



Exercice Scientifique et technique

Développer un framework partagé

Constant Théo

Sommaire

Plan de la présentation

1. Qu'est-ce qu'un framework ?
2. Le framework interne Linedata
3. Problématique
4. Solutions niveau applicatif
5. Solutions niveau framework
6. Lien avec les matières académiques
7. Conclusion
8. Bibliographie et sources

1. Qu'est-ce qu'un framework ?

Définition

Un framework est une infrastructure logicielle réutilisable qui fournit une base standardisée pour développer des applications.

Caractéristiques clés

- Inversion of Control : c'est le framework qui appelle le code métier
- Composants réutilisables partagés entre plusieurs projets
- Convention over configuration : réduit le code répétitif
- Contrat d'interface stable pour les consommateurs

Exemples connus

- Spring Framework (Java) — injection de dépendances, MVC
- Angular / React (JS) — composants d'interface réutilisables
- Django (Python) — ORM, routage, templates

Différence framework / bibliothèque

Une bibliothèque est appelée par votre code. Un framework, lui, appelle votre code (Hollywood Principle).

2. Le framework interne Linedata

Linedata est un éditeur de logiciels financiers. Son framework interne est utilisé par l'ensemble des applications de gestion d'actifs. Il existe un ancien, et un nouveau framework qui se doivent d'être compatibles.

Configuration XML

- Les développeurs décrivent les écrans et configurent l'application via des fichiers XML
- Le framework lit ces fichiers et génère les interfaces

Composants et configurations partagés

- Nombreux composants UI prêts à l'emploi
- Propriétés applicatives configurables
- Héritage de comportements communs entre projets

Multi-applications

- Plusieurs applications s'appuient sur le même framework
- Chaque application a ses propres méthodes et besoins spécifiques
- Une mise à jour du framework impacte toutes les applications

Mon rôle : concevoir, développer et maintenir ce framework — correction de bugs, développement d'évolutions, assistance technique.

3. Problématique

Comment concevoir, développer et maintenir un framework utilisé par de nombreuses applications en répondant à leurs besoins spécifiques sans impacter ni casser les autres applications ?

Enjeu 1 — Stabilité

Toute modification du framework peut provoquer des régressions dans les applications qui l'utilisent.

Enjeu 2 — Évolutivité

Chaque application a des besoins différents. Le framework doit rester générique tout en permettant la personnalisation.

Enjeu 3 — Maintenabilité

Le framework est partagé par de nombreuses équipes. Il doit être documenté, prévisible et facilement maintenable.

4. Solutions niveau applicatif

Quand un besoin peut être satisfait sans modifier le framework, on agit au niveau de l'application.

Surcharge XML

Principe

Un écran XML défini dans une application peut être surchargé par le client.

Avantages

- Zéro impact sur les autres applications
- Personnalisation des écrans

Personnalisation CSS

Principe

Chaque application peut déposer ses propres fichiers CSS sur son serveur applicatif pour styler l'interface sans toucher au framework. Des variables CSS sont mises à disposition pour modifier directement le style des composants.

Avantages

- Isolation complète : le CSS ne sort pas du serveur de l'application
- Pas de risque de régression cross-application
- Flexibilité de style sans modifier le code source

Désavantages

- Un changement du framework peut rentrer en conflit avec le code css applicatif

4. Solutions niveau applicatif — Assistance technique

Parfois le problème n'est pas un bug du framework, mais une difficulté des développeurs applicatifs à l'utiliser.

Mon rôle d'assistance

- Analyse du besoin et du contexte applicatif
- Identification de la source du problème (applicatif vs framework)
- Aide à l'écriture des fichiers XML
- Aide à la rédaction des styles CSS personnalisés
- Transmission de bonnes pratiques et de la documentation

Processus de triage

Étape 1

Analyse du comportement attendu vs observé

Étape 2

Localisation : le problème est-il dans le XML applicatif, le CSS, ou dans le framework ?

Étape 3

Si applicatif : guider le développeur.

Si framework : ouvrir une correction côté framework.

5. Solutions niveau framework

Quand le problème est intrinsèque au framework, il faut intervenir dessus sans casser l'existant et sans modifier le comportement par défaut pour les autres applications.

Principe du flag d'activation

Toute évolution ou correction de comportement est protégée par un flag booléen dans la configuration XML du composant ou de l'application.

Valeur par défaut : false

Le comportement existant est donc préservé pour toutes les applications qui ne positionnent pas le flag.

- Pas de régression sur les apps existantes
- L'évolution est disponible pour toutes les apps dès la livraison
- Adoption progressive et maîtrisée

Enjeux

- Garder la rétrocompatibilité avec l'ancien framework
- Ne pas modifier le comportement des autres applications (sauf correction de bug sur toutes les applications)
- Satisfaire les besoins de chaque applications
- Éviter un rework important des applications

6. Lien avec les matières académiques

Développement web

- Expertise en CSS
- Framework frontend JS
- Composants réutilisables
- DOM et rendu navigateur

Qualité logicielle

- Tests avec des cas pratiques pour éviter les régressions
- Revues de code et versioning
- Documentation technique

Analyse et compétences sociales

- Analyser les besoins techniques
- Prendre en considération les contraintes de chaque côté
- Communiquer une solution adapté

7. Conclusion

Ce que j'ai appris

- Analyser et comprendre les besoins
- Concevoir des solutions à la fois génériques et personnalisables
- Anticiper l'impact de chaque changement sur un écosystème partagé
- Communiquer avec des équipes aux besoins très différents
- Documenter et maintenir un code utilisé en production par plusieurs applications

Merci pour votre attention.

8. Bibliographie & Sources

Ressources

MDN Web Docs — Documentation CSS, cascade et héritage de styles.

<https://developer.mozilla.org>

Kendo UI for Angular

<https://www.telerik.com/kendo-angular-ui>

Inversion of Control (IoC)

https://en.wikipedia.org/wiki/Inversion_of_control

Hollywood Principle

<https://wiki.c2.com/?HollywoodPrinciple>

Documentation interne Linedata