



黑客派

# Swagger 注解 传参

作者: [junjiecheng](#)

原文链接: <https://hacpai.com/article/1534237475654>

来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

<h3 id="--Api-">[@Api]</h3>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p><code>@ Api</code> 用于声明 Swagger 资源 API。它有双重用途 - 它会影响资源列表_和_
PI 声明。只有使用 <code>@ Api</code> 注释的类才会被 Swagger 扫描。</p>
<p>在资源清单中，注释将转换为[资源对象]</p>
<p>在 API 声明中，它基本上将作为[API 声明]本身的基础。</p>
<p>JAX-RS 的用法是：</p>
<div>
  <pre class="has"><code class="language-java hljs">

  <ol>

  </ol></code>

  <div class="hljs-button"></div></pre>
  <p>&nbsp;</p>
</div>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p></p>
<h3 id="--ApiOperation-">[@ApiOperation]</h3>
<p><code>@ ApiOperation</code> 用于在 API 资源中声明单个操作。操作被认为是路径和 HT
P 方法的唯一组合。只扫描使用 <code>@ ApiOperation</code> 注释的方法并添加 API 声明。<
p>
<p>注释将影响 Swagger 输出的两个部分，[API 对象]，每个路径将创建一个，以及[操作对象]，将
据@ApiOperation 创建一个。请记住，在使用 Servlet 时，<code>@ Api</code> 会在设置路径
影响 API 对象。</p>
<table>
  <thead>
    <tr>
      <th align="center">JAX-RS 的用法是</th>
    </tr>
  </thead>
</table>
<div>

```

```
<pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
```

```
<p>&nbsp;</p>
```

```
</div>
```

```
<p></p>
```

```
<h3 id="--ApiResponse-"">[@ApiResponses],&nbsp;[@ApiResponse]</h3>
```

```
<p>使用 HTTP 状态码返回错误 (或其他成功消息) 是一种常见做法。虽然操作的常规返回类型在
@ApiOperation]中定义, 但应使用这些注释描述其余的返回代码。</p>
```

```
<p><code>@ ApiResponse</code> 描述了具体的可能响应。它不能直接在方法上使用, 需要包
在 <code>@ ApiResponses</code> 的数组值中 (无论是否有一个响应或更多)。</p>
```

```
<p>如果响应伴随身体, 也可以描述身体模型 (每个响应一个模型)。</p>
```

```
<p>用法 (JAX-RS, Servlet 或其他) 之间的使用没有区别: </p>
```

```
<div>
```

```
<pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
```

```
<p>&nbsp;</p>
```

```
</div>
```

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
```

```
<script>
```

```
(adsbygoogle = window.adsbygoogle || []).push({});
```

```
</script>
```

```
<p>有关此注释, 用法和边缘情况的更多详细信息, 请查看 javadoc ([@ApiResponses],&nbsp;[@
ApiResponse]**.</p>
```

```
<p></p>
```

```
<h3 id="--ApiParam-"">[@ApiParam]</h3>
```

```
<p><code>@ ApiParam</code> 仅用于 JAX-RS 参数注释 (<code>@PathParam</code>, <
code>@QueryParam</code>, <code>@HeaderParam</code>, <code>@FormParam</co
de> 和 JAX-RS 2, <code>@BeanParam</code>)。虽然 swagger-core 默认扫描这些注释, 但
<code>@ ApiParam</code> 可用于添加有关参数的更多详细信息, 或在从代码中读取时更改值。
/p>
```

<p>在 Swagger 规范中，这转换为[参数对象]。</p>

<p>Swagger 将获取这些注释的 <code>value () </code> 并将它们用作参数名称，并根据注释置参数类型。对于 body 参数 (JAX-RS 方法的单个输入参数)，名称将自动设置为 <code>body</code> (根据 Swagger 规范的要求)。</p>

<p>如果存在，Swagger 还将使用 <code>@ DefaultValue</code> 的值作为默认值属性。</p></div>

```
<pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
```

<p>&nbsp;</p>

```
</div>
```

<p>这里我们有两个参数。第一个是 <code>areaName</code>，它是路径的一部分。第二个是文，在本例中是一个 Area 对象。请注意，两个参数都将 <code>required</code> 属性设置为 <code>true</code>。对于 @PathParam，这是多余的，因为默认情况下它是强制性的，不能被覆盖</p>

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
<script>
```

```
(adsbygoogle = window.adsbygoogle || []).push({});
```

```
</script>
```

<p>输出将是: </p>

<p><strong>有关此注释，用法和边缘情况的更多详细信息，请查看<a href="https://link.acpai.com/forward?goto=http%3A%2F%2Fdocs.swagger.io%2Fswagger-core%2Fv1.3.12%2Fpidocs%2Findex.html%3Fcom%2Fwordnik%2Fswagger%2Fannotations%2FApiParam.html" target="\_blank" rel="nofollow ugc">javadocs</a>.</strong></p>

<p></p>

### [@ApImplicitParam],&nbsp;[@ApImplicitParams]</h3>

<p>您可能希望手动描述操作参数。这可能有多种原因，例如: </p>

```
<ul>
```

```
<li>使用不使用 JAX-RS 注释的 Servlet.</li>
```

```
<li>想要隐藏定义的参数，并使用完全不同的定义覆盖它.</li>
```

```
<li>描述在到达 JAX-RS 实现之前过滤器或其他资源使用的参数.</li>
```

```
</ul>
```

<p>由于可以包含几个参数，<code>@ ApImplicitParams</code> 允许多个 <code>@ ApImplicitParam</code> 定义。</p>

<p>在 Swagger 规范中，这些转换为[Parameter Object]。</p>

<p>在隐式定义参数时，为 Swagger 的定义设置 <code>name</code>，<code>dataType</code> 和 <code>paramType</code> 是很重要的。</p>

```
</div>
```

```
<pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
<p>&nbsp;</p>
</div>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p>在上面的示例中，我们可以看到具有多个参数的 Servlet 定义。 <code>dataType</code> 可
是基元名称或类名称。 <code>paramType</code> 可以是 Swagger 支持的任何参数类型（有关
多详细信息，请参阅 javadoc 或规范）.</p>
<p></p>
<p><strong>有关此注释，用法和边缘情况的更多详细信息，请查看 javadoc (<a href="https://lin
.hacpai.com/forward?goto=http%3A%2F%2Fdocs.swagger.io%2Fswagger-core%2Fv1.3.12%2
apidocs%2Findex.html%3Fcom%2Fwordnik%2Fswagger%2Fannotations%2FApiImplicitParam
.html" target="_blank" rel="nofollow ugc">@ApiImplicitParam</a>,&nbsp;<a href="https://li
k.hacpai.com/forward?goto=http%3A%2F%2Fdocs.swagger.io%2Fswagger-core%2Fv1.3.12%
Fapidocs%2Findex.html%3Fcom%2Fwordnik%2Fswagger%2Fannotations%2FApiImplicitPara
s.html" target="_blank" rel="nofollow ugc">@ApiImplicitParams</a>)</strong>.</p>
<h3 id="--ApiModel-">[@ApiModel]</h3>
<p>Swagger-core 在整个 API 内省中基于对它们的引用构建模型定义。 <code>@ ApiModel</co
de> 允许您操作模型的元数据，从简单的描述或名称更改到多态的定义。</p>
<p>这转换为 Swagger 规范中的[Model Object]。</p>
<p>在其基本功能中，您可以使用 <code>@ ApiModel</code> 来更改模型的名称并为其添加描
：</p>
<div>
  <pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
<p>&nbsp;</p>
</div>
<p>我们将模型的名称从 <code>Area</code> 更改为 <code>区域domain</code></p>
<p>为了支持多态和继承，我们使用 <code>discriminator</code> 和 <code>subTypes</code>
字段。两者都必须用于 Swagger 输出才有效。</p>
<p>“discriminator” 字段必须是顶部模型中的字段，该字段将用于确定正在使用哪个子模型。例
，如果您有一个 <code>Animal</code> 类，其中 <code>Cat</code>， <code>Dog</code>
```

和 `Chicken` 作为子类，则 `animalType` 字段可以用作鉴别器来定实际使用的是哪种动物。

`subTypes` 必须列出继承模型的类。类本身不必从超类型继承。事实上，Swagger 不会自动读取扩展类，你必须在 `subTypes` 中手动描述这些类，以便对它们进行解析。

<div>

```
<pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
```

```
<p>&nbsp;</p>
```

```
</div>
```

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
<script>
```

```
  (adsbygoogle = window.adsbygoogle || []).push({};
```

```
</script>
```

```
<p>上面的代码片段是一个如何描述继承的简单示例。注意 SubModel 不会扩展 SuperModel。
```

```
同样的方式，您可以添加多个继承类。可以有任意数量的继承级别</p>
```

```
<p></p>
```

```
<p><strong>For further details about this annotation, usage and edge cases, check out the <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.swagger.io%2Fswagger-core%2Fv1.3.12%2Fapidocs%2Findex.html%3Fcom%2Fwordnik%2Fswagger%2Fannotations%2FApiModel.html" target="_blank" rel="nofollow ugc">javadocs</a>.</strong></p>
```

```
<h3 id="--ApiModelProperty-">[@ApiModelProperty]</h3>
```

```
<p>虽然 swagger-core 将内省字段和 setter / getter，但它也会读取和处理 JAXB 注释。<code>ApiModelProperty</code> 允许控制 Swagger 特定的定义，例如允许的值和附加注释。如果您在某些情况下隐藏属性，它还提供其他过滤属性。</p>
```

```
<p>有关 Swagger Spec 中的相关信息，请查看 <a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fgithub.com%2FOAI%2FOpenAPI-Specification%2Fblob%2Fmaster%2Fversions%2F1.2.md%23529-property-object" target="_blank" rel="nofollow ugc">Property Object</a>.</p>
```

```
<div>
```

```
<pre class="has"><code class="language-java hljs">
```

```
<ol>
```

```
</ol></code>
```

```
<div class="hljs-button"></div></pre>
```

```
<p>&nbsp;</p>
```

```

</div>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p>这是向模型属性添加简短描述的简单示例。还可以观察到，虽然 status 是一个
String，但我们将其记录为只有三个可能的值。</p>
<p></p>
<p><strong>有关此注释，用法和边缘情况的更多详细信息，请查看 <a href="https://link.hacpai.
om/forward?goto=http%3A%2F%2Fdocs.swagger.io%2Fswagger-core%2Fv1.3.12%2Fapidoc
%2Findex.html%3Fcom%2Fwordnik%2Fswagger%2Fannotations%2FApiModelProperty.html"
arget="_blank" rel="nofollow ugc">javadocs</a>.</strong></p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#b3_solo_h3_0">@Api</a></td>
<td>将类标记为Swagger资源.</td>
</tr>
<tr>
<td><a href="#b3_solo_h3_4">@ApiImplicitParam</a></td>
<td>表示API操作中的单个参数.</td>
</tr>
<tr>
<td><a href="#b3_solo_h3_4">@ApiImplicitParams</a></td>
<td>一个包装器，允许列出多个ApiImplicitParam对象.</td>
</tr>
<tr>
<td><a href="#b3_solo_h3_5">@ApiModel</a></td>
<td>提供有关Swagger模型的其他信息.</td>
</tr>
<tr>
<td><a href="#b3_solo_h3_6">@ApiModelProperty</a></td>
<td>添加和操作模型属性的数据.</td>
</tr>
<tr>
<td><a href="#b3_solo_h3_1">@ApiOperation</a></td>
<td>描述针对特定路径的操作或通常是HTTP方法.</td>
</tr>
<tr>
<td><a href="#b3_solo_h3_3">@ApiParam</a></td>
<td>欵为操作参数添加其他元数据.</td>

```

```
</tr>
<tr>
  <td><a href="#b3_solo_h3_2">@ApiResponse</a></td>
  <td>描述操作的可能响应.</td>
</tr>
<tr>
  <td><a href="#b3_solo_h3_2">@ApiResponses</a></td>
  <td>鈇一个包装器, 允许列出多个ApiResponse对象.</td>
</tr>
<tr>
  <td>@Authorization</td>
  <td>鈇声明要在资源或操作上使用的授权方案.</td>
</tr>
<tr>
  <td>@AuthorizationScope</td>
  <td>鈇描述OAuth2授权范围.</td>
</tr>
</tbody>
</table>
```