

# HAPI V8

## Changements importants

Changements majeurs et impacts pour les intégrations

*Document destiné aux intégrateurs et partenaires techniques utilisant HAPI.*

*Ce guide présente les principaux changements introduits dans HAPI v8 ainsi que leurs impacts potentiels sur les intégrations existantes.*

## Introduction

Avec l'arrivée de HAPI v8, plusieurs changements structuraux importants ont été apportés à l'API afin de moderniser la plateforme, améliorer la conformité technologique, renforcer l'intégrité des données et optimiser les performances globales du système.

Ces changements peuvent avoir un impact direct sur les intégrations existantes développées pour HAPI v7.

L'objectif de ce document est de permettre aux intégrateurs de :

- Comprendre clairement les changements introduits dans HAPI v8
- Identifier les impacts potentiels sur leurs intégrations actuelles
- Adapter leur stratégie de migration et leurs développements
- Réduire les risques lors du passage vers HAPI v8

### Pourquoi ces changements ?

- Migration de .NET Core 6 vers .NET 8
- Mise à jour des bibliothèques OData
- Renforcement des validations de typage
- Standardisation des formats de données
- Introduction de mécanismes de protection d'intégrité des données

**Important : Microsoft ne supporte plus .NET Core 6 depuis novembre 2024. Cette migration était essentielle afin d'assurer la sécurité, la stabilité et l'évolution future de la plateforme.**

## Résumé des changements majeurs

Changement	Impact potentiel
Suppression du wrapper value	Désérialisation des réponses
Standardisation des champs numériques	Parsing des nombres
Retrait de certains champs sensibles	Données désormais indisponibles
Migration DateTime → DateOnly	Validation des dates
Révision des messages d'erreur	Adaptation des validations
Réponses POST/PUT/PATCH simplifiées	GET supplémentaire requis
Validation stricte des types	Requêtes auparavant acceptées deviennent invalides
Introduction du RowVersion	Gestion obligatoire de concurrence

## 1. Changement du format des GET par ID

Les requêtes GET(id) ne retournent plus la propriété « value ». Ce changement découle directement de la mise à jour des bibliothèques OData dans le cadre de la migration vers .NET 8.

### Ancien comportement

```
{
  "value": {
    "Id": 123,
    "Name": "Client ABC"
  }
}
```

### Nouveau comportement

```
{
  "Id": 123,
  "Name": "Client ABC"
}
```

**Impact intégrateur** : Les intégrations qui désérialisent explicitement la propriété « value » devront être ajustées.

## 2. Standardisation des champs numériques

Les champs numériques ne retournent plus automatiquement « .0 » lorsqu'il s'agit d'un entier. Les nombres décimaux sont désormais standardisés afin d'être soit des entiers, soit des valeurs contenant minimalement deux décimales.

### Ancien comportement

```
{
  "Amount": 10.0
}
```

### Nouveau comportement

```
{
  "Amount": 10
}
```

**Impact intégrateur** : Les validations strictes de format numérique pourraient nécessiter des ajustements.

### 3. Migration DateTime vers DateOnly

Les propriétés DateTime n'acceptent plus les composantes heure/minute/seconde. Les champs de dates n'ont historiquement jamais sauvegardé l'heure dans Primmo ou HAPI. Les requêtes POST/PUT/PATCH doivent maintenant utiliser des DateOnly.

#### Ancien comportement

```
"StartDate": "2026-05-07T12:12:12.000Z"
```

#### Nouveau comportement

```
"StartDate": "2026-05-07"
```

**Impact intégrateur :** Les intégrateurs devront ajuster leurs modèles et sérialiseurs de dates. Les champs représentant des heures demeurent inchangés puisqu'ils sont stockés comme chaînes de caractères.

### 4. Révision des messages d'erreur

Plusieurs messages d'erreur ont été simplifiés et clarifiés. Le format des erreurs demeure toutefois inchangé.

**Impact intégrateur :** Les systèmes analysant le texte exact des erreurs devraient être validés. Les messages d'erreur continuent d'évoluer afin d'améliorer leur clarté.

### 5. Changement des réponses POST / PUT / PATCH

Les réponses POST/PATCH/PUT ne retournent plus l'objet complet créé ou modifié. Ce changement vise principalement l'optimisation des performances et la gestion des sous-entités utilisées dans Hopem Horizon.

#### Ancien comportement

Retour complet de l'entité créée

#### Nouveau comportement

```
{  
  "Id": 12345,  
  "RowVersion": null,  
  "ActionStatus": 4  
}
```

**Impact intégrateur :** Les workflows de création devront être adaptés afin d'effectuer un GET(id) après création pour récupérer l'entité complète et sa RowVersion.

## 6. Validation stricte des types

HAPI v8 devient beaucoup plus stricte concernant les types envoyés. Les conversions implicites précédemment tolérées ne sont plus acceptées.

### Ancien comportement

```
IsActive: "true"  
Name: 123  
StartDate: ""
```

### Nouveau comportement

```
IsActive: true  
Name: "123"  
StartDate: null
```

**Impact intégrateur** : Les sérialiseurs JSON devront être validés attentivement. Les propriétés booléennes doivent utiliser true, false ou null sans guillemets.

## 7. Introduction du RowVersion

Les entités possèdent maintenant une propriété RowVersion servant à assurer l'intégrité des données et la gestion de concurrence.

### Nouveau comportement

```
{  
  "RowVersion": "AAAAAACLxOw="
```

**Impact intégrateur** : Les requêtes PUT/PATCH devront désormais inclure la dernière valeur RowVersion récupérée via un GET. Une erreur spécifique sera retournée en cas de conflit de version.

*Exemple de scénario évité :*

*Un utilisateur modifie une entité dans Primmo pendant qu'une intégration tente simultanément de modifier cette même entité via l'API. Le mécanisme RowVersion permet de détecter ce conflit afin d'éviter l'écrasement involontaire de données.*

## Recommandations de migration

- Effectuer une révision complète des sérialiseurs JSON
- Réviser tous les traitements GET(id)
- Adapter les workflows de création/modification
- Implémenter la gestion du RowVersion
- Prévoir une phase complète de tests de régression
- Valider tous les traitements de dates et conversions de types
- Documenter les impacts potentiels auprès des équipes techniques et partenaires

## Conclusion

HAPI v8 introduit plusieurs changements structureaux importants visant à moderniser la plateforme, renforcer la stabilité, améliorer la conformité technologique et sécuriser l'intégrité des données.

Bien que certains changements nécessitent des ajustements côté intégrateurs, ces améliorations constituent une base plus robuste et durable pour les futures évolutions de la plateforme.

**Nous recommandons fortement aux intégrateurs de débiter leurs analyses et validations le plus tôt possible afin d'assurer une transition fluide vers HAPI v8.**