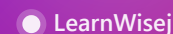


Configuration, sécurité et déploiement

Configuration et secrets · Sécurité et préparation au déploiement



Configuration et secrets

Quels fichiers de configuration examiner avant une mise en production, distinguer debug et production, et gérer en toute sécurité les clés de licence et les secrets.



Objectif de la leçon

Savoir quels fichiers de configuration examiner avant une mise en production, distinguer debug, préproduction et production, et manipuler en toute sécurité les clés de licence, les chaînes de connexion et les autres secrets.

1. Les fichiers de configuration à examiner

Dans un projet Wisej.NET, la configuration peut se trouver à plusieurs endroits. Vous n'avez pas besoin de mémoriser chaque réglage avancé — mais vous devez savoir où regarder avant le déploiement.

Domaine	Ce qu'il faut vérifier	Pourquoi c'est important
<code>Program.cs</code> / démarrage	La fenêtre/page qui démarre, la logique de démarrage propre à l'environnement, la configuration des services.	L'application déployée doit ouvrir le bon point d'entrée.
<code>Web.config</code> / configuration d'hébergement	Indicateur debug, handlers, paramètres du serveur, règles de réécriture, comportement en cas d'erreur.	L'hébergement IIS dépend fortement d'une configuration correcte.
<code>Default.json</code> / JSON de l'application	Point de terminaison Wisej.NET, thème, URL, paramètres de fenêtre/page, références aux ressources statiques.	De mauvaises URL ou de mauvais thèmes peuvent faire échouer une application déployée ou lui donner un aspect cassé.
Thèmes & fichiers statiques	Dossier des thèmes, CSS, images, icônes, fichiers JavaScript.	Les ressources doivent être incluses dans le package de déploiement.
Configuration de la journalisation	Destination des journaux, niveau de journalisation, protection des détails d'erreur.	La production doit permettre le débogage sans exposer les détails internes.

2. Debug, préproduction et production

Les réglages de debug aident pendant le développement, mais les réglages de production doivent être plus sûrs et mieux maîtrisés. Avant la mise en production, examinez l'environnement et assurez-vous que le comportement de debug n'est pas resté activé par erreur.

- **Debug** — erreurs détaillées, tests en local, dépannage facilité ; ne convient pas aux utilisateurs publics.
- **Préproduction (staging)** — un environnement de test proche de la production, utilisé avant la mise en production.
- **Production** — messages d'erreur sans risque, secrets protégés, configuration stable, journalisation activée.



Règle de mise en production sûre

Les utilisateurs doivent voir un message utile ; les développeurs doivent voir les détails techniques dans des journaux sécurisés côté serveur. N'affichez jamais de traces de pile, de chemins de fichiers, de clés de licence ou de chaînes de connexion dans l'interface.

3. Gestion des secrets et des clés de licence

Les valeurs sensibles ne doivent pas être codées en dur dans le code de l'interface, les événements de boutons, les captures d'écran ou le contrôle de source — clés de licence, chaînes de connexion aux bases de données, clés d'API, mots de passe SMTP et identifiants de déploiement.

Pratique risquée	Pratique plus sûre
Coder en dur une clé de licence dans une page ou un événement de bouton.	Utiliser une configuration sécurisée ou des variables d'environnement.
Commiter des chaînes de connexion dans Git.	Garder les secrets hors du contrôle de source ; utiliser les secrets de déploiement.
Afficher aux utilisateurs les détails bruts des exceptions.	Journaliser les détails côté serveur et afficher un message sans risque.
Partager des identifiants de production dans des captures d'écran.	Masquer les secrets ou éviter de les montrer dans les supports de formation.



Vous êtes prêt pour la leçon suivante

Vous savez localiser les principaux fichiers de configuration, séparer le debug de la production et protéger les secrets. Ensuite : authentification et autorisation, journalisation et liste de contrôle de déploiement.

Sécurité et préparation au déploiement

Authentification et autorisation à un niveau junior, journalisation sûre, cibles de déploiement et liste de contrôle obligatoire avant mise en production.



Objectif de la leçon

Comprendre la différence entre authentification et autorisation à un niveau junior, journaliser en toute sécurité et établir une liste de contrôle de déploiement pour IIS, Kestrel ou un hébergement cloud.

1. Authentification & autorisation

L'**authentification** répond à la question *qui est l'utilisateur ?* L'**autorisation** répond à la question *qu'est-ce que cet utilisateur a le droit de faire ?* Le module 9 présente ces notions comme un concept de conception, avant toute implémentation complète de la gestion des identités.

- **Authentification** : connexion, fournisseur d'identité, identité de session, nom d'utilisateur, claims de rôle.
- **Autorisation** : contrôles de rôle, accès aux pages, accès aux commandes, boutons masqués ou désactivés.
- **Exemple** : un Support Agent peut gérer les tickets ; un Admin peut modifier les paramètres de déploiement.
- Ne comptez jamais *uniquement* sur le masquage d'un bouton — la logique côté serveur doit aussi protéger l'action.

```
bool canReviewDeployment =
    currentUser.Role == "Admin" || currentUser.Role == "Team Lead";

btnReviewPackage.Enabled = canReviewDeployment;
lblPermission.Text = canReviewDeployment
    ? "Deployment review available."
    : "Review is restricted.";
```

2. Liste de contrôle de journalisation et de dépannage

La journalisation doit consigner ce qui s'est passé sans exposer de secrets. Un bon journal de dépannage est court, lisible et utile quand quelque chose tourne mal.

- Journalisez le début et la fin des workflows importants.
- Journalisez les échecs de validation et les échecs de revue de déploiement.
- Journalisez les exceptions côté serveur avec suffisamment de détails pour un développeur.
- Ne journalisez pas de mots de passe, de clés de licence ni de données utilisateur privées.
- Ajoutez des horodatages pour que l'ordre des événements soit clair.

3. Cibles de déploiement

Cible	Vérifications de niveau junior
IIS	Publier le dossier de l'application, examiner <code>Web.config</code> , vérifier le hosting bundle si nécessaire, confirmer le pool d'applications, tester l'URL.
Kestrel / auto-hébergement	Confirmer les variables d'environnement, les ports, le reverse proxy, HTTPS et les fichiers de configuration JSON.
Cloud	Confirmer les paramètres de l'application, les secrets, les fichiers statiques, la journalisation et la configuration du slot de déploiement / de la préproduction.

4. Liste de contrôle obligatoire avant mise en production

- `Web.config` examiné et mode debug correctement réglé.
- Clé de licence Wisej.NET gérée de façon sécurisée.
- Fenêtre par défaut/de démarrage, thème et paramètres d'URL vérifiés.
- Thèmes et ressources statiques inclus.
- Destination de la journalisation configurée.
- Plan d'authentification et d'autorisation examiné.
- Les fichiers sensibles ne sont pas téléchargeables publiquement.
- Build Release testé en local avant le déploiement.
- Exigences de la cible de déploiement vérifiées.



Vous êtes prêt pour le lab

Ensuite : regardez la vidéo pas à pas sur la revue de déploiement, réussissez la vérification des connaissances, puis construisez un écran de liste de contrôle de mise en production — vérifications obligatoires et facultatives, notes de déploiement, journal de dépannage et message d'état du package qui change lorsque les vérifications obligatoires sont terminées.