

Documentation du Projet Salesforce : Gestion des Événements

Introduction

Ce projet vise à créer une application de gestion d'événements sur Salesforce. L'application permet aux utilisateurs de créer de nouveaux événements, de les visualiser, de les supprimer et de s'inscrire à ces événements. Le projet utilise des composants Lightning Web Components (LWC) pour l'interface utilisateur et des classes Apex pour la logique côté serveur.

Composants du Projet

1. Composants Lightning Web Components (LWC)

- **EventManager** : Ce composant fournit une interface utilisateur permettant de gérer les événements. Les utilisateurs peuvent créer ou modifier des événements, les afficher dans une liste, et effectuer des actions telles que la suppression ou l'inscription.
 - **Interface Utilisateur** :
 - Un formulaire permettant de saisir les détails de l'événement, incluant le nom, le lieu, la description et la date.
 - Une liste des événements existants présentée dans une table avec des options pour supprimer ou s'inscrire.

2. Classes Apex

- **EventController** : Cette classe gère la logique métier pour les opérations relatives aux événements.
 - **Méthodes** :
 - **Récupération des Événements** : Retourne une liste d'événements existants, triée par date.
 - **Création/Modification d'Événement** : Crée un nouvel événement avec les détails fournis ou met à jour un événement existant.
 - **Suppression d'Événement** : Supprime un événement spécifié.
 - **Inscription à un Événement** : Permet à un utilisateur de s'inscrire à un événement spécifique.

Configuration du Composant

Création du Composant

1. **Déploiement** : Le composant **eventManager** est déployé dans l'organisation Salesforce à l'aide de Salesforce CLI ou via l'interface utilisateur Salesforce.
2. **Exposition** : Le composant est exposé sur les pages d'application, de dossier et de page d'accueil de Lightning Experience, ce qui le rend accessible à partir de différents endroits dans l'interface utilisateur.

Ajout du Composant à une Page

1. **Accéder à l'App Builder** : Utilisez l'App Builder de Salesforce pour ajouter le composant à une page Lightning.
2. **Drag-and-Drop** : Faites glisser le composant **eventManager** sur la page de votre choix (page d'application, page d'accueil ou page de dossier).

Fonctionnalités

1. **Création d'Événements** : Les utilisateurs peuvent saisir les détails d'un nouvel événement et le créer en utilisant le formulaire fourni.
2. **Affichage des Événements** : Les événements existants sont affichés dans une table, triés par date.
3. **Suppression d'Événements** : Les utilisateurs peuvent sélectionner un événement de la liste et le supprimer.
4. **Inscription à des Événements** : Les utilisateurs peuvent sélectionner un événement et s'inscrire pour y participer.

Gestion des Événements

Nom

Lieu

Description

Date
Date Time

[Créer/Mettre à jour l'événement](#)

<input type="checkbox"/> Nom	<input type="checkbox"/> Lieu	<input type="checkbox"/> Description	<input type="checkbox"/> Date
<input type="checkbox"/> Soutenance 2	Paris	Soutenance 2024	
<input type="checkbox"/> Soutenance 1	Marseille	Soutenance 2024	

[S'inscrire](#) [Supprimer](#)

Test du Composant

Tester les Fonctionnalités

1. **Création d'Événement :**
 - Remplissez le formulaire de création d'événement.
 - Cliquez sur le bouton de création pour ajouter l'événement.
 - Vérifiez que l'événement apparaît dans la liste des événements.
2. **Suppression d'Événement :**
 - Sélectionnez un événement dans la liste.
 - Cliquez sur le bouton de suppression.
 - Assurez-vous que l'événement est supprimé de la liste après actualisation.
3. **Inscription à un Événement :**
 - Sélectionnez un événement dans la liste.
 - Cliquez sur le bouton d'inscription.
 - Vérifiez que l'inscription est enregistrée (notez que cela peut nécessiter une vérification dans les enregistrements de l'objet `Event_Registration__c`).

Description des Classes et Méthodes

1. Classe Apex : `EventController`

Cette classe gère toutes les opérations de backend liées à la gestion des événements.

- Méthode `getEvents`
 - Fonction : Récupère la liste des événements existants dans Salesforce.
 - Retour : Liste des événements, incluant les champs `Id`, `Name`, `Lieu__c`, `Description__c`, et `Date__c`, triée par date (`Date__c`) de manière décroissante.
 - Utilisation : Utilisée pour afficher les événements dans l'interface utilisateur.
- Méthode `saveEvent`
 - Fonction : Crée un nouvel événement ou met à jour un événement existant.
 - Paramètres : `name`, `location`, `description`, `eventDate` (tous les détails de l'événement à créer ou à mettre à jour).
 - Action : Crée un nouvel enregistrement `Events__c` avec les détails fournis.
 - Utilisation : Utilisée lorsque l'utilisateur soumet le formulaire de création ou de modification d'un événement.
- Méthode `deleteEvent`
 - Fonction : Supprime un événement spécifique.
 - Paramètre : `eventId` (ID de l'événement à supprimer).
 - Action : Supprime l'enregistrement `Events__c` correspondant à l'ID fourni.
 - Utilisation : Utilisée lorsque l'utilisateur choisit de supprimer un événement de la liste.
- Méthode `registerForEvent`
 - Fonction : Permet à un utilisateur de s'inscrire à un événement spécifique.
 - Paramètre : `eventId` (ID de l'événement auquel l'utilisateur souhaite s'inscrire).

- Action : Crée un nouvel enregistrement `Event_Registration__c` liant l'utilisateur à l'événement spécifié.
- Utilisation : Utilisée lorsque l'utilisateur choisit de s'inscrire à un événement depuis la liste.

2. Composant Lightning Web Component : `eventManager`

Le composant `eventManager` gère l'interface utilisateur pour la gestion des événements.

- Fichier `eventManager.js`
 - Variables :
 - `events` : Liste des événements récupérés depuis le serveur.
 - `eventName`, `eventLocation`, `eventDescription`, `eventDate` : Champs pour stocker les valeurs du formulaire de création/édition d'événement.
 - Méthodes :
 - `connectedCallback()` : Méthode de cycle de vie appelée lorsque le composant est inséré dans le DOM. Utilisée pour récupérer les événements au chargement du composant.
 - `handleNameChange()`, `handleLocationChange()`, `handleDescriptionChange()`, `handleDateChange()` : Méthodes de gestion des changements dans les champs du formulaire.
 - `fetchEvents()` : Appelle la méthode Apex `getEvents` pour récupérer et stocker la liste des événements.
 - `saveEvent()` : Appelle la méthode Apex `saveEvent` pour créer ou mettre à jour un événement. Rafraîchit la liste des événements après l'enregistrement.
 - `deleteEvent()` : Appelle la méthode Apex `deleteEvent` pour supprimer un événement. Rafraîchit la liste des événements après la suppression.
 - `registerForEvent()` : Appelle la méthode Apex `registerForEvent` pour inscrire l'utilisateur à un événement. Rafraîchit la liste des événements après l'inscription.
- Fichier `eventManager.html`
 - Structure :
 - Formulaire de Création/Modification : Contient des champs pour saisir le nom, le lieu, la description, et la date de l'événement, ainsi qu'un bouton pour enregistrer l'événement.
 - Liste des Événements : Utilise `lightning-datatable` pour afficher les événements récupérés, avec des boutons pour supprimer ou s'inscrire aux événements.
- Fichier `eventManager.js-meta.xml`
 - Fonction : Définit les paramètres de métadonnées du composant, tels que les cibles de page Lightning où le composant peut être utilisé (pages d'application, d'accueil, et de dossier).

3. Classe Apex de Test : **EventControllerTest**

Cette classe contient des tests unitaires pour la classe **EventController**, afin de vérifier que la logique fonctionne correctement.

- Méthodes de Test :
 - **testGetEvents()** : Vérifie que la méthode **getEvents** retourne la liste correcte d'événements.
 - **testSaveEvent()** : Vérifie que la méthode **saveEvent** crée correctement un nouvel événement.
 - **testDeleteEvent()** : Vérifie que la méthode **deleteEvent** supprime correctement un événement.
 - **testRegisterForEvent()** : Vérifie que la méthode **registerForEvent** enregistre correctement l'inscription d'un utilisateur à un événement.

Conclusion

Ce projet offre une solution complète pour la gestion des événements dans Salesforce en utilisant les technologies LWC et Apex.