



Programme Odyssey

Consolidation des Acquis - Cahier d'Exigences

Sommaire

Ce document vise à documenter les exigences du programme de consolidation des fondamentaux du programme “Odyssey” offert par CodeBoxx d’une durée de quatorze (14) semaines. Il est une opportunité de revenir de manière individuelle sur les notions vues précédemment

À la fin de chaque thème hebdomadaire, les candidats doivent avoir pu se familiariser et mettre en pratique les notions qui les rendent opérationnels sur les champs en lien avec le thème abordé lors de cette semaine. Les compétences de base d’un développeur Full Stack sont reprises alors que les outils courants et les bonnes pratiques qui y sont rattachées sont utilisées au fil des semaines.

Objectif

1. Décrire la mise en situation, le contexte, les tâches, exigences, livrables et critères d’acceptabilité pour évaluer la réussite du participant au programme

Exercice de Consolidation des Acquis

Après 7 semaines, CodeBoxx à travers son programme Odyssey, a permis d’offrir un contexte et un jeu d’exigences qui ont permis aux participants d’être exposés aux outils, pratiques et disciplines fondamentales du métier de développeur:

- Mécanique du Web
 - HTML 5
 - CSS 3
 - Javascript
 - Bootstrap 4
- Algorithmie
- Langages Interprétés
- Langages Compilés
- Applications Web avec Ruby on Rails
- Bases de Données Relationnelles Transactionnelles
- Bases de Données Décisionnelles
- Restitution Décisionnelle
- Connexion et exploitation d’API Partenaires
- API en REST
- API en GraphQL

L'apprentissage s'est fait de manière individuelle au démarrage puis l'apprentissage s'est accéléré à l'aide de la formation d'équipes capables de parcourir davantage de terrain et obtenir des résultats à l'aide de la force du groupe.

Il se pourrait qu'à ce stade, certains participants n'aient pas été en mesure de s'exposer suffisamment à l'ensemble des notions couvertes étant donné le rythme accéléré du programme. Pour permettre aux participants de revenir sur les notions et consolider leurs acquis, un exercice individuel supplémentaire est proposé:

I. Mise en Situation et Contexte

Une partie importante du système d'information de *Rocket Elevators* est toujours manquante et il s'agit de la gestion des interventions. Pour l'ajouter, il faut s'inspirer des éléments déjà implantés dans la solution actuelle et les utiliser à nouveau pour implémenter une nouvelle entité.

La Gestion des Interventions requiert:

1. Un nouveau formulaire pour formuler la demande
2. La création d'une nouvelle table
3. La sauvegarde de la demande d'intervention en base de données
4. L'envoi d'un ticket de service à Zendesk
5. L'ajout d'une requête API qui retourne les nouvelles demandes d'intervention

II. Jalons et Critères de Réussite

Une nouvelle section du Site Web nommée "Interventions" doit être ajoutée et elle doit être accessible du menu après "Contact". Cliquer sur la nouvelle rubrique "Interventions" amènera le visiteur à une nouvelle vue dans l'application Rails. Attention, ce formulaire doit être visible et utilisable sur mobile. Il est donc impératif de garder les capacités offertes par Bootstrap 4.

Rebaser son système d'information en mode Individuel

Dans un premier temps, il faut récupérer dans son référentiel une version du système d'information avec laquelle le participant est familier et le plus susceptible de retrouver ses repères et c'est de cette fondation qu'il doit pouvoir repartir. La version doit être récupérée au sein d'un nouveau référentiel public de son propre compte GitHub qu'il nommera "Rocket-Elevators-Foundation". Le participant doit pouvoir déployer son système d'information nouvellement rapatrié lui-même derrière son propre nom de domaine. Tous les composants doivent être déployés:

1. L'application Ruby on Rails
2. La base de données MySQL
3. L'API REST

Formulaire de demande d'Intervention

Le formulaire s'adresse aux employés de Rocket Elevators afin qu'ils puissent saisir une nouvelle demande d'intervention. Dans un premier temps Il faut déterminer pour quel entité cette intervention doit être effectuée:

1. Afficher un Sélecteur DropDown listant tous les clients (Table Customers), la sélection d'un seul client à la fois peut être faite. Dès que la sélection est effectuée, le changement déclenche l'action du champ de sélection des buildings du formulaire - étape 2 suivante
2. Une fois la sélection du client effectuée, il faut ensuite afficher un sélecteur listant les bâtiments appartenant à ce client. Le sélecteur Dropdown ne permet la saisie que d'une seul bâtiment à la fois. Dès que la sélection est effectuée, le changement déclenche l'action du champ de sélection des batteries du formulaire - étape 3 suivante
3. Une fois la sélection du bâtiment effectuée, il faut ensuite afficher un sélecteur listant les batteries appartenant à cet immeuble. Le sélecteur Dropdown ne permet la saisie que d'une seule batterie à la fois. Dès que la sélection est effectuée, le changement déclenche l'action du champ de sélection des colonnes du formulaire - étape 4 suivante
4. Une fois la sélection de la batterie effectuée, il faut ensuite afficher un sélecteur listant les colonnes appartenant à la batterie. Le sélecteur Dropdown possède une valeur "Aucune" parmi les choix et il s'agit du choix par défaut. La dropdown ne permet la saisie que d'une seule colonne à la fois. Dès que la sélection est effectuée, le changement déclenche l'action du champ de sélection des colonnes du formulaire - étape 5 suivante
5. Une fois la sélection de la colonne effectuée, il faut ensuite afficher un sélecteur listant les ascenseurs appartenant à la colonne. Le sélecteur Dropdown possède une valeur "Aucune" parmi les choix et il s'agit du choix par défaut. La dropdown ne permet la saisie que d'un seul ascenseur à la fois.
6. Le formulaire comprend également un sélecteur Dropdown qui liste tous les employés de l'entreprise (Table Employees) elle possède une valeur "Aucune" parmi les choix et il s'agit du choix par défaut. Ce dropdown n'est pas lié à la cascade précédente décrite via les dépendances des champs 1 à 5. Ce champ est visible et accessible en tous temps dans le formulaire.
7. Un champ de saisie "Description" permet de décrire la nature de l'intervention requise.
8. Un bouton "Soumettre" permet d'envoyer la demande au centre de traitement

Nouvelle Table "Interventions"

Une nouvelle Table "Intervention" doit être ajoutée au modèle Relationnel dans la base MySQL. Elle doit comprendre les champs suivants afin de persister les données saisies dans le formulaire:

- A. Author (EmployeeID de l'auteur du ticket)
- B. CustomerID
- C. BuildingID
- D. BatteryID (Peut être null)
- E. ColumnID (Peut être null si aucun spécifié)
- F. ElevatorID (Peut être null si aucun spécifié)
- G. EmployeeID (Peut être null si aucun spécifié)
- H. Date de début de l'intervention (doit être null car pas renseignée dans le formulaire. Le champ sera édité plus tard au moment de la prise en charge)
- I. Date et heure de fin de l'intervention (doit être null car pas renseignée dans le formulaire. Le champ sera édité plus tard au moment de la conclusion de l'intervention)
- J. Résultat (Incomplet par défaut)
- K. Rapport (Pour persister le champ Description)
- L. Statut (Pending par défaut)

Création d'un Ticket dans ZenDesk

Au moment de la sauvegarde de la demande d'intervention, le formulaire du site Web crée un nouveau ticket de type "Support" dans ZenDesk. Il ajoute dans la section détail du ticket créé l'ensemble des données saisies du formulaire:

1. L'auteur de la demande
2. Le Client (Company Name)
3. l'ID du Building
4. l'ID de la Batterie
5. l'ID de la Colonne si spécifiée
6. l'ID de l'ascenseur si spécifié
7. L'employé à affecter à la tâche
8. La Description de la demande d'intervention

Création de nouvelles requêtes GET dans l'API REST

Au moment de la sauvegarde de la demande d'intervention, le formulaire du site Web crée un nouveau ticket de type "Support" dans ZenDesk. Il ajoute dans la section détail du ticket créé l'ensemble des données saisies du formulaire:

- GET: Retourne tous les champs de tous les enregistrements de demande de service qui n'ont pas de date de début et qui sont en statut "Pending"

- PUT: Changer le statut de la demande d'intervention à "InProgress" et ajouter une date et heure de début (Timestamp)
- PUT: Changer le statut de la demande d'intervention à "Completed" et ajouter une date et heure de fin (Timestamp)

III. Livrables

Les livrables doivent être déployés dans l'environnement individuel du participant:

1. Application Rails du Participant mise à jour derrière son nom de domaine
2. Adresse du référentiel de code du système d'information
3. Base de Données MySQL mise à jour
4. Nouvelle API REST avec 3 nouveaux points de contact

