

Headless Drupal en pratique

Simon Morvan



<https://www.drupal.org/u/garphy>



@simonmorvan



<http://www.icilalune.com/>

Headless ?

[hédləs]

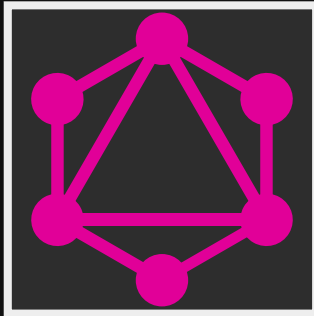
Pourquoi ?



Pourquoi Drupal ?

- Modélisation du contenu
- Richesse fonctionnelle
- Multilingue
- E-commerce
- Communauté
- Back-office
- ...

UX





Canaux multiples

- Web
- Applications
- Diffusion automatique
- IoT, AR, ...

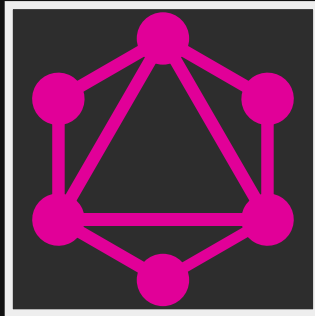
Différents composants

- Backend: stable, pérenne (investissement) ;
- Frontend(s): évolutif et rapidement à jour.

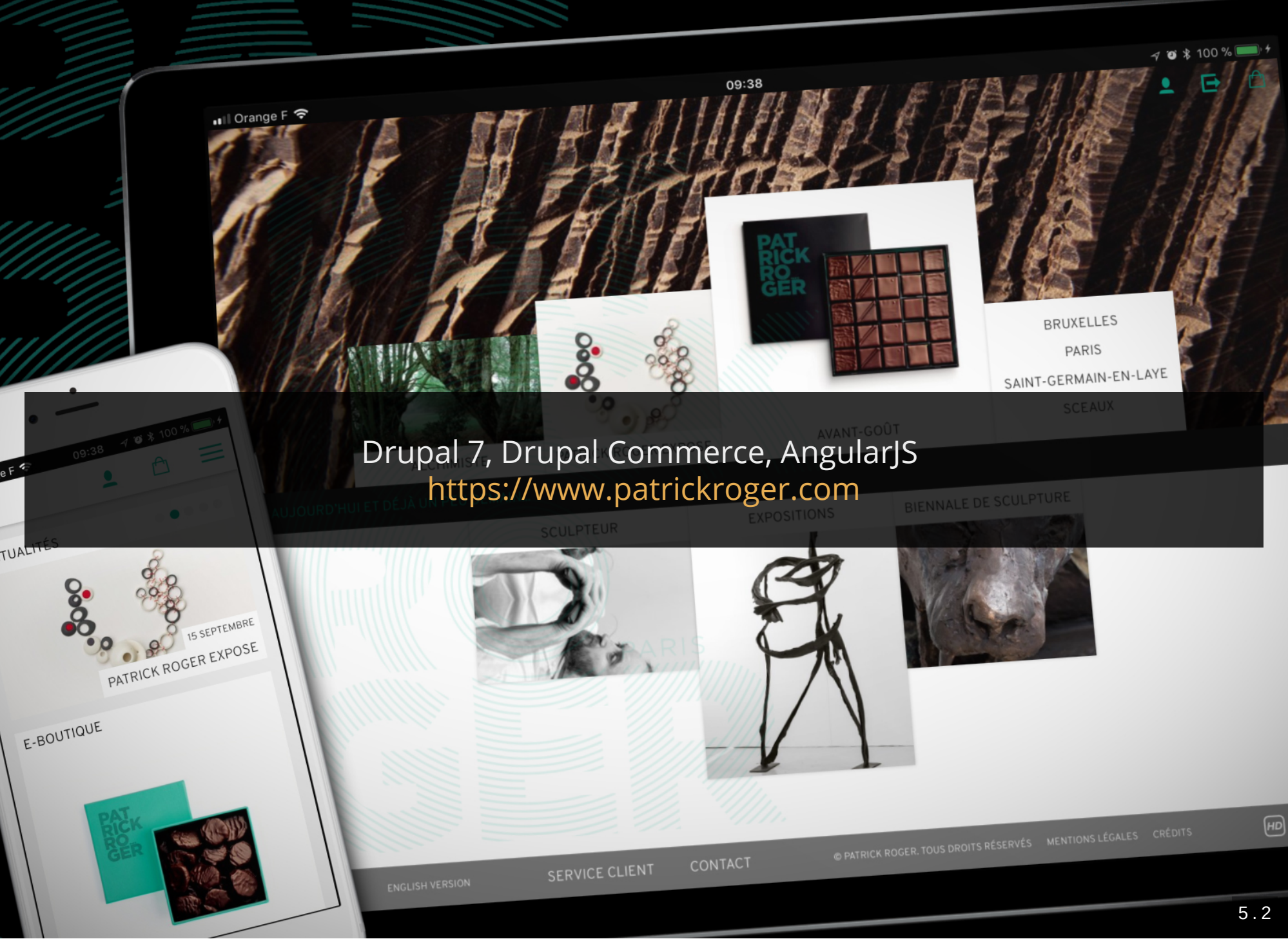
Cycles de développement différents.

Choix technologique

- Angular
- Ember
- React
- Vue.js
- ...



Exemples



Drupal 7, Drupal Commerce, AngularJS
<https://www.patrickroger.com>

Drupal 7, Node.js, AngularJS
<https://www.entrainement-athle.fr>



Drupal 8, Angular 4, ThreeJS
<https://www.icilalune.com>

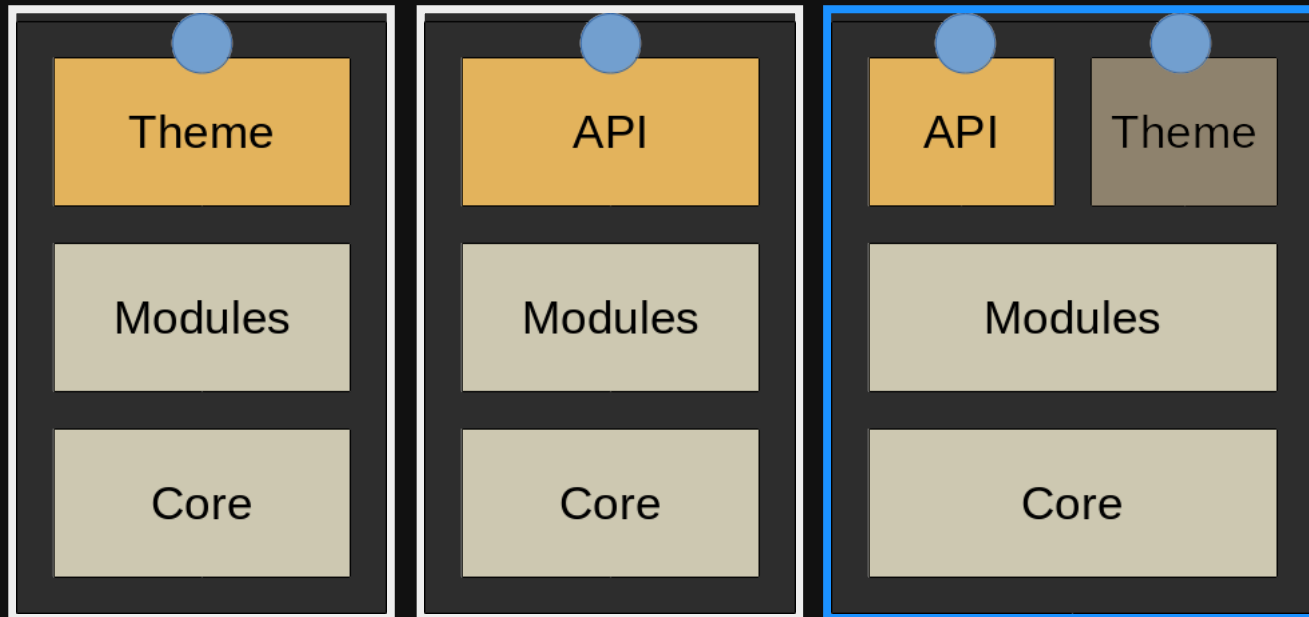
Comment ?

Drupal

- Modèle de contenu
- Logique, Contrôle
- Interface visuelle

Headless Drupal

- Modèle de contenu
- Logique, Contrôle
- ~~Interface visuelle~~ → API



Solution SaaS

- Contentful
- Backendless
- GraphCMS
- Tchop
- ...

Warning !

L'équation headless :

un projet = deux projets
(voire plus)

Composants

- Drupal (+PHP, MySQL)
- Angular (+Javascript, Sass, Webpack)
- Node.js (+Javascript, NPM)

Conséquence(s)

- Prévoir plus de temps
- Hébergement disparate
- Autorise un recrutement plus large
- Permet un certain parallélisme

#1

API

API pour Drupal 8

- REST (core)
- JSON API
- GraphQL

Pour Drupal 7 : Services (& Services Entity)
ou RestWS

REST

- Core
- Entity normalizer
- XML, JSON, ...
- Create, Retrieve, Update, Delete
- Pas de collections → Views

Example

/node/3?_format=json

```
{
  "nid": [
    {
      "value": 3
    }
  ],
  "uuid": [
    {
      "value": "a75a24f1-241a-4e77-9e2e-903e5f5f8563"
    }
  ],
  "vid": [
    {
      "value": 3
    }
  ],
  "langcode": [
    {
      "value": "en"
    }
  ],
  "type": [
    {
      "target_id": "container",
      "target_type": "node_type",
      "target_uuid": "1e009f03-6fc7-456f-8ce6-e2d911860b59"
    }
  ],
  "revision_timestamp": [
    {
      "value": "2017-09-14T14:25:09+00:00",
      "format": "Y-m-d\\TH:i:sP"
    }
  ],
  "revision_uid": [
    {
      "target_id": 1,
      "target_type": "user",
      "target_uuid": "a3015686-d17a-44db-8231-c4a77fab44b9",
      "url": "/user/1"
    }
  ],
  "revision_log": [],
  "status": [
    {
```

Views REST export

Displays

REST export [+ Add](#)

Display name: [REST export](#)

TITLE

Title: [None](#)

FORMAT

Format: [Serializer](#) | [Settings](#)

Show: [Entity](#)

FIELDS

The selected style or row format does not use fields.

FILTER CRITERIA

[Add](#) ▼

[Content: Published \(= Yes\)](#)

SORT CRITERIA

[Add](#) ▼

[Content: Authored on \(desc\)](#)

PATH SETTINGS

Path: [/rest/node](#)

Authentication: [No authentication is set](#)

Access: [Permission](#) | [View published content](#)

HEADER

The selected display type does not use header plugins

FOOTER

The selected display type does not use footer plugins

NO RESULTS BEHAVIOR

The selected display type does not use empty plugins

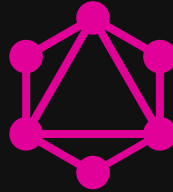
PAGER

Items to display: [Display a specified number of items](#) | [10 items](#)

GET /rest/node?_format=json

Des issues qui restent

- File
- Test coverage
- Cache
- Authentication
- ...



GraphQL

A query language for your API

<http://graphql.org/>

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

Beta!

Query

```
{  
  hero {  
    name  
    height  
    mass  
  }  
}
```

Result

```
{  
  "hero": {  
    "name": "Luke Skywalker",  
    "height": 1.72,  
    "mass": 77  
  }  
}
```

{json:api}

A specification for building APIs in JSON

«*your anti-bikeshedding tool*»

<http://jsonapi.org/>

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

JSON API

- CRUD
- Collection (filtrage, pagination)
- Inclusion des références

JSON API ↔ Drupal

- Attributs → Fields
- Relations → Entity Reference
- Collections → ~~Views~~

Typage

- Type: Entity Type + Bundle

Notation: node--article

/jsonapi/node/article

- Deux content-types: deux types
- Pas de collection pour plusieurs types

<https://www.drupal.org/node/2886540>

Entité

/jsonapi/node/article/42

```
{
  "data": {
    "type": "node--article",
    "id": "42",
    "attributes": {
      "title": "First node",
      "created": 2147483647
    },
    "relationships": {
      "author": {
        "data": { "type": "user--user", "id": "9" }
      }
    },
  },
}
```

Collection

/jsonapi/node/article

```
{
  "data": [
    {
      "type": "node--article",
      "id": "42",
      "attributes": {...}
    },
    {
      "type": "node--article",
      "id": "84",
      "attributes": {...}
    }
  ]
}
```

Inclusion

`/api/node/product/4?include=field_category`

```
{
  "data": {
    "id": 4,
    "type": "node--product",
    "attributes": ...,
    "relationships": {
      "field_category": {
        "data": {
          "type": "taxonomy_term--category",
          "id": "3"
        }
      }
    }
  },
  "included": [{
    "id": 3,
    "type": "taxonomy_term--category",
    "attributes": {
      "name": "Drupalcamp goodies",
      ...
    }
  }]
}
```


Filtre

```
GET /jsonapi/node/article?  
  filter[published][condition][path]=status  
&filter[published][condition][value]=1  
&filter[published][condition][operator]=%3D // URL encoded "="
```

```
GET /jsonapi/node/product?  
  filter[published][condition][path]=field_category.id  
&filter[published][condition][value]=42  
&filter[published][condition][operator]=%3D // URL encoded "="
```

- filtrage
- également tri, pagination

Documentation

<https://www.drupal.org/docs/8/modules/json-api>

Implémentation



Liste

```
loadNews(){
  this.http.get('/jsonapi/node/article?sort=-created')
    .subscribe(result => {
      this.nodes = result.data;
    });
}
```

```
<ul>
  <li *ngFor="let node of nodes" (click)="loadNode(node.id)">
    {{node.attributes?.title}}
  </li>
</ul>
```

Detail

```
loadNode(id){  
  this.http.get(['/jsonapi/node/article',id].join('/'))  
    .subscribe(result => {  
      this.node = result.data;  
    });  
}
```

```
<div>  
  <h1>{{node?.attributes?.title}}</h1>  
  <div [innerHTML]="node?.body?.value">  
  </div>  
</div>
```

Distributions

- Contenta
<http://www.contentacms.org/>
- Reservoir
<https://github.com/acquia/reservoir>

→ JSON API

#1

API

#2

Navigation

SPA

Single Page Application

```
<html>
  <body>
    <script src="app.js"></script>
  </body>
</html>
```

Deep linking

Chaque contenu devrait avoir sa propre URL

- Google
- Bookmarks
- Précédent/Suivant
- Réseaux sociaux

Techniques

- ~~URL Fragment~~

`https://www.escapefactory.fr/#!/fr/news/halloween`

`Bowling`

- History API

`https://www.icilalune.com/fr/articles`

`history.pushState({}, "Contact", "/fr/contact")`

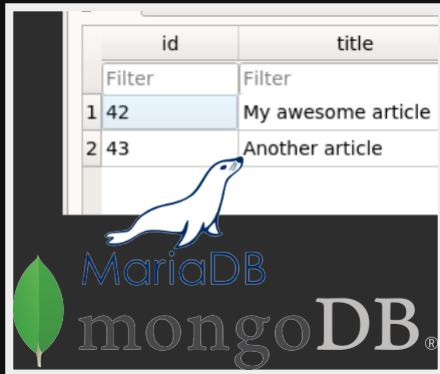
Routing

```
const appRoutes: Routes = [  
  { path: 'news/:id', component: NewsDetailComponent },  
  { path: 'news', component: NewsListComponent },  
  { path: '', redirectTo: '/news' },  
  { path: '**', component: PageNotFoundComponent }  
];
```

Fonctionnement par motif

- 😊 Définit les états
- 😊 Modélise les paramètres
- 😞 Couplage avec l'URL

From scratch



- Base de donnée: Clé primaire 42
- API: /api/article/42
- Front: <https://fromscratch.wtf/news/42>

Drupal backend



▼ URL PATH SETTINGS

- ☒ Generate automatic URL alias
Uncheck this to create a custom alias below. [Configure URL alias patterns.](#)

URL alias

/articles/2015/10/anthologie

Specify an alternative path by which this data can be accessed. For example, type "/about" when writing an about page.

Content Management System

L'URL désigne le contenu.

L'URL doit être stable.

L'URL est une propriété du contenu.

Problématique

<https://www.icilalune.com/fr/articles/2017/04/objectif-montreal>

<https://www.icilalune.com/fr/breaking-news>

- Activer le bon contexte ("newsDetail")
- Charger le bon contenu (nid = 42)

Résolution de l'URL

- Contextes indépendants de l'URL
- Service (API) spécialisé

Routing, revisité

Pour Angular et React : UI-Router

<https://ui-router.github.io/>

Etats de l'application

```
export const STATES: any[] = [
  {
    name: 'front',
    component: FrontPageComponent,
    params: {
      id: {
        type: 'string'
      }
    }
  },
  {
    name: 'newsList',
    component: NewsListComponent
  },
  {
    name: 'newsDetails',
    component: NewsDetailComponent,
    params: {
      id: {
        type: 'string'
      }
    }
  },
]
```

Path request

<https://backend.icilalune.com/api/path?path=/fr/articles/2017/04/objectif-montreal>

Module Services

```
$provided_path = $request->query->get('path');
$provided_path_request = Request::create($provided_path);

$route = $router->matchRequest($provided_path_request);

if(preg_match('/^entity\.[a-zA-Z0-9_]+\canonical$/', $route['_route'])) {

    $entity_key = preg_replace('/^entity\.[a-zA-Z0-9_]+\canonical$/', '\1',
        $route['_route']);
    $entity = $route[$entity_key];

    $result['entity'] = [
        'type' => $entity->getEntityTypeId(),
        'id' => $entity->id(),
        'uuid' => $entity->uuid(),
        'bundle' => $entity->bundle(),
    ];
}
```

Path response

<https://backend.icilalune.com/api/path?path=/fr/articles/2017/04/objectif-montreal>

```
{
  "language": "fr",
  "frontPage": false,
  "entity": {
    "type": "node",
    "id": "33",
    "uuid": "4affd605-30ee-4169-972c-b4eeb8e0cb89",
    "bundle": "article",
  },
}
```

Routing, end

```
resolvePath(path:string){
  httpClient.get('/api/path', {
    params: new HttpParams().set('path', path),
    headers:new HttpHeaders({'Accept':'application/json'})
  }).map(pathInfo => {
    if (pathInfo.frontPage) {
      return {
        name: 'front',
        params: {
          id: pathInfo.entity.uuid
        }
      };
    }
    if (pathInfo.entity && pathInfo.entity.type === 'node') {
      switch (pathInfo.entity.bundle) {
        case 'article':
          return {
            name: 'newsListDetails',
            params: {
              id: pathInfo.entity.uuid
            }
          };
      }
    }
    return {'name': '404'};
  }).subscribe(state => {
    this.uiRouter.stateService.go(state.name, state.params);
  });
}
```

A la recherche d'une

Solution générique

- Côté Drupal

https://www.drupal.org/project/services_path

⚠ Experimental

- Côté Front : dépend du framework

Module pour angular et ui-router "bientôt"

#2

Navigation

#3

Crawlers

icilalune à montreal



Tous

Maps

Actualités

Images

Vidéos

Plus

Paramètres

Outils

Environ 277 000 résultats (0,55 secondes)

Objectif Montréal ! | Ici la Lune

<https://www.icilalune.com/fr/articles/2017/04/objectif-montreal> ▼

Bientôt une base lunaire au Canada... Afin de développer notre activité d'agence et chercher un relais de croissance international, l'Agence va créer une filiale ...



garphy

@simonmorvan

Abonné



#chocolat



Inauguration Atelier Patrick Roger | Ici la Lune

Après un terrible incendie qui a détruit son atelier, Patrick Roger inaugure aujourd'hui le nouveau visage de son espace de création et de production. L'Agence sera re...

icilalune.com

18:07 - 10 oct. 2017

1 J'aime



1



Crawlers

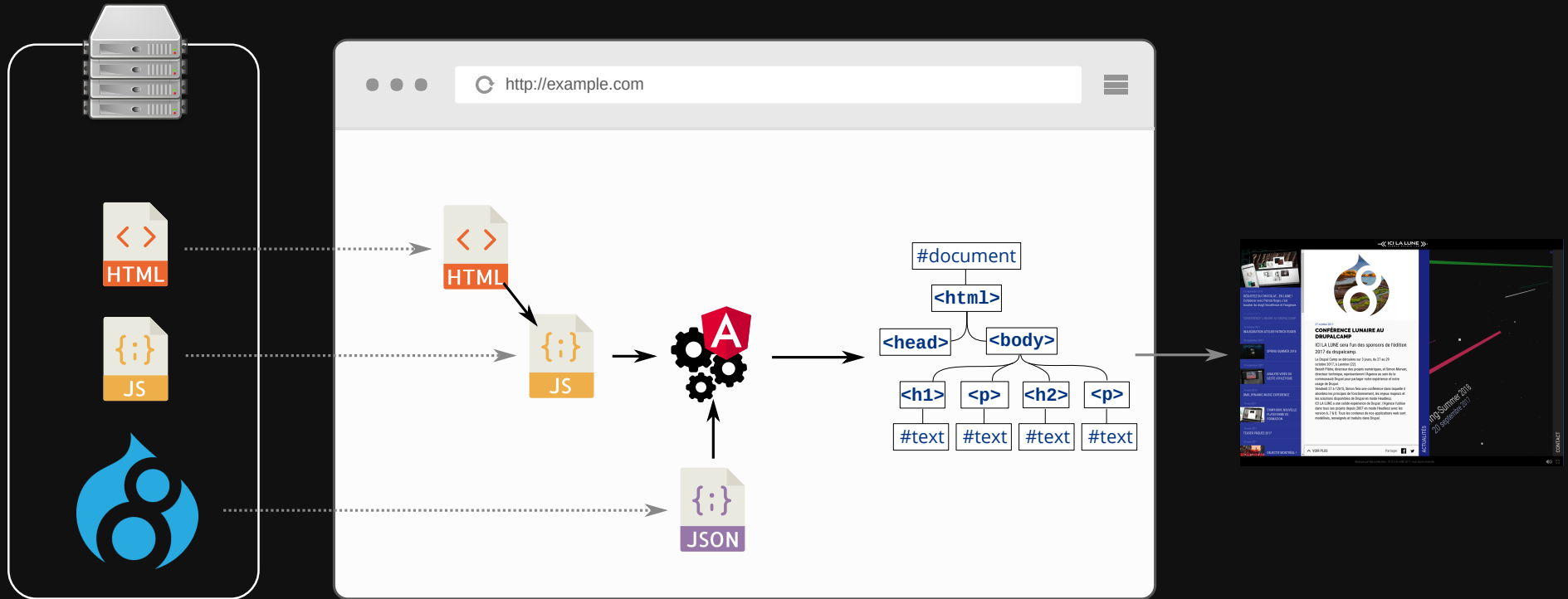
- Google
- Facebook
- Twitter
- ...

A partir de l'URL

cURL

```
$ curl http://www.headlessisfun.com/  
$ curl http://www.headlessisfun.com/about  
$ curl http://www.headlessisfun.com/article/42
```

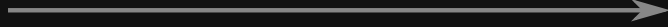
```
<html>  
  <body>  
    <script src="app.js"></script>  
  </body>  
</html>
```

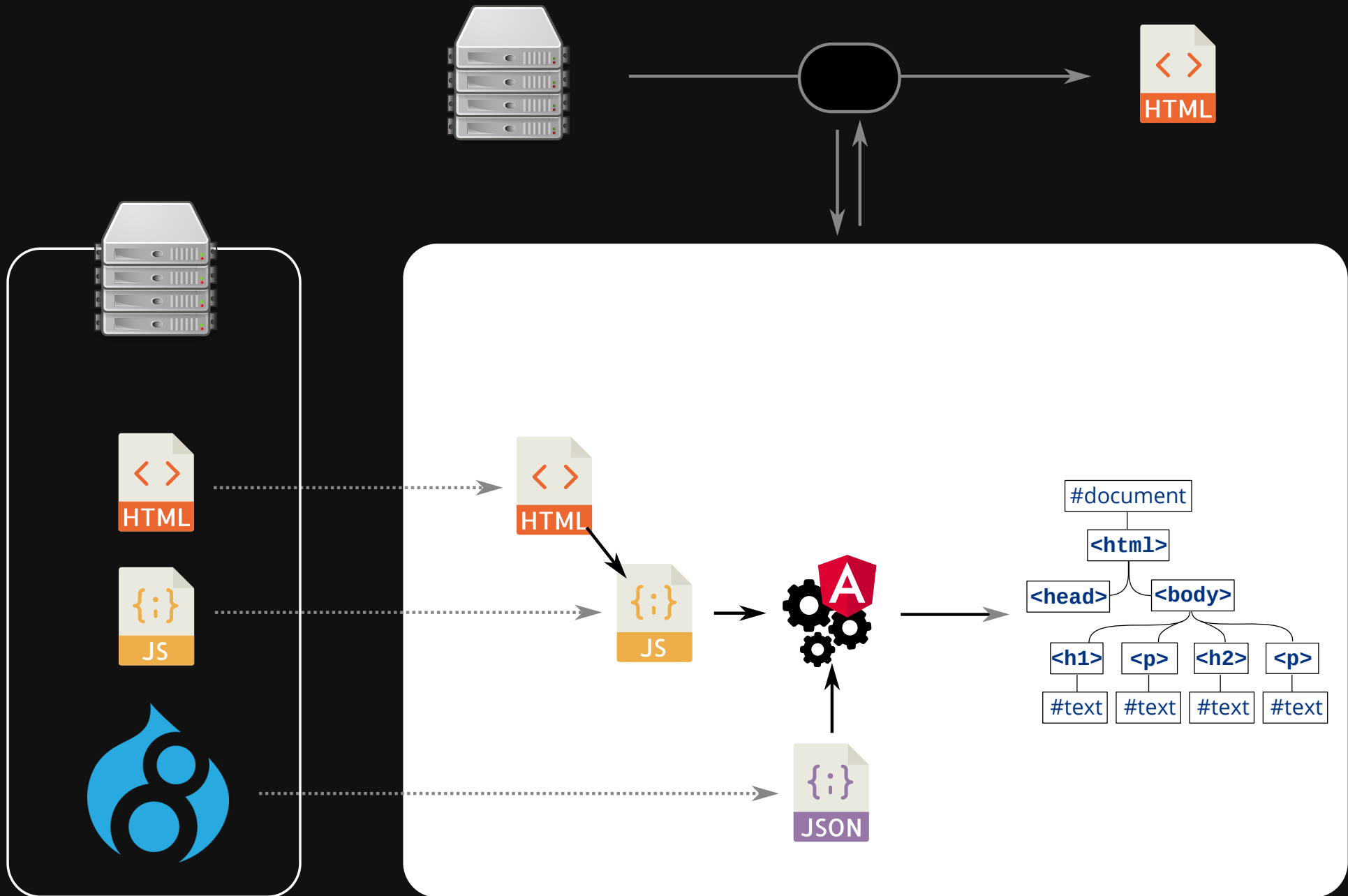


Prerender

<https://github.com/prerender/prerender>

- Node.js
- PhantomJS
- Execution complète
- DOM.toString()





Prerender

- Solution générique
- Résultat statique
- Browser ou bot ?

Google AJAX Crawling

- Proposé en 2009
- Dépréciée en 2015
- Toujours largement utilisée

```
<meta name="fragment" content="!">
```

https://www.icilalune.com/?_escaped_fragment_

Pour les autres bots : User-agent

Progressive enhancement

- Prerender généralisé
- Résultat dynamique
- Dépendant du framework

Server-side rendering

- Angular Universal
- Ember fastboot
- ...

Level #3

Crawlers

The Headless Game

- Level #1 : L'API
- Level #2 : La navigation
- Level #3 : Les crawlers

Et ensuite...

- Authentication : OAuth2
- Caching
- JSONAPI et/ou GraphQL in core
- Sécurisation du back
- ...

Merci !

Des questions ?



<http://www.icilalune.com/>