

## GraphQL を Drupal で学ぶ

ハンズオンで Drupal を学ぶ勉強会

**2017 年 12 月 改訂版 v1**

開催日 2017 年 12 月 22 日(金) 19:00-21:30

# 目次

|    |                                                          |    |
|----|----------------------------------------------------------|----|
| 1. | <b>トピックス</b> .....                                       | 2  |
|    | Drupal Global Training Day + Global Sprint               |    |
|    | DrupalCon Nashville 2018                                 |    |
| 2. | <b>今月の Drupal デザインテンプレート</b> .....                       | 3  |
|    | Drupal 8 で使えるテーマ                                         |    |
| 3. | <b>役立つモジュール紹介</b> .....                                  | 3  |
| 4. | <b>ハンズオンで学ぶ</b> .....                                    | 3  |
|    | GraphQL モジュールを既存の Drupal にインストールしてテストしたけど                |    |
|    | ...                                                      |    |
|    | Docker ベースの Decoupled Drupal+GraphQL+React 環境のインストールで OK |    |
| 5. | <b>次回の予定</b> .....                                       | 19 |
| 6. | <b>クレジット、謝辞、ライセンス</b> .....                              | 20 |
|    | クレジット                                                    |    |
|    | 謝辞                                                       |    |
|    | ライセンス                                                    |    |

## トピックス

### Drupal Global Training Day + Global Sprint

Drupal Global Training Day + Global Sprint January 27, 2018

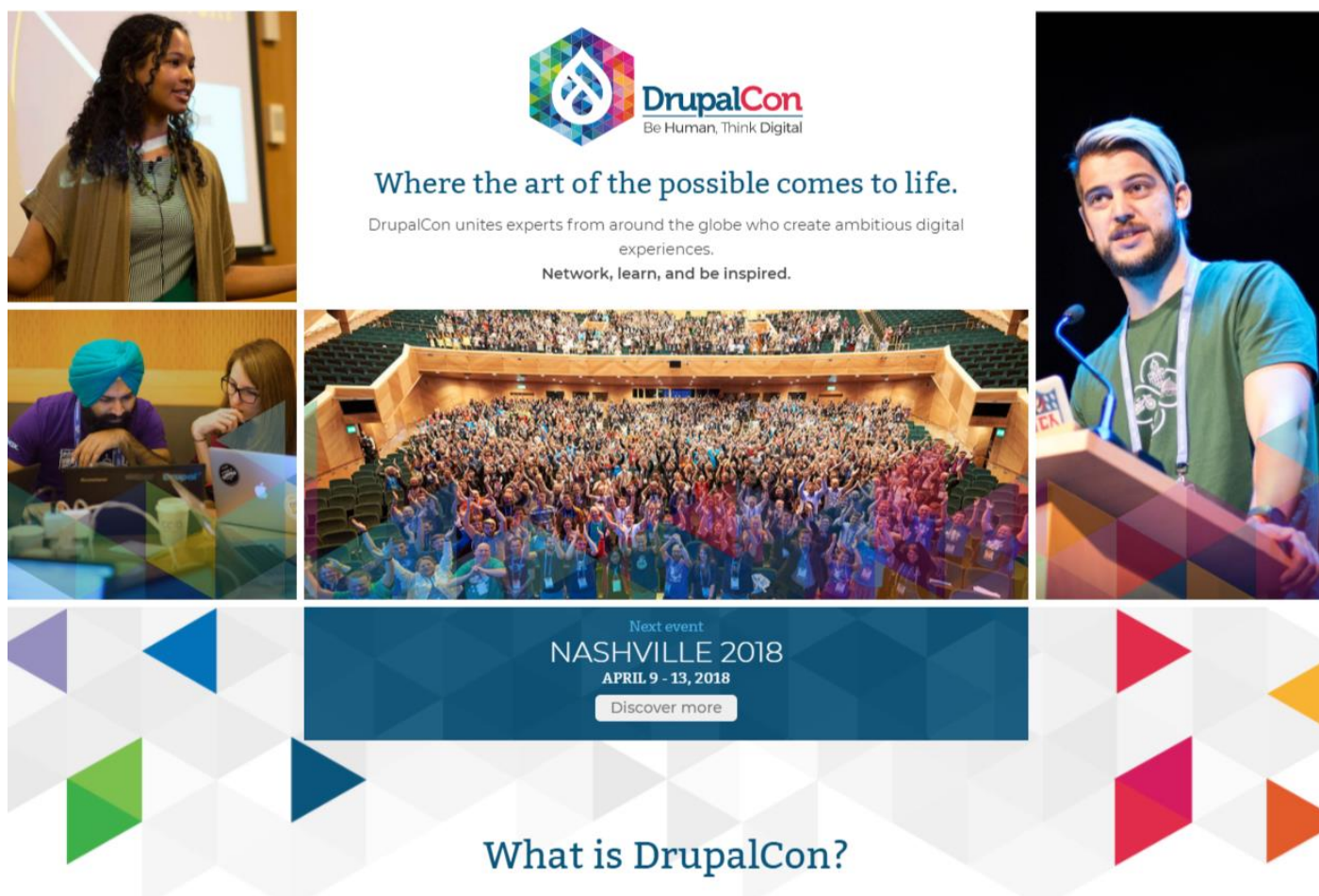
Drupal Association さんからの誘いで、Global Sprint と Training を一緒に開催したらどうかとの連絡があり、開催することになりました。

Sprint では、Decoupled の勉強で、Ruby や JavaScript をフロントエンドにして皆さんとサンプルを作ってみよう、という企画になる予定です。

### DrupalCon Nashville 2018

<https://events.drupal.org/>

CMSLABO LLP でブロンズスポンサー展示出展検討中です。一緒に参加希望される方は、スポンサー特典で無料参加チケットを差し上げます。詳細は、ほどたまで、お問合せください。



The banner features a collage of images: a woman speaking, a man at a laptop, a large audience, and a man at a podium. The central text reads: "Where the art of the possible comes to life. DrupalCon unites experts from around the globe who create ambitious digital experiences. Network, learn, and be inspired." Below this, it says "Next event NASHVILLE 2018 APRIL 9 - 13, 2018" with a "Discover more" button. The bottom text asks "What is DrupalCon?"

DrupalCon  
Be Human, Think Digital

Where the art of the possible comes to life.

DrupalCon unites experts from around the globe who create ambitious digital experiences.

Network, learn, and be inspired.

Next event  
NASHVILLE 2018  
APRIL 9 - 13, 2018  
Discover more

What is DrupalCon?

## 今月の Drupal デザインテンプレート

---

### Drupal 8 で使えるテーマ

- Indiana University Theme  
<https://www.drupal.org/project/iu>

Indiana University Website Style Guide

<https://styleguide.iu.edu/>

事例

<https://libraries.indiana.edu/>

<https://brand.iu.edu/>

- Drupal 8 Admin Theme  
<https://www.drupal.org/project/hostinstantlyadmintheme>

## 役立つモジュール紹介

---

Rules の GitHub にある開発版が、少し動くようになりました。簡単なログイン後の指定 URL に移行する設定が Rules で動きました。詳しくは、その他の機能は未確認です。

## ハンズオンで学ぶ

---

Drupal で GraphQL をやさしく学びます。GraphQL の概念、decoupled Drupal で GraphQL を体験します。

REST API と GraphQL の違いも確認しましょう。

GraphQL モジュールとは

<https://www.drupal.org/project/graphql/>

## GraphQL

[View](#) [Version control](#) [View history](#) [Automated testing](#)

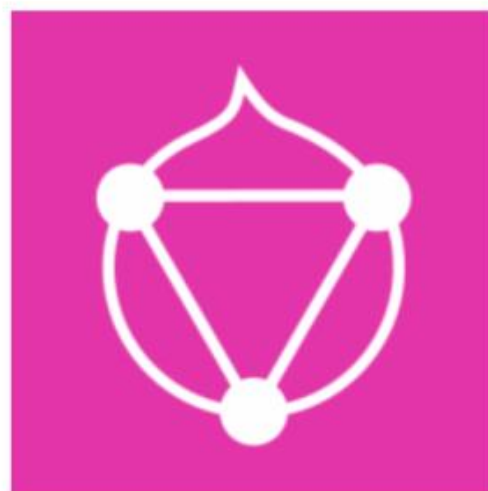
Posted by [fubhy](#) on 20 March 2015, updated 17 October 2017

This module lets you craft and expose a GraphQL schema for Drupal 8.

It is built around <https://github.com/Youshido/GraphQL>. As such, it supports the full official [GraphQL](#) specification with all its features.

You can use this module as a foundation for building your own schema through custom code or you can use and extend the generated schema using the plugin architecture of the contained sub-modules.

For ease of development, it includes the [GraphiQL](#) interface at `/graphql/explorer`.



デモ環境

[example of a fully decoupled React and GraphQL application](#)

# Decoupled Drupal with GraphQL and React

---

## Prerequisites

---

- amaze.io local Docker development environment ([https://docs.amaze.io/local\\_docker\\_development/local\\_docker\\_development.html](https://docs.amaze.io/local_docker_development/local_docker_development.html))
- Composer (<http://getcomposer.org>)
- Yarn (<http://yarnpkg.com>)
- Node (<http://nodejs.org>)

## Usage

---

First, you need to clone this repository.

```
git clone git@github.com:fubhy/drupal-decoupled-app
```

Then, you need to install the dependencies.

```
cd backend && composer install  
cd frontend && yarn install
```

## GraphQL モジュールを既存の Drupal にインストールしてテストしたけど…

# Drupal8 GraphQL モジュールインストール

Drupal8.4.3 に GraphQL モジュールを組み込む

> サンプルテストの実施環境

\* CentOS7.4.1708

\* php7.0.25-1

\* drush 8.1.15

- \* mariaDB 10.1.29
- \* drupal 8.4.3
- \* composer 1.5.6
- \* GraphQL 8.x-3.0-beta2
- \* node 8.9.3
- \* npm 5.5.1
- \* yarn 1.3.2

#### > モジュールの組み込み

- \* ブラウザの機能拡張から、URL を指定してインストール

(<https://ftp.drupal.org/files/projects/graphql-8.x-3.0-beta2.tar.gz>)

※ここで単純に有効化しても、上手く動かないがモジュールインストールだけはしておく

- \* REST サービスを有効化しておく

(HAL/HTTP Basic Authentication/RESTful Web Services/Serialization)

#### > composer のアップデート

...

# alias

alias composer='php /root/.composer/composer.phar'

が無ければ、 ~/.bash\_profile に定義して

# source ~/.bash\_profile

# composer self-update

...

> Youshido/GraphQL の組み込み

これを composer を使って組み込む必要がある

[参考 URL](<https://fgm.gitbooks.io/graphql-for-drupal/content/install/>)

...

```
# cd /var/www/html
```

```
# vi composer.json
```

※先頭に追加

```
{  
  "repositories": [  
    {  
      "type": "vcs",  
      "url": "https://github.com/fubhy/graphql-drupal"  
    }  
  ],  
  "name": "drupal/drupal",  
  "extra": {  
    "installer-paths": {  
      "modules/contrib/{$name}": ["type:drupal-module"],  
      "modules/custom/{$name}": ["type:drupal-custom-module"],  
      "profiles/contrib/{$name}": ["type:drupal-profile"],  
      "themes/contrib/{$name}": ["type:drupal-theme"],  
      "themes/custom/{$name}": ["type:drupal-custom-theme"]  
    }  
  }  
}
```

※既存にあるが確認

```
"extra": {  
  "installer-paths": {  
    "modules/contrib/{$name}": ["type:drupal-module"],  
    "modules/custom/{$name}": ["type:drupal-custom-module"],  
    "profiles/contrib/{$name}": ["type:drupal-profile"],  
    "themes/contrib/{$name}": ["type:drupal-theme"],  
    "themes/custom/{$name}": ["type:drupal-custom-theme"]  
  }  
}
```



```
}  
}
```

```
# composer require drupal/graphql --update-with-dependencies
```

```
--
```

```
--
```

```
> Drupal¥Core¥Composer¥Composer::vendorTestCodeCleanup
```

```
- Installing youshido/graphql (v1.5.2): Downloading (100%)
```

```
> Drupal¥Core¥Composer¥Composer::vendorTestCodeCleanup
```

```
- Installing drupal/graphql (3.0.0-beta2): Downloading (100%)
```

```
--
```

```
--
```

```
Writing lock file
```

```
Generating autoload files
```

```
> Drupal¥Core¥Composer¥Composer::preAutoloadDump
```

```
> Drupal¥Core¥Composer¥Composer::ensureHtaccess
```

```
...
```

```
> インストール確認
```

```
...
```

```
# ls -d vendor/youshido/graphql modules/contrib/graphql
```

```
modules/contrib/graphql/ vendor/youshido/graphql/
```

```
...
```

```
> GraphQL モジュールの有効化
```

\* ブラウザの機能拡張から、GraphQL を有効化

(GraphQL/GraphQL Core)

\* 環境設定⇒Web サービスの中に、GraphQL が現れる

Schemas⇒Explorer で QUERY テスト用の画面が出てくる

※最初エラーが出る場合があるが、リロードすると出てくる

# テスト React サイト構築

> リポジトリ clone

...

# cd /var/www

# mkdir graphql-test

# cd graphql-test

# git clone https://github.com/drupal-graphql/drupal-decoupled-app.git

...

> node の最新化

...

※古いバージョンの削除

# yum remove -y nodejs npm

# curl -sL https://rpm.nodesource.com/setup\_8.x | bash -

# yum install -y nodejs gcc-c++ make

...

> yarn の導入

...

※グローバルにインストール

```
# npm install -g yarn
```

...

> フロントのインストール

...

※backend は Docker で Drupal 環境を含む形で作られているが

今回は、Drupal を立ち上げたので、フロントのみ使用

```
# cd /var/www/graphql-test/drupal-decoupled-app/frontend
```

```
# yarn install
```

```
yarn install v1.3.2
```

```
[1/5] Validating package.json...
```

```
--
```

```
--
```

```
[5/5] Building fresh packages...
```

```
Done in 61.33s.
```

...

> 実行

...

```
# cd /var/www/graphql-test/drupal-decoupled-app/frontend
```

```
# yarn run dev
```

...

旨くつながらない

> ターゲットの変更

```

```
# cd /var/www/graphql-test/drupal-decoupled-app/frontend
# vi .env.defaults
// Copy this file to '.env' to specify environment specific configuration.
//コメント API="http://decoupled.backend.docker.amazee.io/graphql"
↓
API="http://(Drupal の起動している IP)/graphql"
```
```

でも動かない

## Docker ベースの Decoupled Drupal+GraphQL+React 環境のインストールで OK

以下のサイトのインストール手順で動きました。

<https://github.com/drupal-graphql/drupal-decoupled-app>

## Decoupled Drupal with GraphQL and React

### Prerequisites

- amazee.io local Docker development environment ([https://docs.amazee.io/local\\_docker\\_development/local\\_docker\\_development.html](https://docs.amazee.io/local_docker_development/local_docker_development.html))
- Composer (<http://getcomposer.org>)
- Yarn (<http://yarnpkg.com>)
- Node (<http://nodejs.org>)

### Usage

First, you need to clone this repository.

```
git clone git@github.com:fubhy/drupal-decoupled-app
```

Then, you need to install the dependencies.

```
cd backend && composer install
cd frontend && yarn install
```

Then, you need to boot the backend container.

```
docker-compose up -d
docker-compose exec --user drupal drupal bash
```

Once connected to the container, you can now install Drupal.

```
drush si config_installer -y --account-name=admin --account-pass=admin
```

Now you can create some content (basic page or article) and run the frontend application

## 手順の概要

最初に以下の環境をインストールします

- Docker 環境をインストール

- 以下の Drupal 8 の Docker 環境をインストールします。

amazee.io local Docker development environment

[https://docs.amazee.io/local\\_docker\\_development/local\\_docker\\_development.html](https://docs.amazee.io/local_docker_development/local_docker_development.html)

- Composer のインストール <http://getcomposer.org>

- Yarn のインストール <http://yarnpkg.com>

- Node のインストール <http://nodejs.org>

- Decoupled Drupal with GraphQL and React 環境のソースを GitHub からクローン

git clone [git@github.com:fubhy/drupal-decoupled-app](https://github.com/fubhy/drupal-decoupled-app)

backend に移行して composer でインストール、frontend に移行して yarn でインストール

cd backend && composer install

cd frontend && yarn install

バックエンドの Docker を起動させます

docker-compose up -d

```
$ sudo docker-compose up -d
```

```
Starting decoupled.backend.docker.amazee.io
```

docker-compose exec --user drupal drupal bash

```
$ sudo docker-compose e
```

```
xec --user drupal drupal bash
```

```
? ? drupal@decoupled.backend.docker.amazee.io:~/public_html/backend/web$
```

docker に接続した状態

```
? ? drupal@decoupled.backend.docker.amazee.io:~/public_html/backend/web$ ls -la
total 88
drwxr-xr-x  7 root root 4096 Dec 22 08:31 .
drwxr-xr-x  6 root root 4096 Dec 22 08:29 ..
-rw-r--r--  1 root root  385 Dec 11 13:22 autoload.php
drwxr-xr-x 12 root root 4096 Dec 22 08:30 core
-rw-r--r--  1 root root 1025 Dec 11 13:22 .csslintrc
-rw-r--r--  1 root root  350 Dec 11 13:22 .editorconfig
-rw-r--r--  1 root root  206 Dec 11 13:22 .eslintignore
-rw-r--r--  1 root root   36 Dec 11 13:22 .eslintrc
-rw-r--r--  1 root root   41 Dec 11 13:22 .eslintrc.json
-rw-r--r--  1 root root 3774 Dec 11 13:22 .gitattributes
-rw-r--r--  1 root root 7866 Dec 11 13:22 .htaccess
-rw-r--r--  1 root root 2088 Dec 11 13:22 humans.txt
-rw-r--r--  1 root root  549 Dec 11 13:22 index.php
drwxr-xr-x  3 root root 4096 Dec 22 08:31 modules
drwxr-xr-x  3 root root 4096 Dec 22 08:30 profiles
-rw-r--r--  1 root root 1596 Dec 11 13:22 robots.txt
drwxr-xr-x  4 root root 4096 Dec 11 13:22 sites
drwxr-xr-x  2 root root 4096 Dec 11 13:22 themes
-rw-r--r--  1 root root  848 Dec 11 13:22 update.php
-rw-r--r--  1 root root 4555 Dec 11 13:22 web.config
```

Docker の Contanier に接続できたら、drush コマンドで Drupal をインストールします。

```
drush si config_installer -y --account-name=admin --account-pass=admin
```

ユーザー名 admin パスワード admin で作成

Frontend のアプリケーションを起動させる前に、ローカルでサイトが稼働するように、  
/etc/hosts に以下の設定を追加します。docker の IP アドレスを指定しています。

```
172.17.0.2    decoupled.backend.docker.amazee.io
```

Frontend のアプリケーションを実行させます

```
cd frontend
```

```
yarn run dev
```

```
yarn run v1.2.1
```

```
$ react-scripts dev
```

```
Happy[1]: Version: 4.0.0. Threads: 3
```

```
Happy[1]: Version: 4.0.0. Threads: 3
```

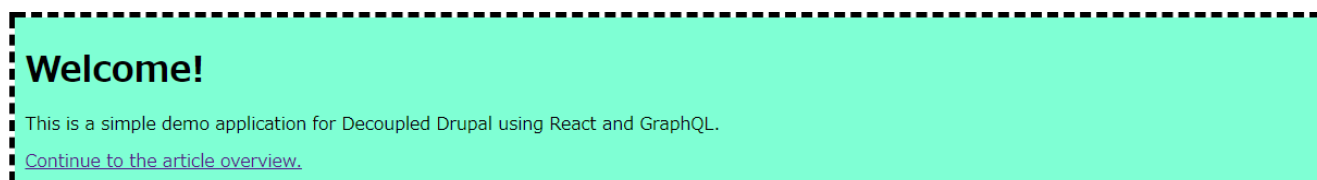
```
Starting the development server ...
```

```
Happy[1]: All set; signaling webpack to proceed.
```

```
Happy[1]: All set; signaling webpack to proceed.
```

```
webpack building...
```

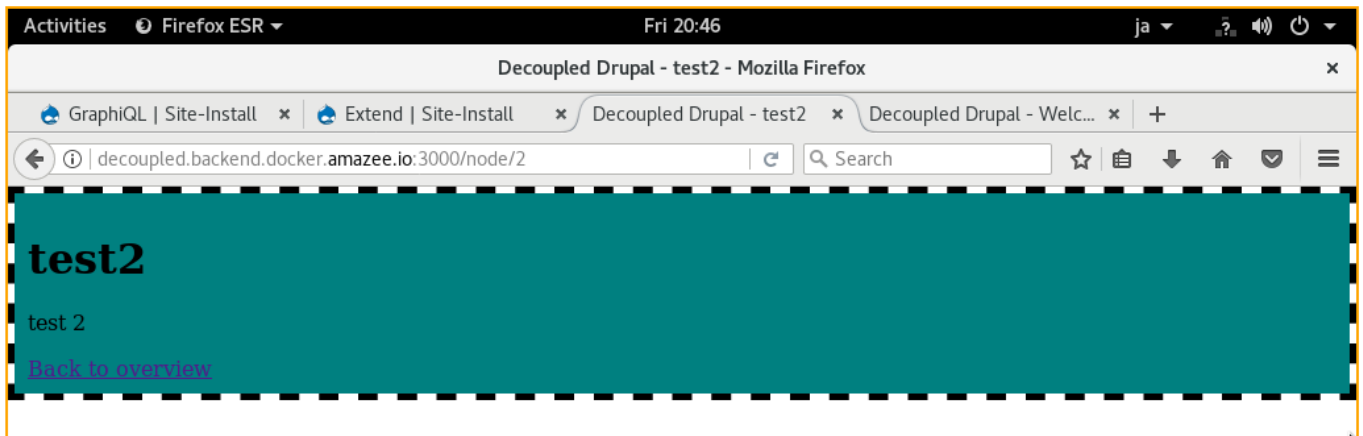
サーバー内の GUI 環境で、ローカルな（172.17.0.2:3000 もしくは decoupled.backend.docker.amazee.io:3000）接続すると以下の画面が表示されます



**continue to the article overview** をクリックすると、article のコンテンツページが表示されます。事前にコンテンツタイプ（Article）でページを作成した状態での表示サンプルです。



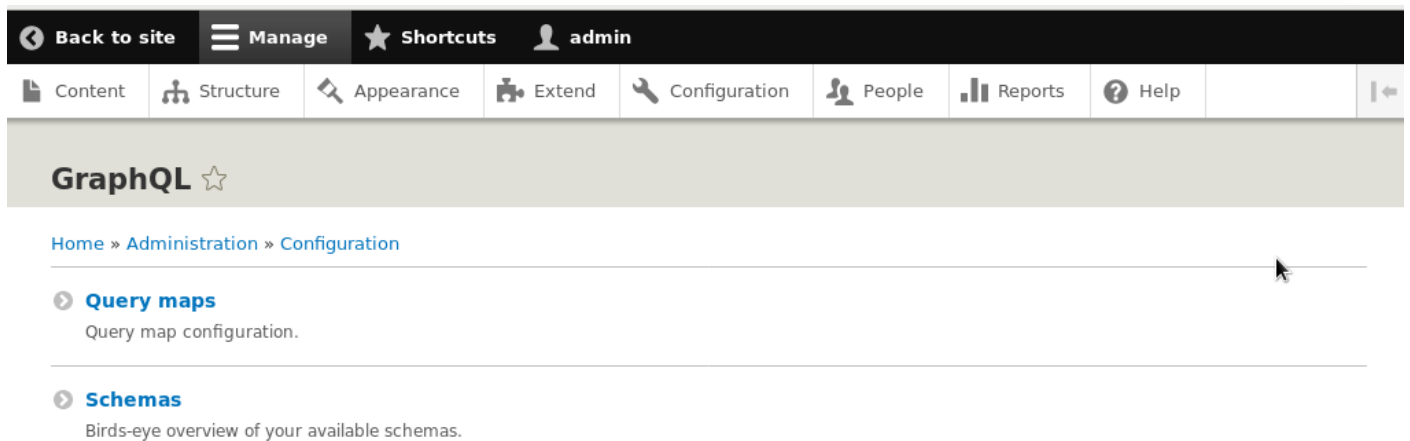
**test2** というページをクリックして表示させると、そのページが表示されます。



以上のサンプルは、シンプルですが、GraphQL で Drupal のフロント画面を作れる事例として。その他、Drupal のバックエンドで、GraphQL モジュールなどの機能を見てみると。

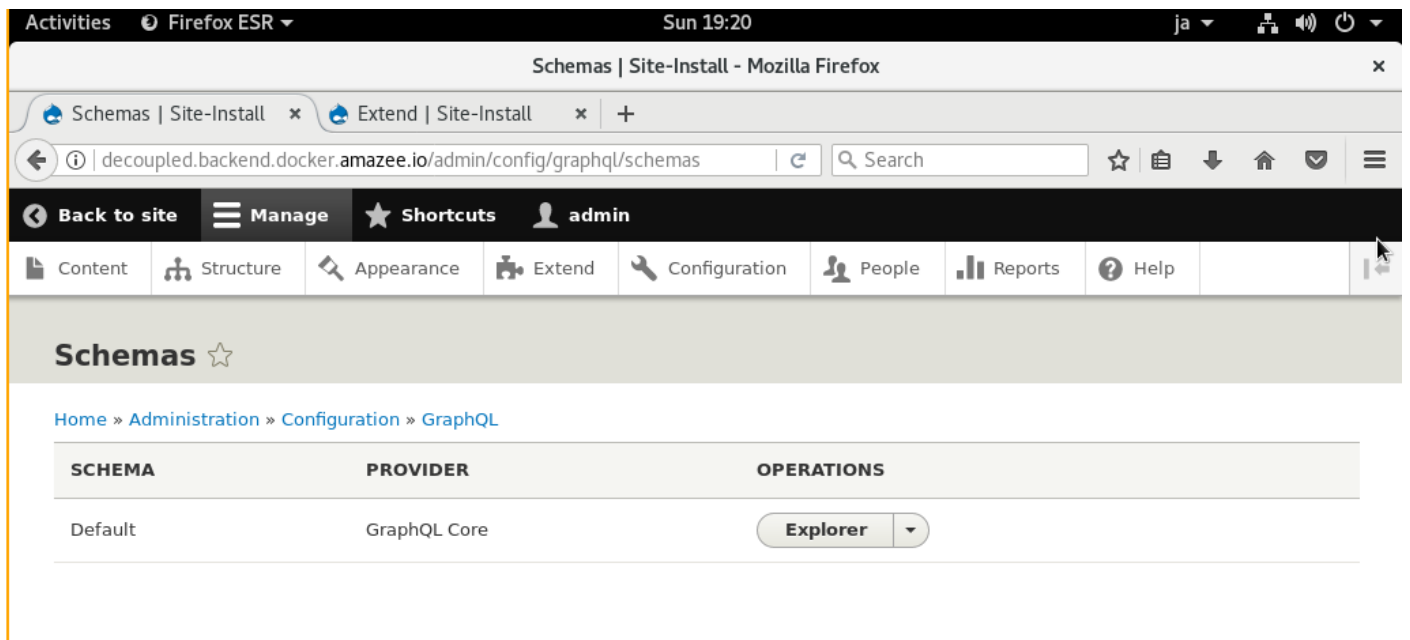
## GraphQL モジュールの概要

### クエリーマップおよびスキーマの選択

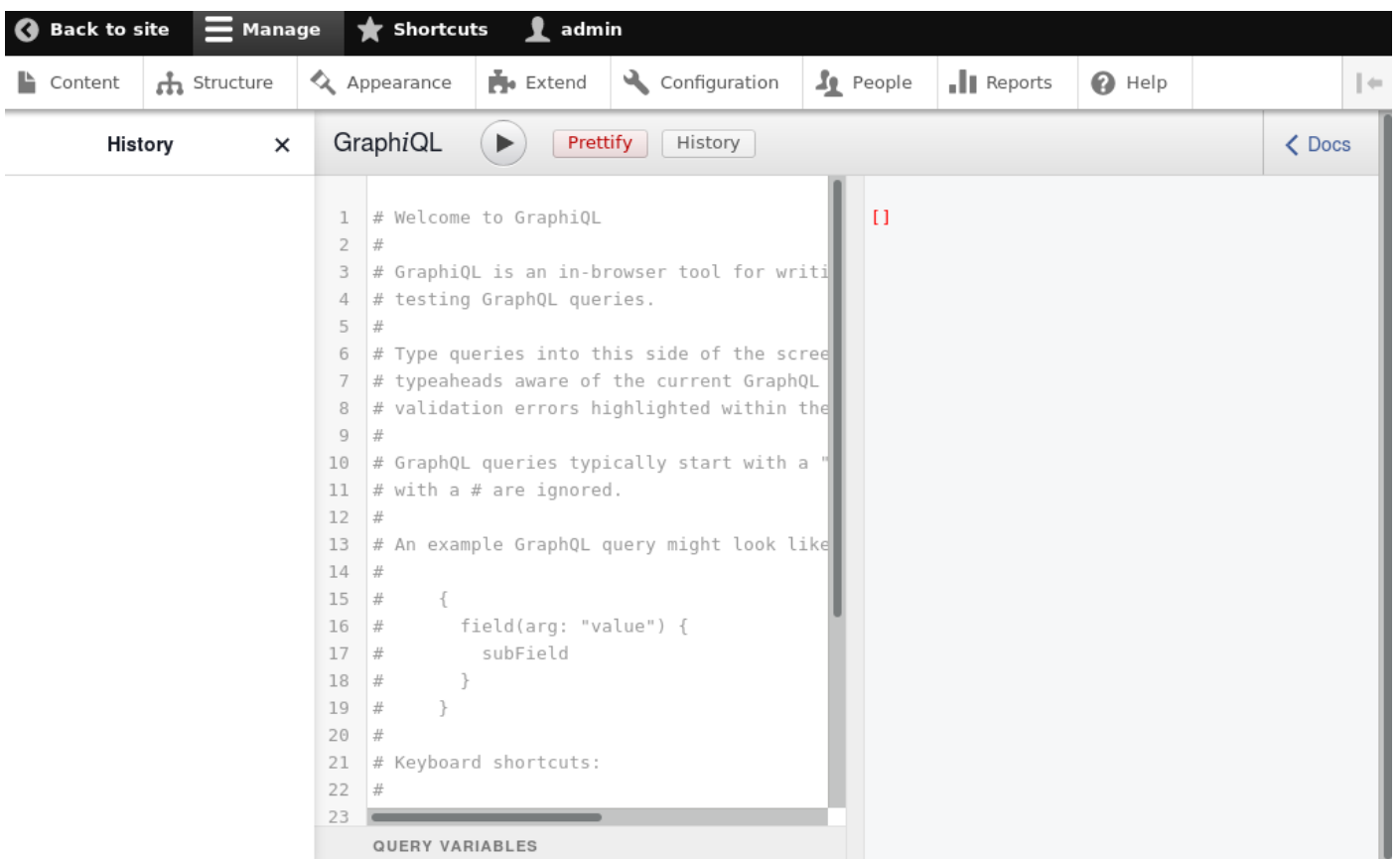


スキーマを選択し、Explorer を選択





## 表示サンプル



## GraphQL モジュールのスキーマの Voyager を選択

Back to site
Manage
Shortcuts
admin

Content
Structure
Appearance
Extend
Configuration
People
Reports
Help

# GRAPHQL VOYAGER

## Type List

No Description

**MenuLink**

No Description

**MenuLinkContent**

The 'Custom menu link' entity type.

**MenuLinkContentMenuLinkContent**

The 'Custom menu link' bundle of the 'Custom menu link' entity type.

**NodeArticle**

The 'Article' bundle of the 'Content' entity type.

**NodePage**

The 'Basic page' bundle of the 'Content' entity type.

**Shortcut**

The 'Shortcut link' entity type.

+
-
RESET

Back to site
Manage
Shortcuts
admin

Content
Structure
Appearance
Extend
Configuration
People
Reports
Help

# GRAPHQL VOYAGER

## Type List

The time that the custom block was last edited.

**defaultLangcode: Int**  
A flag indicating whether this is the default translation.

**entityAccess( operation: String! ): Boolean**

**entityBundle: String**

**entityChanged: String**

**entityCreated: String**

**entityId: String**

**entityLabel: String**

**entityLanguage: Language**

### BlockContent

<<interface>>

|                             |                               |
|-----------------------------|-------------------------------|
| changed                     | Int                           |
| defaultLangcode             | Int                           |
| entityAccess                | Boolean                       |
| entityBundle                | String                        |
| entityChanged               | String                        |
| entityCreated               | String                        |
| entityId                    | String                        |
| entityLabel                 | String                        |
| entityLanguage              | Language                      |
| entityOwner                 | User                          |
| entityPublished             | Boolean                       |
| entityTranslation           | Entity                        |
| entityType                  | String                        |
| entityUri                   | Uri                           |
| entityUuid                  | String                        |
| id                          | Int                           |
| info                        | String                        |
| langcode                    | FieldBlockContentLangcode     |
| revisionCreated             | Int                           |
| revisionId                  | Int                           |
| revisionLog                 | String                        |
| revisionTranslationAffected | Int                           |
| revisionUser                | FieldBlockContentRevisionUser |
| type                        | FieldBlockContentType         |
| uuid                        | String                        |
| implementations             |                               |
| BlockContentBasic           |                               |

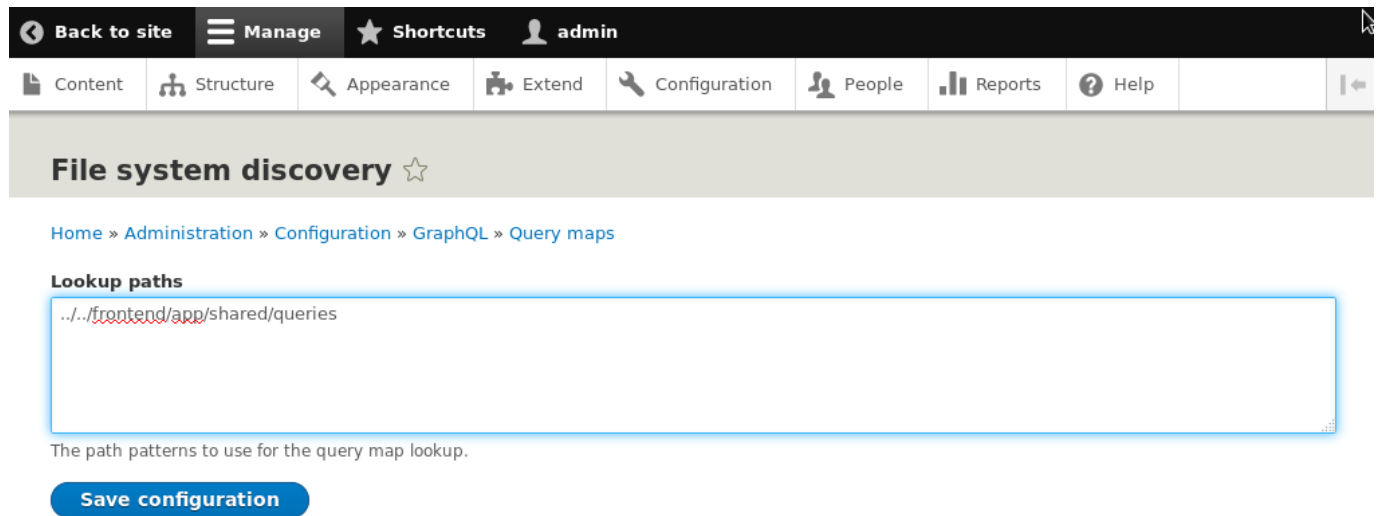
RootQuery: blockContentById

BlockContent: entityTranslation

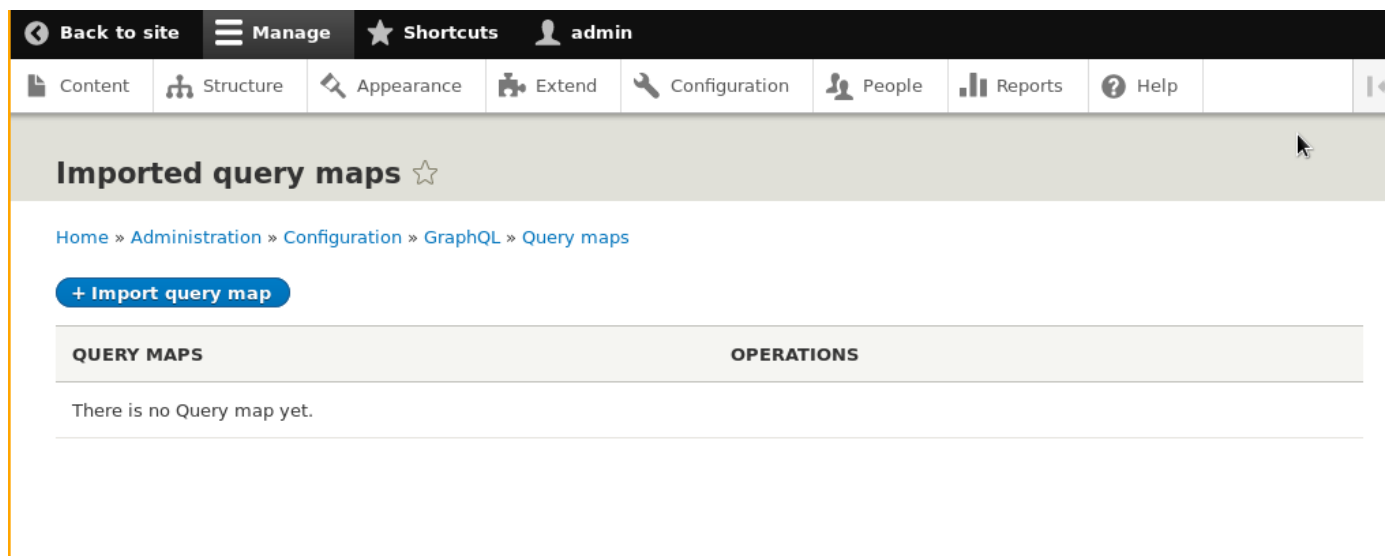
+
-
RESET

## クエリーマップ関連

ファイルシステムディスカバリー（クエリーマップを調べる-探索で使うパスパターン）



Json 形式のクエリーマップをインポート



実際にさくらのクラウドでハンズオン！

## 次回の予定

---

1月17日(水)、かながわ県民センター7階 **705号室**です。内容は未定です。

12月25日(月) 久松町区民館で、MailChimp, HubSpot, CiViCRM などの CRM 分野です。記事「Using lots of different tools? Do it all in Drupal instead!」を参考に

<https://www.mydropwizard.com/blog/using-lots-different-tools-do-it-all-drupal-instead>

1月29日(月) 久松町区民館ですが、1月27日に Global Sprint があるので2月になる予定。

## クレジット、謝辞、ライセンス

---

### クレジット

このマニュアル作者は、Gennai3 株式会社の程田和義です。

お問合せ 電子メール [hodota@gennai3.co.jp](mailto:hodota@gennai3.co.jp) 電話 044-220-1588

### 謝辞

本マニュアル作成は、主に以下のサイトを参考にしました。心より感謝いたします。

出典 : <https://Drupal.org>

### ライセンス

Drupal は Dries Buytaert による登録商標です。その他本マニュアルで使われている製品および名称については、それぞれの所有者の商標または登録商標です。