**Versionnez votre API**

1. **Stratégies de versionnement :**

il ya plusieurs statégies :

|  |  |
| --- | --- |
| **Stratégie** | **Exemple** |
| Dans le nom de domaine | http://**v2**.domain.name/api |
| En préfixe de l'URI | http://domain.name/api/**v2** |
| En query string | http://domain.name/api?v=**v2.0.0** |
| En entête HTTP personnalisée | X-API-Version: **2.0.0** |
| En entête HTTP  Accept | Accept: application/vnd.app.articles+json; version=**2.0.0** |

il est mieux de faire en sorte que votre application n'ait pas d'URL qui change

1. **Formalisation de la version en entête HTTP Accept :**

Il faut maintenant indiquer depuis quelle version les champs sont disponibles :

<?php

namespace AppBundle\Entity;

use JMS\Serializer\Annotation as Serializer;

/\*\*

\* …

\*/

class Article

{

// …

/\*\*

\* @ORM\Column(type="string", length=100)

\* @Serializer\Since("1.0")

\*

\* @Assert\NotBlank()

\*/

private $title;

/\*\*

\* @ORM\Column(type="text")

\* @Serializer\Since("1.0")

\* @Assert\NotBlank()

\*/

private $content;

/\*\*

\* @ORM\Column(type="text", nullable=true)

\* @Serializer\Since("2.0")

\*/

private $shortDescription;

// …

}

Notre exemple va afficher shortDescription pour que la version 2.0 de l’API

La configuration :

Dans un premier temps, il faut activer le versionnement grâce à l'entête de requête HTTP Accept avec la configuration suivante :

# app/config/config.yml

fos\_rest:

#…

versioning:

enabled: true

resolvers:

media\_type: # Accept header

enabled: true

regex: '/(v|version)=(?P<version>[0-9\.]+)/'

Il nous faut ensuite déclarer les versions disponibles grâce au MIME types :

# app/config/config.yml

fos\_rest:

# …

view:

mime\_types:

json: ['application/json', 'application/json;version=1.0', 'application/json;version=2.0']