

Article

[Kevin Koloska](#) · Nov 10, 2022 6m de lecture

## InterSystems IRIS avec Laravel (via ODBC)

Il a été demandé à plusieurs reprises récemment, comment on peut faire fonctionner [Laravel Framework](#) avec InterSystems IRIS Data Platform. Cela fait un moment que cet [article](#) sur Laravel et InterSystems Caché n ' a pas été publié. Pour le mettre à jour, l ' article donne un bref ensemble d ' instructions sur la façon d ' installer et de configurer un projet Laravel pour une utilisation avec InterSystems IRIS via ODBC.

### Qu ' est-ce que Laravel?

Laravel est un framework PHP basé sur l ' architecture MVC. L ' utilisation de Laravel simplifie et accélère le développement backend tout en créant des applications Web sécurisées modernes. Il est très convivial pour les débutants, largement utilisé dans le monde PHP et tend à être le framework backend le plus populaire selon github.com nombre d ' étoiles mesuré dans cette [vidéo](#). La combinaison de tout cela avec la flexibilité et les capacités de performance fournies par InterSystems IRIS en tant que base de données est considérée comme bénéfique pour les deux mondes.

Ce post est organisé en 4 étapes qui représentent une séquence d ' actions à effectuer pour que la connexion fonctionne. Les façons spécifiques d ' effectuer chaque étape peuvent varier en fonction de la plate-forme. Les commandes ici sont affichées pour Ubuntu 22.02 (x64).

### Gestionnaire de pilotes d ' installation (unixODBC)

Afin de faire fonctionner la connexion, nous devons installer un gestionnaire de pilotes. Les gestionnaires de pilotes les plus couramment utilisés sont 'unixODBC' et 'iODBC'. Ce guide utilise 'unixODBC', qui peut être téléchargé ici <http://www.unixodbc.org/> . Veuillez vous référer à la section « Téléchargement » du site Web pour télécharger et construire. Alternativement, les instructions de construction peuvent également être trouvées [ici](#). Nous utiliserons ici les paquets du gestionnaire de paquets 'apt' pour Ubuntu.

### Installer des packages

Installez le package unixodbc accompagné de libodbcrc2 qui fournit la bibliothèque unixODBC Cursor.

```
sudo apt mise à jour
```

```
sudo apt -y installer unixodbc libodbcrc2 odbcinst
```

### Créer un lien pour la bibliothèque de curseurs

Dans certains cas, il peut y avoir des problèmes avec les dépendances d ' objets partagés après l ' installation d ' unixODBC. Ceci est affiché comme l ' erreur « Impossible d ' ouvrir la bibliothèque du curseur ». Il existe peu de solutions de contournement décrites sur Internet. Afin de résoudre ce problème, nous faisons un lien symbolique vers une bibliothèque souhaitée. Tout d ' abord, nous localisons la bibliothèque:

```
sudo find / -type f -name « libodbc* »
```

Et puis nous créons un lien

```
sudo ln -s /usr/lib/x86_64-linux-gnu/libodbc.so.2.0.0 /etc/libodbc.so
```

## Configuration du pilote ODBC pour InterSystems IRIS

Le pilote ODBC pour InterSystems IRIS peut être obtenu de différentes manières. Par exemple, le pilote ODBC est inclus dans tous les kits IRIS InterSystems. L' autre option serait Portail de distributions sur [wrc.intersystems.com](http://wrc.intersystems.com).

Vous pouvez également trouver des pilotes pour toutes les plates-formes prises en charge ici : <https://intersystems-community.github.io/iris-driver-distribution/>

Téléchargez, décompressez et installez le pilote ODBC :

```
sudo mkdir -p /usr/lib/intersystems/odbc
```

```
cd /usr/lib/intersystems/odbc/
```

```
sudo wget -q https://raw.githubusercontent.com/intersystems-community/iris-driver-dis...
```

```
sudo tar -xzf /usr/lib/intersystems/odbc/ODBC-2022.1.0.209.0-lnxubuntu2004x64.tar.gz
```

```
sudo ./ODBCinstall
```

```
sudo rm -f ODBC-2022.1.0.209.0-lnxubuntu2004x64.tar.gz
```

Après cela, le pilote sera situé dans le dossier suivant `/usr/lib/intersystems/odbc/bin/`.

Des informations supplémentaires sur les pilotes et leur utilisation peuvent être trouvées dans la [documentation](#). Ce guide `libirisodbc6435.so` utilise `libirisodbc6435.so` comme bibliothèque de pilotes.

## Configurer le projet Laravel

Le moyen traditionnel et pratique d'interagir avec une base de données de Laravel serait d'utiliser son package [Eloquent](#). Eloquent est un cartographe relationnel objet (ORM) inclus par défaut dans le framework Laravel. Seuls quelques fournisseurs de SGBD sont [pris en charge prêts à l'emploi](#). Ainsi, afin d'implémenter les spécificités de connexion et de générateur de requêtes SQL pour InterSystems IRIS (via ODBC), du code PHP supplémentaire doit être écrit. Grâce à [@Jean Dormehl](#), cette lacune a été comblée pour InterSystems Caché. Le même pourrait être utilisé pour InterSystems IRIS.

Donc, dans cet article, nous décrivons les étapes de configuration de la connexion pour le projet Laravel existant en utilisant [le paquet jeandormehl/laracache](#), [en supposant que l'installation et la](#)

[configuration de php](#), [composer](#) et Laravel sont effectuées avant cela.

### Installer php-odbc

Assurez-vous que le module php-odbc est installé. Vous pouvez consulter la liste des modules installés avec la commande suivante :

```
php -m | Le grep ODBC
```

Pour installer l'extension php-odbc, utilisez la commande suivante en utilisant une version appropriée de php installée sur votre environnement

```
sudo apt -y install php8.1-odbc
```

### Installer le paquet 'jeandormehl/laracache'

Allez dans le répertoire de votre projet Laravel, installez le paquet et publiez son fichier de configuration.

Compositeur requis jeandormehl/laracache

```
php artisan vendor:publish --tag=isc
```

### Configurer la connexion IRIS

Modifiez votre fichier .env pour qu'il contienne les paramètres nécessaires à la connexion à une base de données. Pour les utilisateurs d'Unix, il devrait ressembler à ceci:

```
DB_CONNECTION=ISC
```

```
DB_WINDSN=
```

```
DB_UNIXDRIVER=/usr/lib/intersystems/odbc/bin/libirisodbc6435.so
```

```
DB_HOST=127.0.0.1
```

```
DB_PORT=1972
```

```
DB_DATABASE=UTILISATEUR
```

```
DB_USERNAME=SYSTEM
```

```
DB_PASSWORD=sys
```

Après avoir édité le fichier .env, vous pouvez trouver utile de vider le cache de configuration de l'application:

```
php artisan config:cache
```

### Usage

Essayons de récupérer quelques données en utilisant notre nouveau package. À titre

d ' exemple, nous allons créer un modèle hérité de `Laracache/Cache/Eloquent/Model` . Juste à des fins de test, nous allons compter le nombre de messages envoyés dans l ' espace de noms activé pour l ' interopérabilité.

```
nano app/Models/EnsMessageHeader.php
```

```
<?php
```

```
espace de noms App\Models ;
```

```
utiliser Laracache/Cache/Eloquent/Model;
```

```
la classe EnsMessageHeader étend le modèle
```

```
{  
  
protected $table = 'Ens.MessageHeader';  
  
$fillable protégé = [  
    'MessageBodyClassName'  
];  
}
```

Pour exécuter une requête, nous pouvons créer une commande de console artisanale comme celle-ci:

```
Nano routes/console.php
```

```
<?php
```

```
utilisez Illuminate/Foundation/Inspiring;
```

```
utiliser Illuminate/Support/Facades/Artisan;
```

```
utilisez App\Models/EnsMessageHeader;
```

```
Artisan::command('iris:test', function () {  
    echo EnsMessageHeader::count() . PHP_EOL;  
});
```

Ensuite, l ' exécution de la commande suivante devrait récupérer le nombre d ' enregistrements

```
PHP artisan iris:test
```

Ce scénario devrait fonctionner pour une large gamme de produits IRIS InterSystems.

[#ODBC](#) [#InterSystems](#) [IRIS](#)

---

URL de la source: <https://fr.community.intersystems.com/post/intersystems-iris-avec-laravel-odbc>