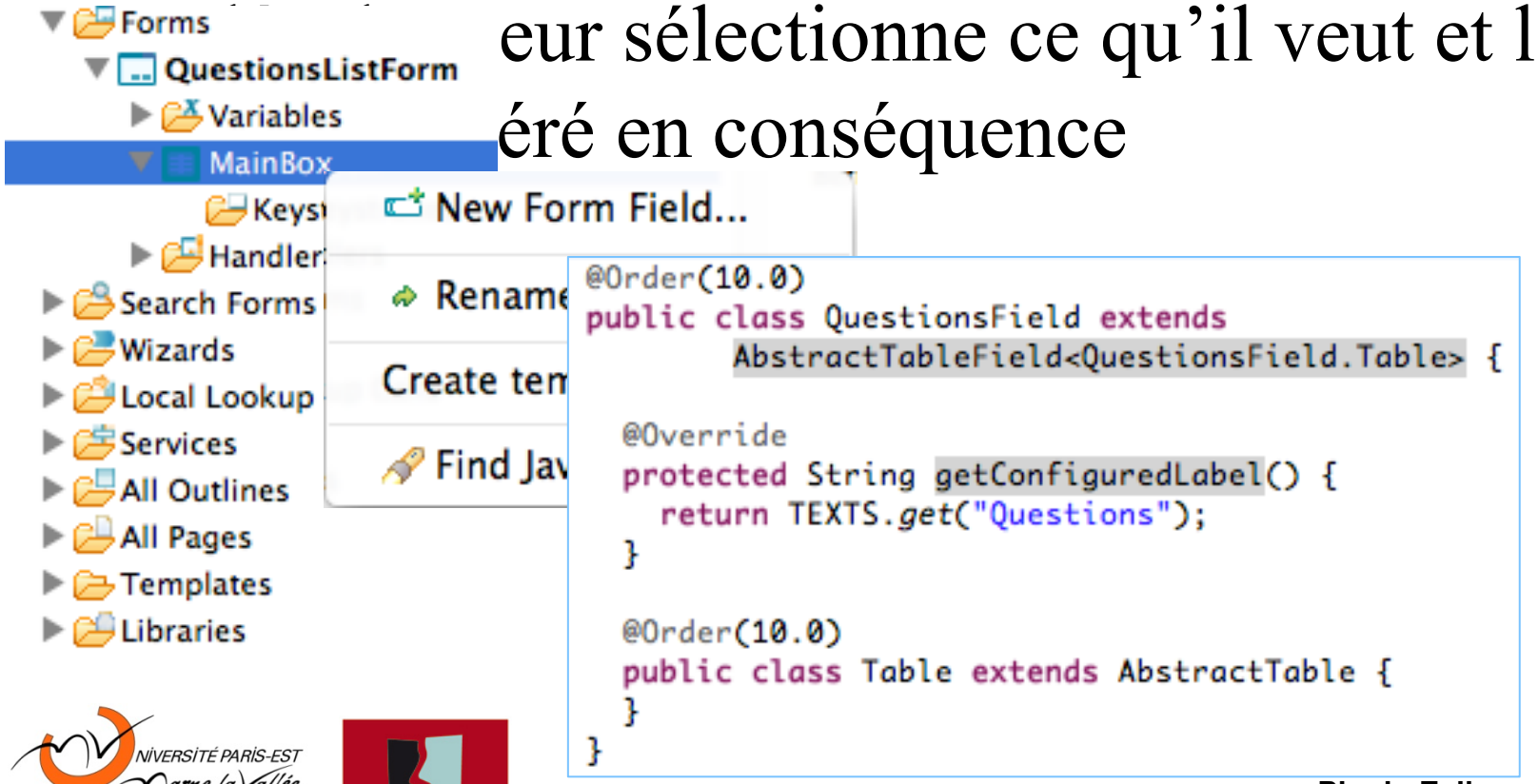
Applications réalisées avec Scout



Sommaire

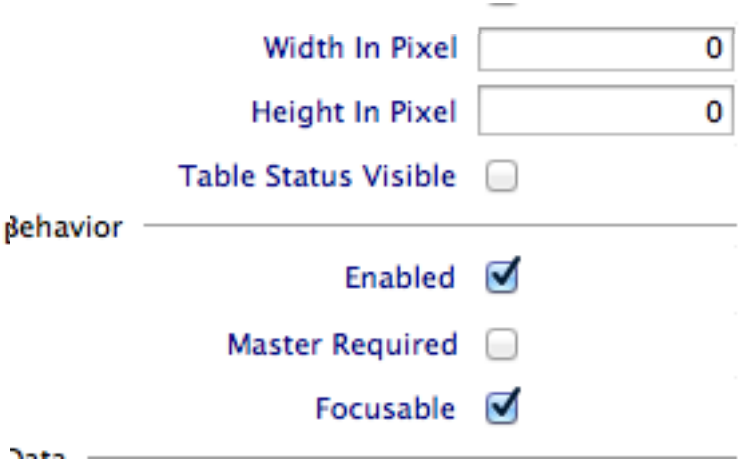
- Introduction
- Présentation
- Construire une application
  - Développement par « click »
  - Modification simple et ciblée
- Exemple d'utilisation
- Conclusion

Développement par « click »



eur sélectionne ce qu'il veut et le éré en conséquence

Développement par « click »




```
@Override
protected boolean getConfiguredEnabled() {
  return false;
}

@Override
protected String getConfiguredLabel() {
  return TEXTS.get("Questions");
}
```

Modification simple et ciblée

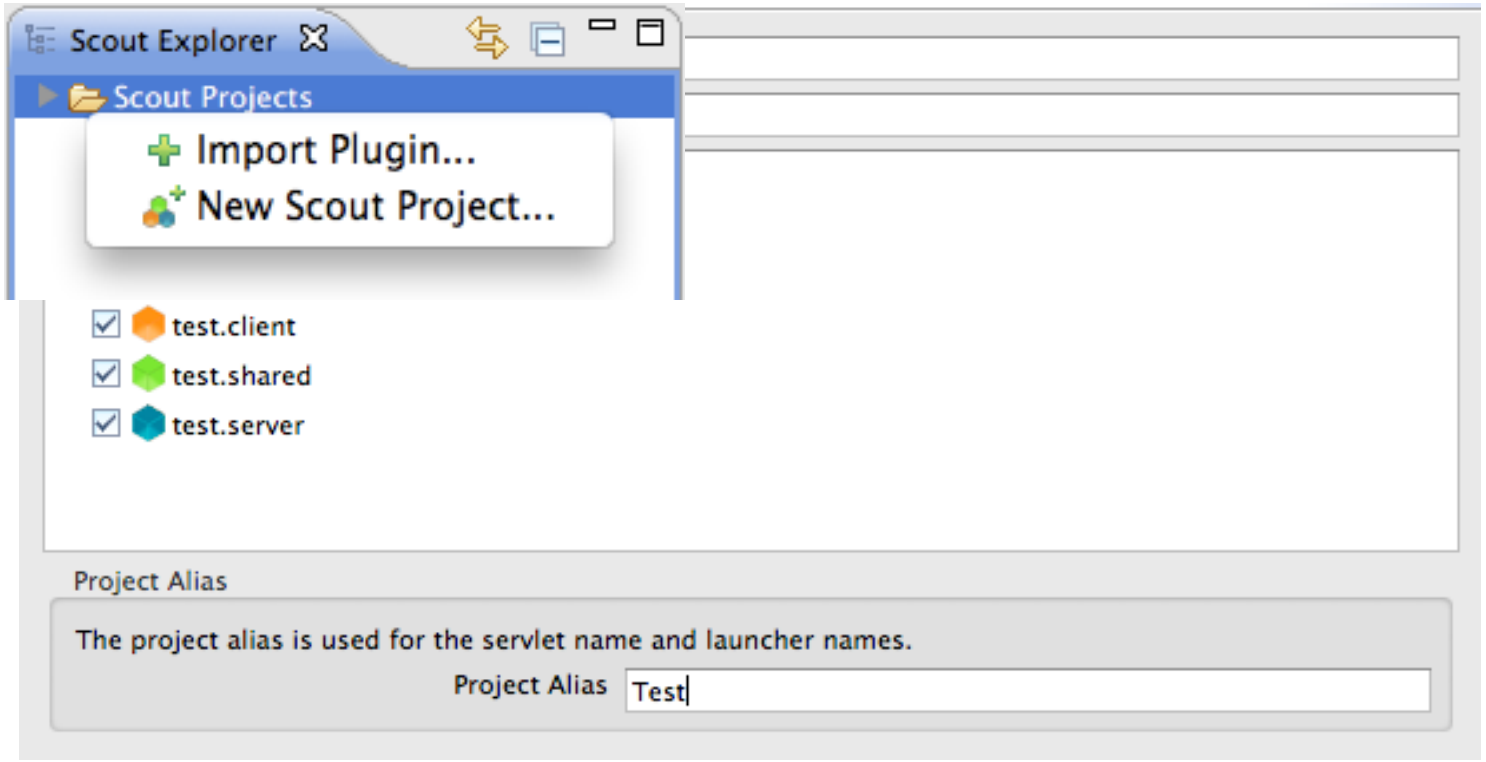
- Modification manuelle autorisée et conseillée
- Ajouts d'éléments fonctionnels particuliers

```
@Override
protected void execMarkSaved() throws ProcessingException {
  //TODO Mettre le code que l'on souhaite réaliser
}
```

Sommaire

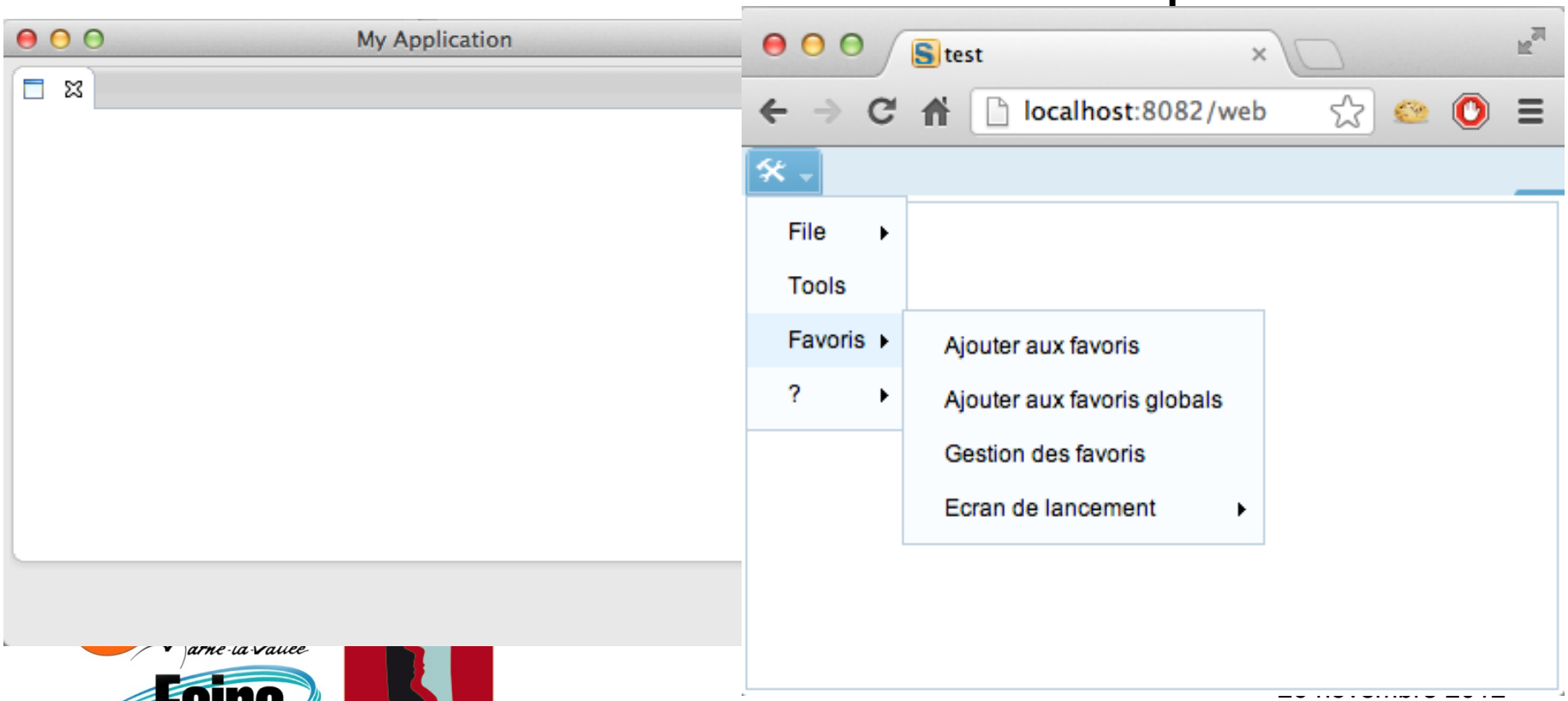
- Introduction
- Présentation
- Construire une application
- Exemple d'utilisation
  - Création d'un projet
  - Ajout d'éléments
  - Connexion à une base de données
  - Liaison du contenu à la base de données
  - Mise en production
- Conclusion

Création d'un projet

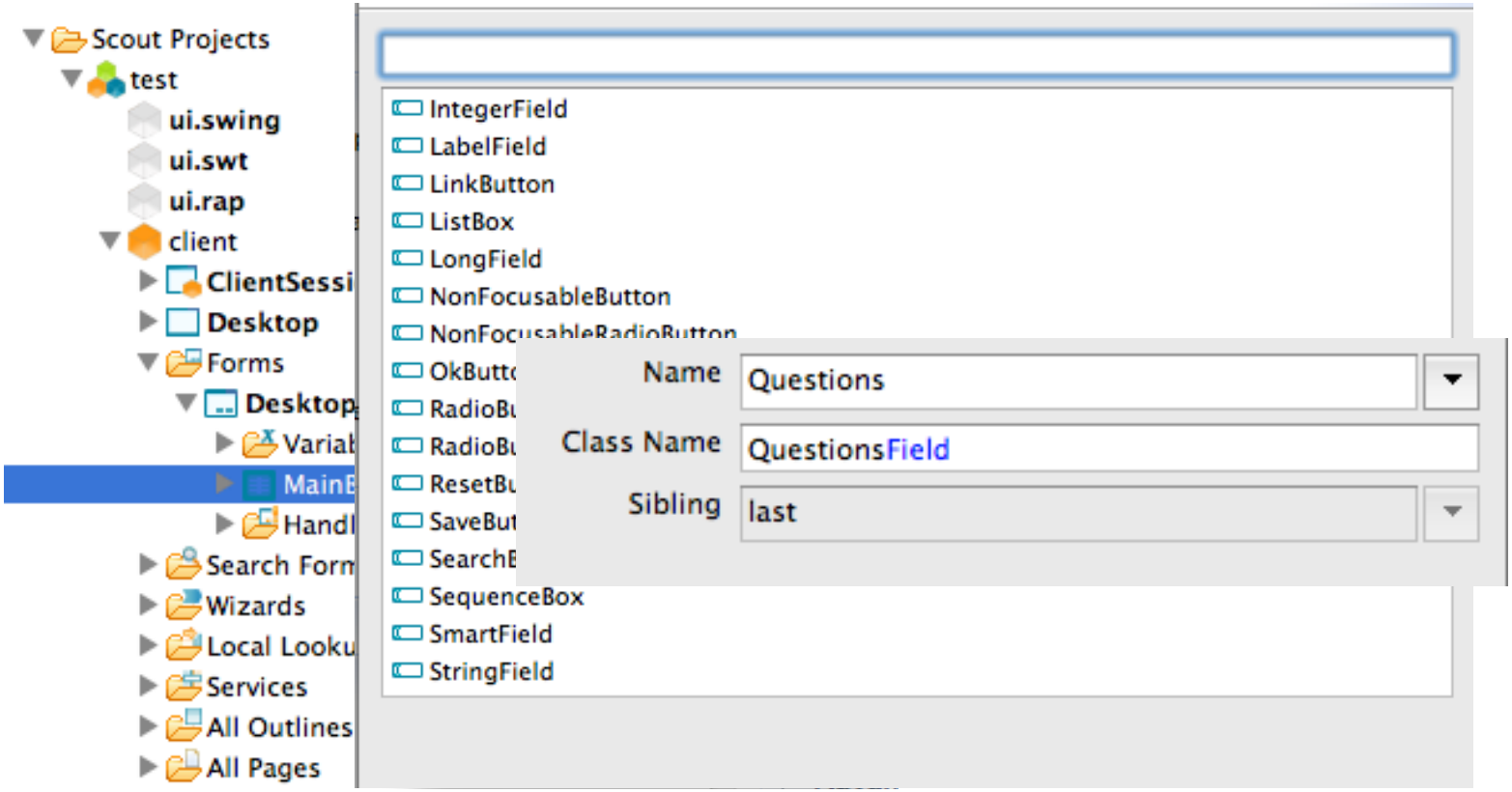


Création d'un projet

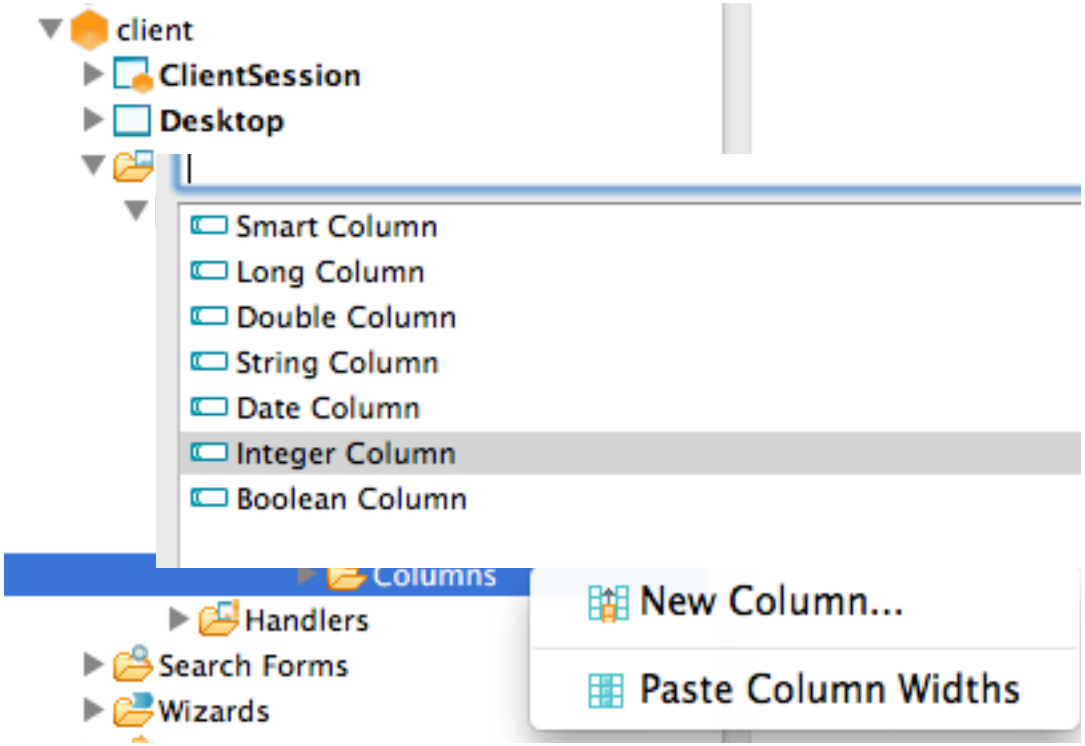
- Architecture fonctionnelle dès le départ du

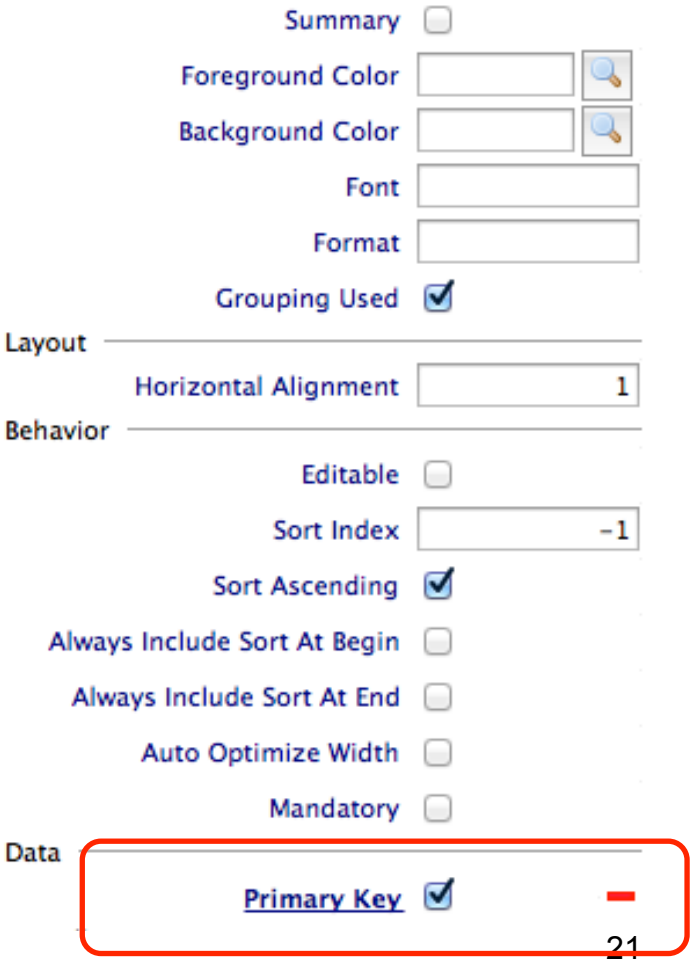


Ajout d'éléments



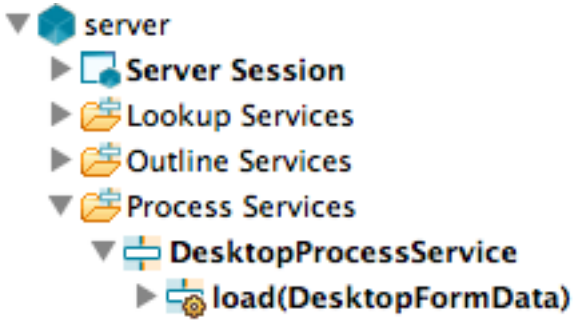
Ajout d'éléments





Ajout d'éléments

- Remplissage du tableau créé

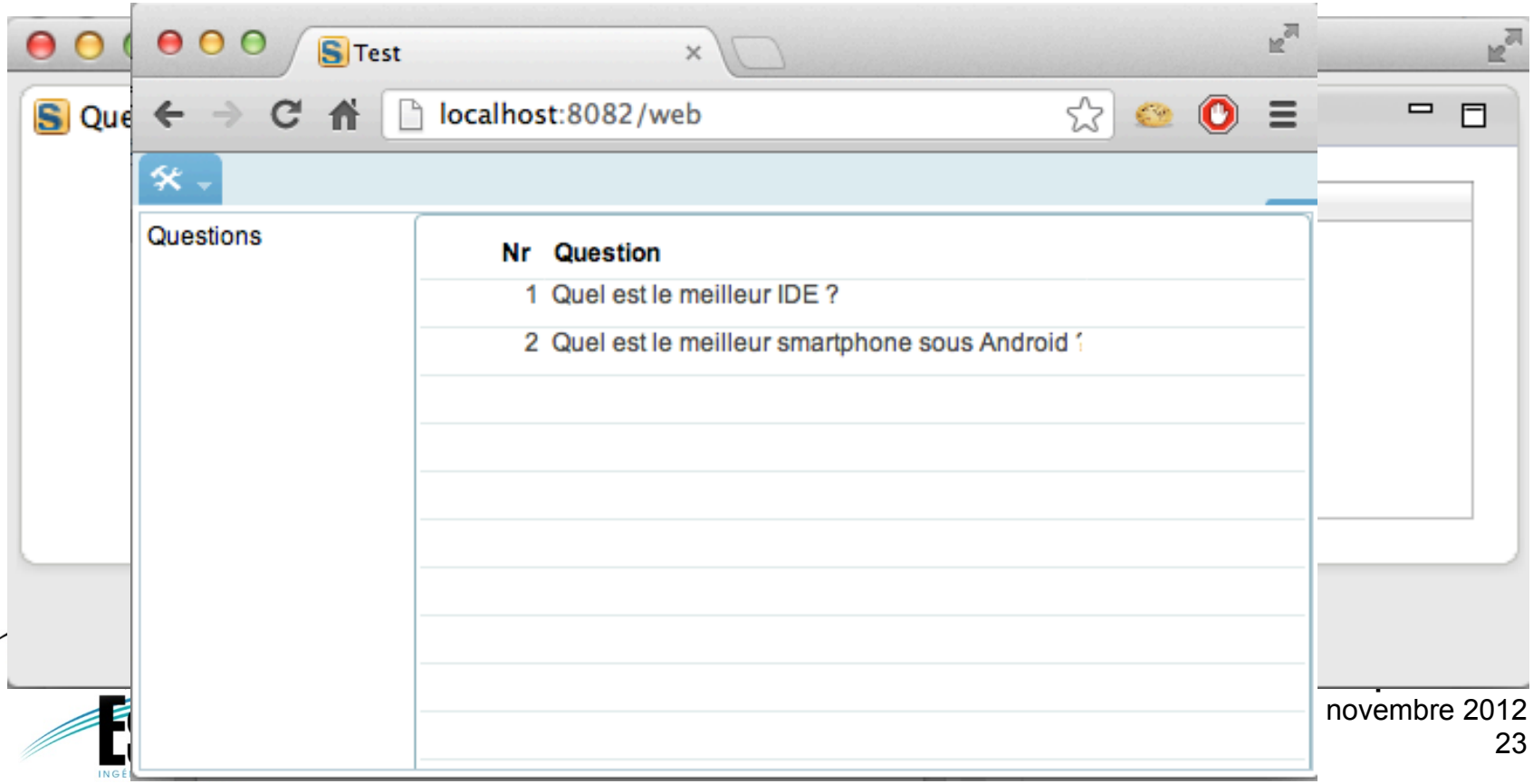



```
@Override
public DesktopFormData load(DesktopFormData formData) throws ProcessingException {
  formData.getQuestions().addRow(new Object[]{1, "Quel est le meilleur IDE ?"});
  formData.getQuestions().addRow(new Object[]{2, "Quel est le meilleur smartphone sous Android ?"});
  return formData;

}
```

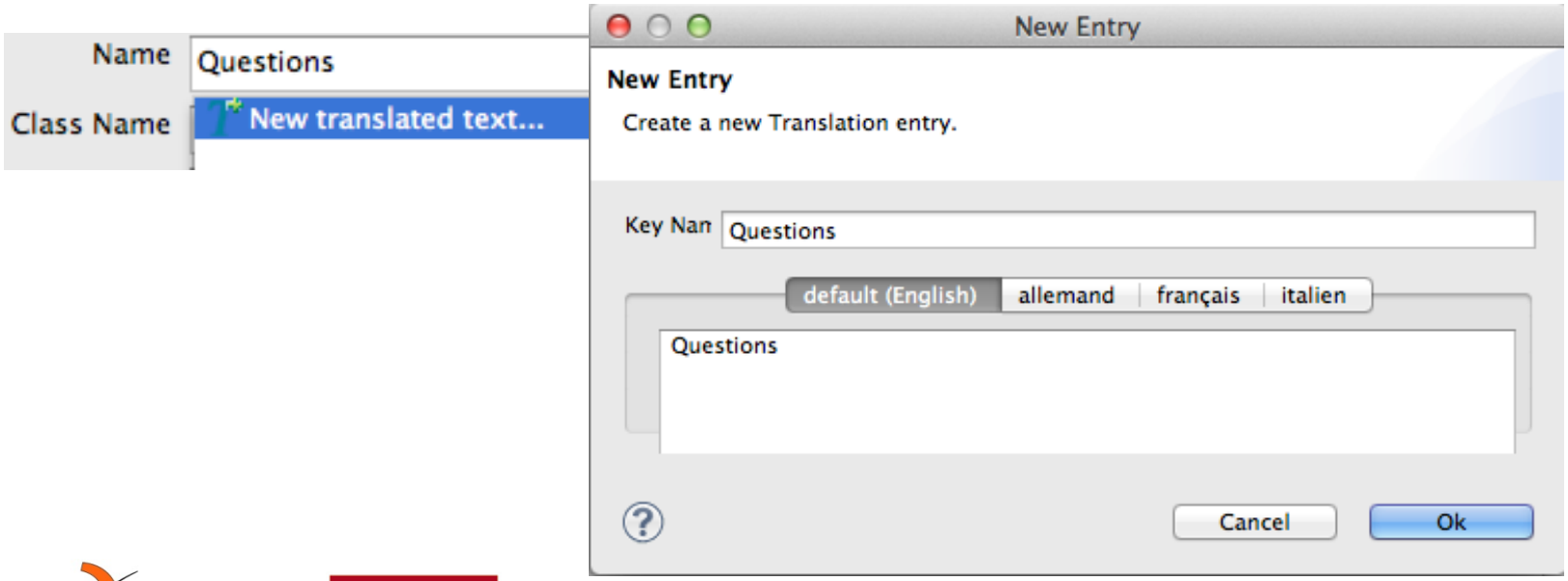
Ajout d'éléments

- Vision d'avancement



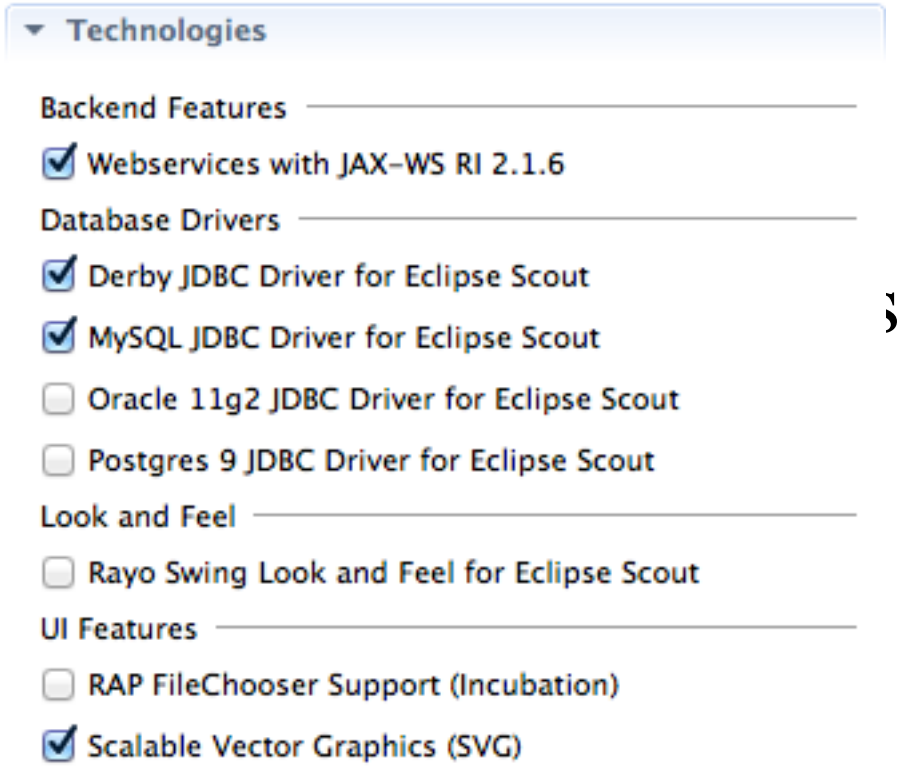
Ajout d'éléments

- Traduction de tous les éléments créés



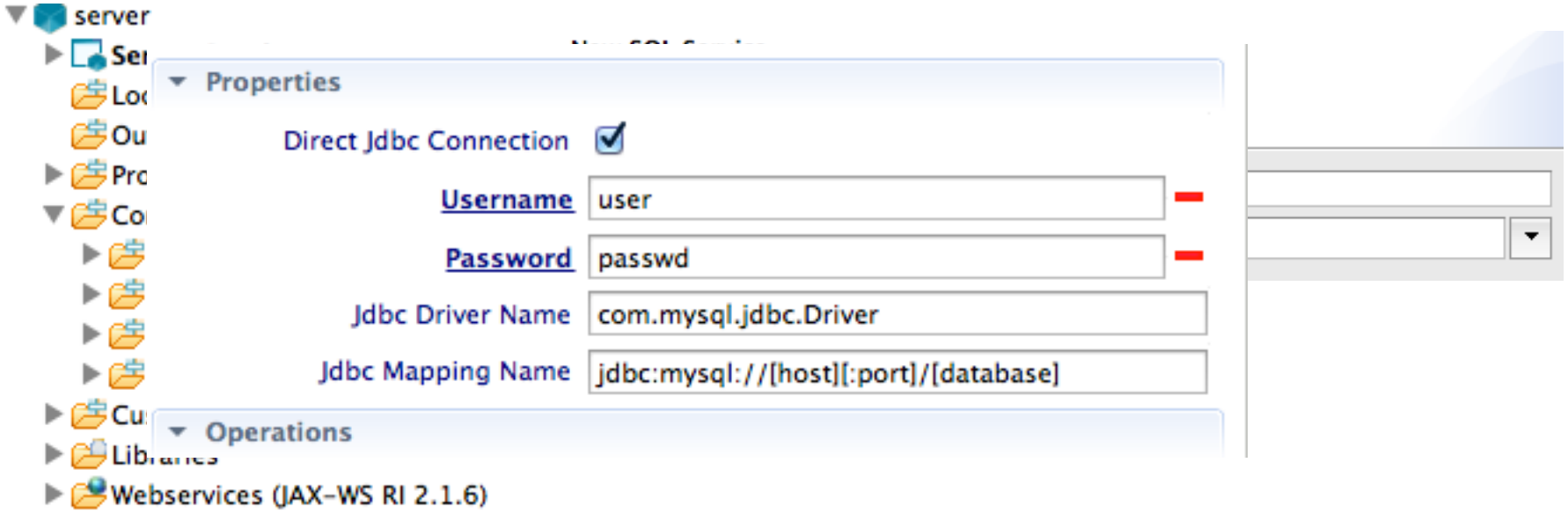
Connexion à une base de données

- BDD autorisées dans
- s besoins

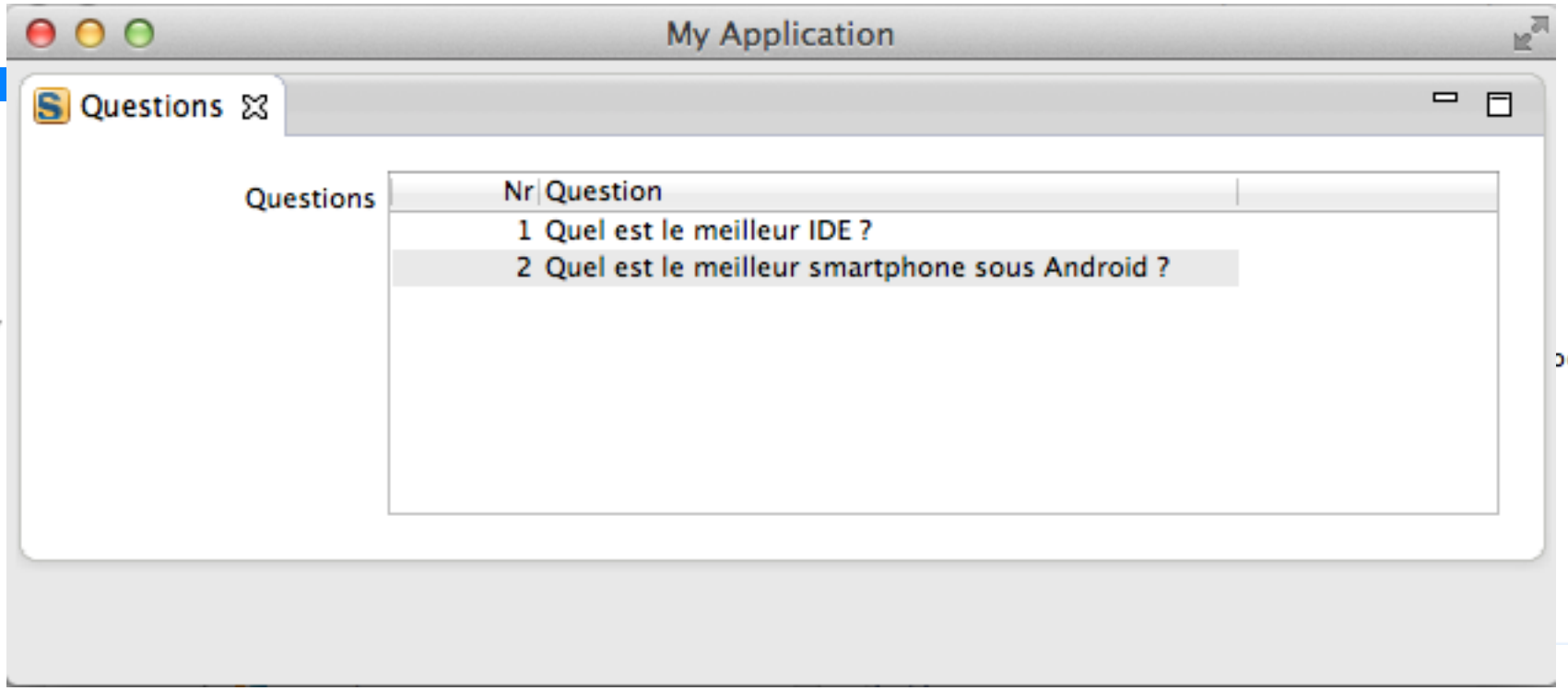


Connexion à une base de données

- Configuration de la BDD

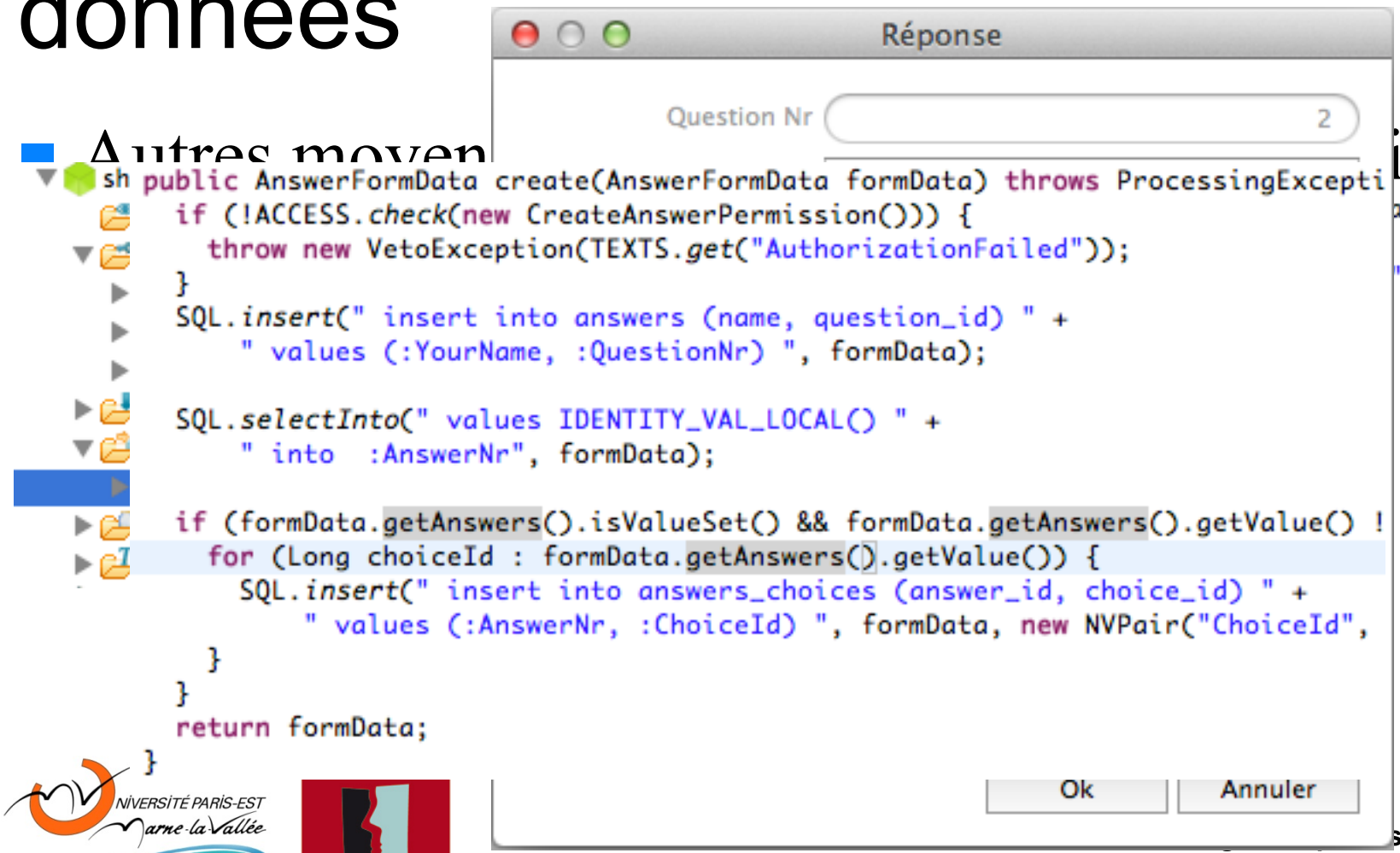


Liaison du contenu à la base de données



Liaison du contenu à la base de données

- Autres moyen… …ia




```
public AnswerFormData create(AnswerFormData formData) throws ProcessingExcepti
  if (!ACCESS.check(new CreateAnswerPermission())) {
    throw new VetoException(TEXTS.get("AuthorizationFailed"));
  }
  SQL.insert(" insert into answers (name, question_id) " +
      " values (:YourName, :QuestionNr) ", formData);

  SQL.selectInto(" values IDENTITY_VAL_LOCAL() " +
      " into  :AnswerNr", formData);

  if (formData.getAnswers().isValueSet() && formData.getAnswers().getValue() !
    for (Long choiceId : formData.getAnswers().getValue()) {
      SQL.insert(" insert into answers_choices (answer_id, choice_id) " +
          " values (:AnswerNr, :ChoiceId) ", formData, new NVPair("ChoiceId",
    }
  }
  return formData;
}
```

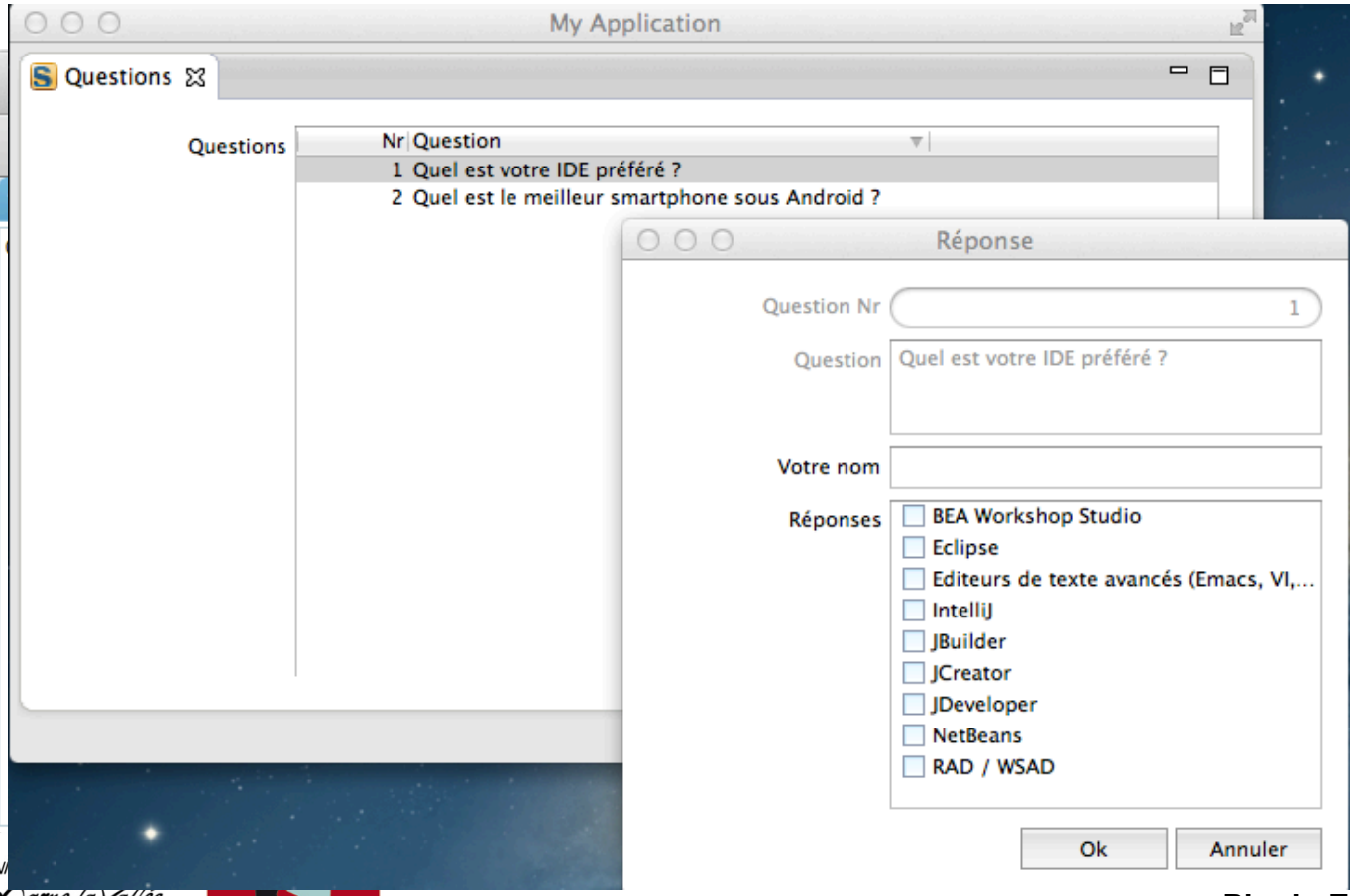
Mise en production

- Déploiement de l'application
  - Simple et rapide
  - Export en WAR pour serveur Tomcat

- Suite de la configuration sur le serveur Tomcat

Mise en production

- R



Extension

- Gestion optimisée des permissions
- Intégration du Framework Jython
- Intégration de JasperReports

Conclusion

- Projet récent pas actuellement adapté pour toutes les applications
- A grand potentiel
- Commence à être utilisé dans le monde de l'entreprise
- Nécessite des bases en java et en SQL
- Extension du Framework possible

Sources

- Site officiel
  - http://www.eclipse.org/scout/
- Wiki
  - http://wiki.eclipse.org/Scout/

Merci de votre attention

- Questions ?