



Solution building block

Client : Consortium MedHead

Projet : Preuve de concept (l'allocation de lits d'hôpital pour les urgences)

Table des matières

Instruction.....3

Modèle de module de construction.....3

Modules de construction génériques.....4

Modules de construction de la preuve de concept (PoC).....5

Instruction

Tous les modules de construction doivent être succincts et précis comme dans l'exemple ci-dessous. Vous pouvez créer des renvois vers des informations supplémentaires stockées ailleurs dans le répertoire d'architecture.

Modèle de module de construction

1. Nom du module de construction : Solution pour créer un nouveau projet logiciel.
2. Fonctionnalité fournie : Le module de construction permet de normaliser la création de projets dans le cadre de solutions.
3. Lien vers des exemples d'implémentation ou d'interfaces : <https://start.spring.io>
4. Travail supplémentaire pour terminer ce module de construction : Implémenter des conteneurs Docker pour normaliser l'exécution Spring-CLI.
5. Alignement architectural : Ce module de construction permet ou reflète les objectifs et principes métiers suivants :
 - Objectif 1 : xxxxx
 - Principe 1 : yyyy
 - Objectif 2 : xxxxx
 - Principe 2 : yyyy

Modules de construction génériques

1. Nom du module de construction : Spring
2. Fonctionnalité fournie :
 1. Découplage des composants. Moins d'interdépendances entre les différents modules.
 2. Rendre plus aisés les tests des applications complexes c'est-à-dire des applications multicouches.
 3. Diminuer la quantité de code par l'intégration de frameworks tiers directement dans Spring.
 4. Permettre de mettre en oeuvre facilement la programmation orientée aspect.
 5. Un système de transactions au niveau métier qui permet par exemple de faire du "two-phases-commit".
 6. Un mécanisme de sécurité.
 7. Pas de dépendances dans le code à l'api Spring lors l'utilisation de l'injection. Ce qui permet de remplacer une couche sans impacter les autres.
 8. Une implémentation du design pattern MVC
 9. Un support du protocole RMI. Tant au niveau serveur qu'au niveau du client.
 10. Déployer et consommer des web-services très facilement.
 11. Echanger des objets par le protocole http
3. Lien vers des exemples d'implémentation ou d'interfaces : <https://start.spring.io>
4. Travail supplémentaire pour terminer ce module de construction : Implémenter des conteneurs Docker pour normaliser l'exécution Spring-CLI.
5. Alignement architectural:
 - Objectif 1 : Gagner la confiance des utilisateurs quant à la simplicité d'un tel système.
 - Principe 1 : Clarté grâce à une séparation fine des préoccupations

Modules de construction de la preuve de concept (PoC)

1. Nom du module de construction : **Urgence**
2. Fonctionnalité fournie :
 1. Retourner une liste des urgences complété
 2. Poster une urgence que le système va devoir compléter
3. Lien vers des exemples d'implémentation ou d'interfaces :
4. Travail supplémentaire pour terminer ce module de construction : Implémenter des conteneurs Docker pour normaliser l'exécution Spring-CLI.
5. Alignement architectural :
 - Objectif 1 : améliorer la qualité des traitements d'urgence et de sauver plus de vies
 - Principe 1 : Favoriser une culture de “learning” avec des preuves de concept, des prototypes et des Spike

1. Nom du module de construction : **Hopital**
2. Fonctionnalité fournie :
 1. Retourner la liste des hopitaux du parc hospitaliers
 2. Retourner un hopital selon une spécialisation et une coordonnée d'origine
3. Lien vers des exemples d'implémentation ou d'interfaces :
4. Travail supplémentaire pour terminer ce module de construction : Implémenter des conteneurs Docker pour normaliser l'exécution Spring-CLI.
5. Alignement architectural :
 - Objectif 1 : que plus de 90 % des cas d'urgence sont acheminés vers l'hôpital compétent le plus proche du réseau
 - Principe 1 : Favoriser une culture de “learning” avec des preuves de concept, des prototypes et des Spike