

当我们使用Post类的第一个接口来获取文章列表时，通常希望能够对列表中的结果进行过滤。比如说查询具有某个关键字的文章，或者查询某个作者的文章等等。Api Platform提供了过滤器的功能，可以对结果进行过滤。

我们查看Api Platform文档，在左边有个Filters文档。我们往下看，在文档中这里有一段示例代码，Api Platform提供了`ApiFilter`这个注解。我们可以在注解中设置过滤器类型，然后设置当前类中的属性所使用的过滤策略。

我们往下看，Api Platform提供了很多过滤器，比如说Search Filter和Date Filter。我们查看一下Search Filter，使用SearchFilter我们可以对某些属性进行过滤，这里有一些过滤的策略。比如说exact就是使用精确过滤，partial就是对字符进行模糊查询，在数据库中它使用LIKE关键字对数据进行模糊查询。然后是start end也是进行模糊查询，但是它们查询的是某个字符串开头或者某个字符串结尾的数据。

然后我们看日期过滤器，使用日期过滤器，我们可以查询某个日期之前或者某个日期之后的资源数据。Api Platform还内置了其他几种过滤器，这里就不再介绍了，它们的使用方法都是添加`ApiFilter`注解，然后在注解中添加过滤器的类型，然后设置属性的过滤策略。

我们查看API文档，在文章的API中，之前我们使用`ApiSubresource`生成了一个API，用来获取某个文章下所有的评论，使用ApiSubresouce会对数据进行多层嵌套，这样可能出现一些问题。

我们往上看，我们已经为Comment类添加了ApiResource注解，这样Comment类也自动的生成了API，我们可以在第一个接口中添加过滤器。使用过滤器来进行条件查询，来获取某个文章下所有评论。

回到项目，打开Comment类，在Comment类前我们添加`ApiFilter`注解，首先我们要指定过滤器的类型。这里我们使用SearchFilter，这里使用Api Platform ORM组件下的SearchFilter。然后是对Comment类的属性进行过滤条件的设置`properties`，这里需要设置为数组。

我们现在查询\$message中具有某个关键字的评论，这里我们输入message，它的查询策略，我们使用模糊查询。查看文档，复制partial，粘贴。回到浏览器刷新文档页面，展开第一个接口，现在我们看到查询参数中。除了page之外，还多了一个message。message和page都是query参数。

查看管理端。登录超级管理员用户，查看评论列表，我们获取message中包含Minus关键字的评论，回到文档点击Try it out，这里我们输入Minus，点击Execute。

现在我们查看响应的结果，所有message中包含minus的数据都进行了显示，往下看一共有4条数据，这就是SearchFilter一个应用。

回到项目，现在希望能够获取某个文章的所有评论数据，这里我们输入`post.id`，我们对文章的id进行精确查询，查询策略我们输入exact。回到浏览器刷新文档，展开Comment类第一个接口，现在查询参数中就多了一个post.id，我们点击Try it out，我们获取id为20的文章所有的评论，点击Execute。

```
#src/Entity/Comment.php

/**
 * @ORM\Entity(repositoryClass=CommentRepository::class)
 * @ORM\HasLifecycleCallbacks
 * @ORM\EntityListeners({"App\Listener\CommentListener"})
 * @Assert\EnableAutoMapping
 */
#[ApiResource]
```

```
#[ApiFilter(SearchFilter::class, properties: [
    'message' => 'partial',
    'post.id' => 'exact'
])]
class Comment
{
    use DateTimeTrait;
}
```

现在我们看到查询到的结果，拉到最下方，一共有28条顶级评论，如果我们想使用其他的Filter。回到项目，我们可以在注解类前继续添加ApiFilter，然后指定其他的Filter类名，并且对属性进行设置，这就是过滤器的简单使用。

回到Post类，如果我们在类前配置了ApiResource注解，那么Api Platform将会自动的为我们添加六个操作，我们可以通过配置`collectionOperations`和`itemOperations`对默认的操作进行删除。

我们也可以添加自定义的操作，在下节课我们将自定义文件上传的操作。