

Modernisation de projets vers Wisej.NET 4 et stratégie de migration

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



Guide de la leçon

Planification de la mise à niveau, analyse de compatibilité, migration incrémentale, sécurité face aux régressions. Ce que cela signifie pour une équipe Wisej.NET avancée — et comment aborder ce module.



Enjeu de production

Planification de la mise à niveau, analyse de compatibilité, migration incrémentale, sécurité face aux régressions.

1. Le problème en production

De nombreuses équipes Wisej.NET avancées maintiennent de grandes applications existantes. Elles doivent moderniser la structure des projets, les frameworks cibles, les ressources, les thèmes et les hypothèses de déploiement sans jouer avec la continuité de l'activité.

2. Concepts clés

Concept	Pourquoi c'est important
inventaire de migration	Liste complète des projets, packages, thèmes, ressources et composants personnalisés de l'application existante, pour que rien ne soit découvert à mi-chemin de la mise à niveau.
matrice de compatibilité	Tableau des frameworks cibles, des versions de Wisej.NET et des packages tiers indiquant quelles combinaisons sont prises en charge avant toute modification du code.
mise à niveau incrémentale	Migrer un projet ou une couche à la fois avec un build fonctionnel après chaque étape, au lieu d'une conversion « big bang » impossible à vérifier.
cartographie des ressources	Suivre l'emplacement des images, scripts, thèmes et ressources incorporées avant et après la migration, pour que rien ne casse silencieusement en production.
migration des thèmes	Porter les thèmes et mixins personnalisés vers le format de thème actuel et vérifier que chaque écran s'affiche correctement avec le nouveau moteur.
harnais de régression	Ensemble reproductible de vérifications automatisées et scriptées qui prouve que le comportement existant fonctionne toujours après chaque étape de migration.

3. Déroulé de la leçon

Moderniser, c'est gérer le risque

Une mise à niveau doit produire une meilleure application, mais elle ne doit pas devenir une réécriture incontrôlée. Les développeurs avancés traitent la migration comme une suite de changements vérifiables : compiler, exécuter, comparer le comportement, contrôler les thèmes, vérifier les sessions, revoir le déploiement et mesurer les performances.

L'inventaire avant l'action

Avant de toucher au code, listez le framework cible, les versions des packages, les dépendances du concepteur, les widgets personnalisés, les thèmes, les ressources incorporées, les fichiers de démarrage, les points d'entrée de l'authentification, les cibles de déploiement et les intégrations tierces. Cet inventaire devient la carte de la migration et la checklist de revue finale.



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 :

- ✓ Établir un inventaire de migration pour des projets Wisej.NET 3.x existants ou de style hérité.
- ✓ Identifier les changements de projet, de ressources, de thème, de déploiement et d'API qui exigent des tests explicites.
- ✓ Concevoir un chemin de migration incrémental avec des points de repli.
- ✓ Créer une checklist de régression qui protège le comportement, pas seulement la compilation.

Guide du lab / de l'examen

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



Lab pratique

Créez un dossier de migration pour la TicketOps Console du cours intermédiaire et prolongez-le jusqu'au socle du projet avancé. Incluez l'inventaire de l'état actuel, l'inventaire de l'état cible, la matrice des risques, le plan de régression des écrans et la stratégie de retour arrière.

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

- ✓ Tableau d'inventaire de migration
- ✓ Matrice de compatibilité et de risques
- ✓ Plan de tests de régression pour dix flux clés
- ✓ Plan de retour arrière
- ✓ Note de décision de modernisation

Questions de révision

- ✓ Quel comportement exact doit rester inchangé ?
- ✓ Quelle étape de migration peut être annulée indépendamment ?
- ✓ Quels écrans sont les plus susceptibles de cacher des hypothèses de timing ou d'état ?



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.