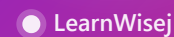


Applications Wisej.NET hybrides, PWA, hors ligne et adaptées à l'appareil

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



Guide de la leçon

Modes connecté/hors ligne, données locales, API de l'appareil, schémas de conflits de synchronisation, scénarios terrain. Ce que cela signifie pour une équipe Wisej.NET avancée — et comment aborder ce module.



Enjeu de production

Modes connecté/hors ligne, données locales, API de l'appareil, schémas de conflits de synchronisation, scénarios terrain.

1. Le problème en production

Certains workflows d'entreprise exigent un fonctionnement sur mobile, tablette, terrain, borne ou hors ligne. Les développeurs Wisej.NET avancés doivent décider quand les schémas Hybrid/PWA sont pertinents et comment préserver un modèle d'application unique sans ignorer les réalités des appareils.

2. Concepts clés

Concept	Pourquoi c'est important
Wisej.NET Hybrid	La prise en charge par le framework de l'exécution de la même application dans le navigateur, en tant qu'application de bureau ou mobile, et avec un accès natif à l'appareil.
mode connecté	Le modèle standard où l'application s'exécute sur le serveur et où l'appareil est un client. C'est le plus simple, mais il nécessite une connexion réseau.
application hors ligne/locale	Exécuter l'application localement sur l'appareil pour qu'elle continue de fonctionner sans connexion, au prix de la gestion des données locales et de la synchronisation.
abstraction de l'appareil	Placer la caméra, le système de fichiers, le GPS et les autres fonctionnalités de l'appareil derrière des interfaces de service, pour que les écrans ne dépendent pas d'une plateforme précise.
SQLite local	Base de données embarquée légère sur l'appareil, qui stocke les données de travail hors ligne et alimente la file de synchronisation.
file de synchronisation	Liste durable des modifications locales en attente d'envoi au serveur au retour de la connectivité, traitées dans l'ordre et relancées sans risque.
résolution des conflits	Règles qui déterminent ce qui se passe quand le même enregistrement a été modifié à la fois localement et sur le serveur pendant la période hors ligne, décidées par le produit et non par le hasard.
shell PWA	Le manifeste de progressive web app et le service worker qui permettent d'installer l'application sur un appareil et de charger son shell sans réseau.

3. Déroulé de la leçon

Le hors ligne est une décision produit

La prise en charge du hors ligne n'est pas une case à cocher. Elle modifie l'identité, la fraîcheur des autorisations, le stockage des données, la gestion des conflits, le moment de l'audit, les procédures de support et le déploiement. Les développeurs avancés limitent le périmètre hors ligne aux workflows qui en ont réellement besoin.

Garder le code de l'appareil derrière des services

La caméra, le scanner, le presse-papiers, les notifications et l'état du réseau doivent être exposés via des services. Le reste de l'application doit appeler des interfaces parlantes pour le métier au lieu de disséminer du code spécifique à l'appareil dans les écrans.

Quatre formes d'application

Wisej.NET Hybrid permet à un même modèle d'application de s'exécuter sous quatre formes. En mode connecté, l'application s'exécute sur un serveur et le navigateur, ou un shell natif autour d'un navigateur, est un client léger ; l'appareil a besoin d'une connexion réseau pour chaque action. En mode distant, un shell natif sur l'appareil se connecte à cette même application serveur, mais accède aux fonctionnalités de

l'appareil via le shell. En mode local, l'application elle-même est packagée avec l'appareil et exécute son processus serveur sur l'appareil ; elle fonctionne donc sans réseau. Le mode hors ligne est le mode local avec des données locales et une frontière de synchronisation vers le système central.

Chaque étape sur cet axe échange de la simplicité contre de l'indépendance. Le mode connecté a une seule copie des données et un seul endroit où appliquer les règles. Le mode hors ligne a une copie sur chaque appareil, et les règles doivent être respectées sur chacun d'eux. La bonne forme est la moins indépendante qui satisfait le workflow. Un dispatcher à son bureau reste connecté. Un technicien dans un sous-sol sans réseau a besoin d'une file hors ligne pour le travail qu'il y termine.

Indiquez la forme dans la note d'architecture avant de concevoir les écrans, car elle détermine où vivent l'identité, les données et les permissions.

Workflows hors ligne et frontières de synchronisation

Un workflow capable de fonctionner hors ligne se conçoit autour de ce que l'appareil a le droit de conserver et de modifier. Le mode Field Technician du lab met en cache dans un magasin SQLite local les ordres de travail attribués au technicien, et seulement ceux-là. Clôturer un ordre de travail hors ligne ne met pas à jour l'enregistrement ; cela ajoute une `OfflineCommand` à une file locale avec un `SyncState` explicite à l'état en attente. L'écran affiche cet état en attente pour que le technicien sache que la clôture n'est pas encore parvenue au bureau.

La frontière de synchronisation est le point où la file est rejouée sur le serveur. Chaque commande est envoyée dans l'ordre, exécutée par le même service serveur que celui qui traite le cas en ligne, et marquée comme synchronisée ou en conflit. Comme le service serveur est le même, chaque règle, contrôle de permission et entrée d'audit s'applique au travail hors ligne exactement comme au travail en ligne. L'appareil ne parle jamais directement à la base de données.

Limitez le périmètre de manière stricte. Chaque entité supplémentaire mise en cache hors ligne représente plus de données sur un appareil qui peut être perdu, plus de cas de synchronisation à tester et plus de conflits à résoudre.

Les capacités de l'appareil derrière des interfaces de service

La caméra, le lecteur de codes-barres, le GPS, les notifications locales et l'état du réseau parviennent à l'application via le shell Hybrid, et chacun se comporte différemment sur un téléphone, une tablette, une borne et un navigateur de bureau. Le lab les masque tous derrière une abstraction `IDeviceServices`. Un écran appelle une méthode comme prendre une photo ou lire un code-barres, et l'implémentation propre à la forme courante décide s'il s'agit de la caméra native, d'un sélecteur de fichiers ou d'un message indiquant que la fonctionnalité n'est pas disponible.

L'abstraction est rentable à trois titres. Les écrans restent identiques d'une forme à l'autre ; un même formulaire sert donc le bureau et le terrain. Les tests peuvent substituer un faux service d'appareil qui renvoie une photo ou un code-barres connus. Et un nouvel appareil ou un nouveau shell ne demande qu'une nouvelle implémentation, pas une fouille de chaque formulaire à la recherche d'appels spécifiques à une plateforme.

Traitez les données produites par l'appareil comme des entrées non fiables. Un code-barres scanné est validé sur le serveur comme n'importe quelle autre valeur, et une position GPS est une indication pour

l'utilisateur, jamais une preuve de l'endroit où le travail a été effectué.

Conflits, permissions obsolètes et sécurité hors ligne

Un conflit survient quand le même enregistrement a été modifié sur l'appareil et sur le serveur pendant qu'ils étaient séparés. Dans le chemin d'échec du lab, le dispatcher annule un ordre de travail pendant que le technicien, hors ligne, le clôture. Aucune des deux versions n'est simplement fausse ; l'écran des conflits de synchronisation affiche donc les deux, les consigne toutes les deux et demande à une personne de trancher. Les règles de résolution, comme les rôles autorisés à passer outre ou la possibilité de clôturer un ordre annulé, sont des décisions produites documentées avant l'écriture du code.

Les permissions deviennent obsolètes hors ligne. Le rôle du technicien a pu être retiré pendant qu'il était déconnecté, et l'appareil conserve encore le jeu de permissions de la veille. La frontière de synchronisation gère ce cas en réévaluant sur le serveur chaque commande en file au regard des permissions actuelles ; les commandes en file d'un utilisateur révoqué sont donc rejetées et auditées au lieu d'être appliquées.

L'appareil rafraîchit aussi ses permissions en cache à chaque reconnexion.

- Les données locales sont chiffrées au repos et effacées à la déconnexion ou après une période de verrouillage.
- Les identifiants en cache expirent, et un appareil volé ne peut plus rejouer de commandes au-delà.
- Chaque commande hors ligne est auditée sur le serveur avec son horodatage d'origine sur l'appareil et son heure de synchronisation.



Réflexe de production avancé

Ne vous arrêtez pas quand l'écran paraît correct. Expliquez à qui appartient l'état, les implications de sécurité, les chemins d'échec et le comportement en production avant de considérer le module comme terminé.



Objectifs d'apprentissage

À la fin de ce module, vous saurez :

- ✓ Expliquer les formes d'application Hybrid connectée, distante, locale et hors ligne.
- ✓ Concevoir des workflows capables de fonctionner hors ligne avec stockage local et frontières de synchronisation.
- ✓ Utiliser les capacités de l'appareil via des abstractions de service explicites.
- ✓ Gérer les conflits de synchronisation, les permissions obsolètes et les compromis de sécurité hors ligne.

Guide du lab / de l'examen

Ce que vous allez construire dans le lab pratique, l'approche suggérée et les livrables attendus.



Lab pratique

Concevez et prototypez un mode Field Technician pour EnterpriseOps. Ajoutez une mise en page adaptée à l'appareil, un cache hors ligne des ordres de travail, une file locale des clôtures, une simulation de synchronisation à la reconnexion et l'affichage des conflits.

Approche suggérée

- ✓ Lisez le problème en production et identifiez le risque d'architecture avant de toucher à l'UI.
- ✓ Créez ou mettez à jour d'abord l'encapsulation par un service, de commande, de modèle ou de composant concernée.
- ✓ Construisez l'écran ou le composant Wisej.NET en suivant les standards de projet établis.
- ✓ Ajoutez une démonstration du chemin d'échec, pas seulement du chemin nominal.
- ✓ Consignez ce qui a changé dans le journal d'apprentissage et mettez à jour la checklist concernée.

Livrables attendus

- ✓ Note d'architecture Hybrid/hors ligne
- ✓ Abstraction des services de l'appareil
- ✓ Modèle de file hors ligne
- ✓ Écran des conflits de synchronisation
- ✓ Checklist d'utilisabilité du mode terrain

Questions de révision

- ✓ Quelles données sont autorisées hors ligne ?
- ✓ Comment les permissions sont-elles rafraîchies ?
- ✓ Que se passe-t-il quand des modifications locales entrent en conflit avec celles du serveur ?



Réflexe de production avancé

Ne vous arrêtez pas quand l'écran paraît correct. Expliquez à qui appartient l'état, les implications de sécurité, les chemins d'échec et le comportement en production avant de considérer le module comme terminé.