. 20% - Rappels

a. Langage Groovy :

i. Comment fonctionne l’interprétation et la convention de nommage ?

ii. Les différences avec Java

b. Modèle MVC

2. 70% - Approfondissement du framework Grails 2.5.5 (niveau intermédiaire)

a. Bonnes pratiques de développement :

i. Organisation modulaire/package

ii. Convention de nommage/arborescence

iii. Mettre des paramètres avec valeur par défaut dans les méthodes est-ce une bonne pratique ou non ?

iv. Typage vs def

v. Où mettre les beans ?

vi. Gestion des classes utilitaires, les builders

vii. Optimisation de la maintenance du code

b. Ensemble des mécanismes de Grails du BO jusqu’à la View : de notre point vue c’est très opaque, même si derrière on pressent que c’est une surcouche de Spring, Hibernate, ect….

c. Fonctionnement détaillé des interactions avec le Controller :

i. En particulier le lien Controller <->  View : pourquoi certains Controllers appellent-ils directement la vue ?

ii. Quels sont les paramètres que l’on récupère de la Vue ?

iii. Pourquoi passer par un objet  \*CMD.groovy parfois au lieu de la map « params » ?

iv. Comment Grails fait le mapping avec cet objet

v. Gestion du retour d’erreur et validation des formulaires

vi. Gestion des appels asynchrones (Ajax)

d. La gestion des services :

i. Propagation des transactions

ii. Comment créer un service avec une nouvelle transaction ?

iii. Comment charger un service dans un objet ?

iv. Service vs GORM, dans un Controller lequel choisir ?

e. Configuration des dépendances (Config.groovy) : où déposer les jar ?

f. Gestion et configuration du démarrage (Bootstrap)

g. Paramétrage multi environnements : développement/intégration/production

h. Compilation

i. Comment bien compiler (pour éviter d’appeler des méthodes qui n’existent pas ou avec trop de paramètres et de s’en rendre compte qu’à l’exécution) ?

ii. Gestion des refactors

3. 10% - GORM 4.0.0

a. Fonctionnement de la surcouche GORM : comment Grails le met-il en place ? Est-ce qu’il y a un cache ? Est-ce que c’est couplé avec hibernate ou bien est-ce que c’est indépendant ?

b. Paramétrage des différents modes, lequel choisir ?

c. Est-ce qu’on pourrait s’affranchir des criterias et tout faire avec des « find » « create » « save » « where » « delete » etc. ?

d. Est-il possible de surcharger/interdire l’utilisation de GORM de manière unitaire ?

e. Pourquoi avoir mis des appels dans les BO ?

f. Surcharge d’un getter vs Service

g. Conseils d’optimisation : appels multiples, temps d’exécution des requêtes

4. Facultatif : Migration de Grails (5-10 minutes)

a. Retours d’expérience : Impacts et avantages, plus-value

b. Migration vers la 3 ?