

Correction du problème d'importation py3ari/swaggerpy sous Python 3.12

Diagnostic du problème

Le problème que vous rencontrez avec py3ari provient d'une incompatibilité entre la bibliothèque `swaggerpy` (une dépendance de py3ari) et Python 3.12. Plus précisément, le fichier `__init__.py` de `swaggerpy` utilise des imports non relatifs qui ne sont plus supportés de la même manière dans Python 3.12.

L'erreur se produit dans cette ligne :

```
from swagger_model import load_file, load_json, load_url, Loader
```

Sous Python 3.12, cette syntaxe d'import n'est plus valide pour les modules internes d'un package. Python s'attend à ce que les imports relatifs soient explicitement marqués avec un point (`.`).

Solution : Patch local de swaggerpy

La solution consiste à modifier les imports dans le fichier `__init__.py` de `swaggerpy` pour utiliser la syntaxe d'import relatif explicite. Voici les étapes à suivre :

1. Localiser le fichier à modifier

Trouvez le fichier `__init__.py` de `swaggerpy` dans votre environnement virtuel :

```
find /chemin/vers/votre/venv -path "*site-packages/swaggerpy/__init__.py" -type f
```

2. Sauvegarder le fichier original

Avant toute modification, faites une sauvegarde du fichier original :

```
cp /chemin/vers/swaggerpy/__init__.py /chemin/vers/swaggerpy/__init__.py.backup
```

3. Appliquer le correctif

Modifiez le fichier `__init__.py` en remplaçant les imports non relatifs par des imports relatifs. Vous pouvez le faire manuellement ou avec la commande `sed` :

```
sed -i 's/from swagger_model import/from .swagger_model import/g' /chemin/vers/swaggerpy/__init__.py
sed -i 's/from processors import/from .processors import/g' /chemin/vers/swaggerpy/__init__.py
```

Contenu original du fichier

```
#
# Copyright (c) 2013, Digium, Inc.
#

"""Swagger processing libraries.

More information on Swagger can be found `on the Swagger website
<https://developers.helloverb.com/swagger/>`
"""

__all__ = ["client", "codegen", "processors", "swagger_model"]

from swagger_model import load_file, load_json, load_url, Loader
from processors import SwaggerProcessor, SwaggerError
```

Contenu corrigé du fichier

```
#
# Copyright (c) 2013, Digium, Inc.
#

"""Swagger processing libraries.

More information on Swagger can be found `on the Swagger website
<https://developers.helloverb.com/swagger/>`
"""

__all__ = ["client", "codegen", "processors", "swagger_model"]

from .swagger_model import load_file, load_json, load_url, Loader
from .processors import SwaggerProcessor, SwaggerError
```

Précautions et remarques importantes

1. **Mises à jour futures** : Ce correctif devra être réappliqué après chaque mise à jour de py3ari ou swaggerpy, car les fichiers originaux seront réinstallés.
2. **Solution permanente** : Pour une solution plus permanente, vous pourriez envisager de :
3. Forker py3ari et swaggerpy pour créer des versions compatibles avec Python 3.12
4. Utiliser un script post-installation qui applique automatiquement le correctif
5. Utiliser une version spécifique de Python (3.11 ou antérieure) pour ce projet
6. **Compatibilité** : Cette modification ne devrait pas affecter la compatibilité avec les versions antérieures de Python, car la syntaxe d'import relatif explicite est supportée depuis Python 2.5.

Vérification de la correction

Après avoir appliqué le correctif, vous pouvez vérifier que l'importation fonctionne correctement avec un script de test simple :

```
import sys
print(f'Python version: {sys.version}')
try:
    import ari
    print('Import ari réussi')
except Exception as e:
    print(f'Erreur: {e}')
```

Contexte technique

Ce type de problème est fréquent lors de la migration vers Python 3.12, qui a renforcé certaines règles d'importation. Les bibliothèques plus anciennes ou moins maintenues comme swaggerpy utilisent souvent des pratiques d'importation qui ne sont plus recommandées dans les versions récentes de Python.

La modification que nous avons apportée est une correction standard pour ce type de problème, et elle est conforme aux recommandations de la PEP 328 sur les imports relatifs explicites.