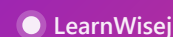


Widgets complexes liés aux données : patterns généraux avec des exemples Kendo et DevExtreme

Guide de la leçon · Guide du lab / de l'examen



Guide de la leçon

Intégrer des widgets de données riches sans rendre le cours dépendant d'une seule bibliothèque tierce. Ce que cela signifie pour un développeur qui intègre avec Wisej.NET — et comment aborder ce module.



Objectif production

Intégrer des widgets de données riches sans rendre le cours dépendant d'une seule bibliothèque tierce.

1. Le problème d'intégration

Des bibliothèques avancées comme Kendo UI et DevExtreme fournissent de puissants grilles, tableaux croisés, tableurs et magasins de données, mais chacune a sa propre API, son propre cycle de vie et son propre modèle d'opérations sur les données.

2. Concepts clés

Concept	Comment l'utiliser
Opérations sur les données	Faire correspondre pagination, tri, filtrage, regroupement et édition à des gestionnaires serveur.
Source de données tierce	Traiter Kendo/DevExtreme comme des exemples d'un pattern d'intégration général.

3. Leçon de production

Kendo UI et DevExtreme sont des exemples avancés, car ils contiennent des widgets riches dotés d'un vrai comportement sur les données. Ils ne définissent pas pour autant l'intégration Wisej.NET. Les mêmes patterns s'appliquent à toute bibliothèque sérieuse de widgets JavaScript : charger les ressources dans le bon ordre, créer le widget dans le bon conteneur, passer les options, câbler les événements, fournir les données, gérer les opérations distantes et exposer une API .NET adaptée à l'application cible. Pour les grilles et les tableaux croisés, la décision de conception la plus importante est le contrat de données. Un tableau croisé en lecture seule peut renvoyer un tableau JSON ou une réponse de type OLAP. Une grille modifiable a besoin d'opérations de chargement, d'insertion, de mise à jour et de suppression. La pagination et le filtrage distants exigent des modèles de requête. La sérialisation de la disposition exige un objet d'état distinct. L'export exige soit une fonctionnalité côté client de la bibliothèque tierce, soit un

pipeline côté serveur. Kendo et DevExtreme ont des API différentes, mais les responsabilités de l'adaptateur Wisej.NET sont les mêmes.

Concevoir d'abord le contrat de données

Un widget riche de grille ou de tableau croisé se définit par les opérations qu'il effectue sur les données : le contrat est donc conçu avant d'écrire le moindre code tiers. Une vue en lecture seule a besoin d'une opération de chargement qui renvoie des lignes. Une vue paginée a besoin de skip et take, ou d'un index et d'une taille de page. Une vue triée ou filtrée a besoin d'une description du tri et du filtre appliqués par le widget. Une vue groupée renvoie des agrégats, et un tableau croisé renvoie une structure organisée par champs de ligne et de colonne. Une vue modifiable ajoute l'insertion, la mise à jour et la suppression, chacune avec une clé et les valeurs modifiées.

Le lab formalise cela par un `GridOperationRequest` typé côté serveur, qui contient skip, take, le tri et le filtre, et par un résultat comportant les lignes et le nombre total. Toute bibliothèque tierce, quel que soit le nom qu'elle donne à ces notions, peut être ramenée à cette requête. Concevoir la requête d'abord en .NET permet au serveur de garder la main sur ce qu'un filtre peut contenir et sur la taille maximale d'une page.

L'état de la disposition, comme la largeur et la visibilité des colonnes, relève d'un contrat distinct, stocké par utilisateur, et n'est jamais mélangé aux opérations sur les données.

Faire correspondre les opérations tierces aux points de terminaison

Le DataSource de Kendo et le CustomStore de DevExtreme diffèrent par la syntaxe, mais tous deux appellent une URL ou une fonction pour le chargement, et appellent à nouveau pour l'insertion, la mise à jour et la suppression lorsque les opérations distantes sont activées. Le rôle du wrapper est de traduire. Pour une source pilotée par URL, l'InitScript oriente la bibliothèque tierce vers l'URL de postback, et le gestionnaire `WebRequest` lit les noms d'arguments de la bibliothèque, les convertit en `GridOperationRequest` et renvoie le résultat sous la forme attendue par la bibliothèque.

Pour une source pilotée par fonctions, les fonctions de l'InitScript appellent une `[WebMethod]` avec des arguments typés et renvoient le résultat attendu à la bibliothèque tierce. Dans les deux cas, le côté serveur est le même gestionnaire avec la même validation, et changer de bibliothèque ne modifie que la traduction dans l'InitScript.

Activer les opérations distantes est le moment où la grille devient une véritable intégration. Jusque-là, la bibliothèque trie et pagine en mémoire et le serveur n'est sollicité qu'une fois ; ensuite, chaque interaction de l'utilisateur est un appel serveur qui doit être rapide et sûr.

Options JSON ou propriétés typées

Kendo comme DevExtreme se configurent au moyen de gros objets d'options, et le moyen le plus rapide d'afficher une grille est d'affecter un tel objet à Options directement depuis C#. C'est la bonne approche pour le prototype de tableau croisé en lecture seule du lab, dont le but est de prouver que le widget fonctionne et que le contrat de données tient.

Un contrôle de grille réutilisable ne peut pas en rester là. Un sac d'options non typé n'a ni validation, ni valeurs par défaut, ni texte d'aide, et ne permet pas de savoir lesquelles de ses cent clés l'application utilise réellement. Des propriétés typées pour les réglages dont l'application dépend, comme la taille de page, le mode modifiable et les définitions de colonnes, donnent au Designer quelque chose à afficher et aux

relecteurs quelque chose à vérifier. Les autres options de la bibliothèque tierce peuvent rester derrière un point d'extension `protected` pour les rares cas qui en ont besoin.

Le compromis oppose la vitesse aujourd'hui à la maintenabilité demain. Prototypiez en JSON, puis promouvez en propriété typée chaque option qui perdure, avant que le contrôle ne soit livré à d'autres développeurs.

Deux bibliothèques, un seul pattern

Kendo et DevExtreme figurent dans ce cours parce qu'ils sont largement utilisés et que leurs différences rendent visible le pattern commun. L'un utilise un `DataSource` avec des réglages de transport, l'autre un `CustomStore` avec des fonctions de chargement et de modification. L'un nomme la pagination `skip` et `take`, l'autre la nomme de façon similaire mais structure autrement l'expression de filtre. Une fois les deux ramenés au même `GridOperationRequest`, les différences se réduisent à quelques lignes de traduction dans chaque `InitScript`.

C'est la leçon à transposer à toute autre bibliothèque. Les responsabilités de l'adaptateur ne changent jamais : charger les ressources dans l'ordre, créer le widget dans le conteneur, passer les options, câbler les quelques événements qui comptent, fournir les données via un contrat maîtrisé par le serveur, gérer les opérations distantes et exposer une API .NET dimensionnée pour l'application. Une équipe capable de le faire pour deux bibliothèques peut le faire pour une troisième sans nouveau cours.

Gardez la connaissance propre à la bibliothèque tierce dans l'`InitScript` et dans la traduction des événements, et gardez le contrat serveur indépendant de toute bibliothèque, pour qu'un changement de licence ou de feuille de route n'impose jamais de réécrire le code applicatif.



Réflexe de production

Gardez clairement séparées la propriété de l'état côté Wisej.NET et le comportement du widget client. Documentez le contrat client/serveur avant de vous appuyer sur l'intégration en production.



Erreur courante à éviter

Copier du JavaScript de démonstration dans les écrans sans documenter le cycle de vie, les ressources, les contrats d'événements, le nettoyage ni les frontières de sécurité.



Objectifs d'apprentissage

À la fin de ce module, vous serez capable de :

- ✓ Concevoir des contrats de données pour des données en lecture seule, modifiables, paginées, filtrées, triées, groupées et croisées.
- ✓ Faire correspondre les opérations de source de données tierces à des gestionnaires de postback ou à des WebMethods Wisej.NET.
- ✓ Comprendre pourquoi les options JSON sont rapides, mais les propriétés .NET typées plus sûres pour les contrôles réutilisables.

- ✓ Utiliser Kendo et DevExtreme comme implémentations de référence des mêmes patterns d'intégration généraux.

Guide du lab / de l'examen

Ce que vous allez construire dans le lab pratique, la démarche recommandée, les livrables attendus et l'auto-vérification avant la remise.



Lab pratique

Construisez deux exemples avancés : un widget de type tableau croisé en lecture seule et un widget de type grille modifiable. L'un peut utiliser la sémantique DataSource de Kendo, l'autre la sémantique CustomStore de DevExtreme.

Démarche recommandée

- ✓ Ouvrez le projet existant du cours d'intégration Wisej.NET et exécutez-le en local.
- ✓ Repérez le composant serveur, le widget client, le fichier de ressources ou le point de terminaison concernés par ce module.
- ✓ Faites une petite modification à la fois et testez-la dans le navigateur avant de passer à la suite.
- ✓ Gardez le JavaScript propre à la bibliothèque tierce dans l'adaptateur ou le wrapper au lieu de le disperser dans les pages.
- ✓ Mettez à jour l'interface pour que l'apprenant voie clairement le succès, l'échec, la progression ou les changements de communication.
- ✓ Prenez des captures d'écran ou des notes qui prouvent que l'intégration fonctionne et qui expliquent le choix de conception.

Livrables attendus

- ✓ Plan d'intégration du tableau croisé en lecture seule
- ✓ Contrat des opérations de la grille modifiable
- ✓ Gestionnaires serveur ou stubs pour load/update/insert/delete
- ✓ Contrats des charges utiles d'événements pour le clic sur une cellule et la mise à jour d'une ligne

Auto-vérification avant la remise

- ✓ Pourquoi les grilles sont-elles plus difficiles que les jauges ?
- ✓ Qu'est-ce qui change lorsque les opérations distantes sont activées ?
- ✓ Pourquoi Kendo et DevExtreme doivent-ils être enseignés comme des exemples, et non comme l'ensemble du cours ?