

50. 请求变量

学习要点：

1. 请求变量
2. 助手函数

本节课我们来学习一下数据请求变量的功能以及助手函数的简化。

一. 请求变量

1. **Request** 对象支持全局变量的检测、获取和安全过滤，支持 `$_GET`、`$_POST`... 等；
2. 为了方便演示，这里一律使用 **Facade** 的静态调用模式；
3. 使用 `has()` 方法，可以检测全局变量是否已经设置：

```
Request::has('id', 'get');  
Request::has('username', 'post');
```

4. **Request** 支持的所有变量类型方法，如下表：

方法	描述
param	获取当前请求的变量
get	获取 <code>\$_GET</code> 变量
post	获取 <code>\$_POST</code> 变量
put	获取 <code>PUT</code> 变量
delete	获取 <code>DELETE</code> 变量
session	获取 <code>SESSION</code> 变量
cookie	获取 <code>\$_COOKIE</code> 变量
request	获取 <code>\$_REQUEST</code> 变量
server	获取 <code>\$_SERVER</code> 变量
env	获取 <code>\$_ENV</code> 变量
route	获取 路由（包括 <code>PATHINFO</code> ）变量
middleware	获取 中间件赋值/传递的变量
file	获取 <code>\$_FILES</code> 变量

5. `param()` 变量方法是自动识别 `GET`、`POST` 等的当前请求，推荐使用：

```
//获取请求为 name 的值，过滤  
Request::param('name');  
//获取所有请求的变量，以数组形式，过滤  
Request::param();  
//获取所有请求的变量(原始变量)，不包含上传变量，不过滤  
Request::param(false);  
//获取部分变量  
Request::param(['name', 'age']);
```

6. 默认情况下，并没有配置字符过滤器，可在 `app\Request.php` 配置；
`protected $filter = ['htmlspecialchars'];`
7. 如果没有设置字符过滤器，或者局部用别的字符过滤器，可以通过第三参数；
`Request::param('name', '', 'htmlspecialchars');`
`Request::param('name', '', 'strip_tags, strtolower');`
8. 如果设置了全局字符过滤器，但又不想某个局部使用，可以只用 `null` 参数；
`Request::param('name', '', null)`
9. 如果获取不到值，支持请求的变量设置一个默认值；
`Request::param('name', '默认值');`
10. 如果采用的是路由 URL，也可以获取到变量，但 `param::get()` 不支持路由变量；
`Request::param('id');`
`Request::route('id');`
`Request::get('id');` //路由参数，get 获取不到
11. 使用 `only()` 方法，可以获取指定的变量，也可以设置默认值；
`Request::only(['id', 'name']);`
`Request::only(['id'=>1, 'name'=>'默认值']);`
12. 使用 `only()` 方法，默认是 `param` 变量，可以在第二参数设置 GET、POST 等；
`Request::only(['id', 'name'], 'post');`
13. 相反的 `except()` 方法，就是排除指定的变量；
`Request::except('id, name');`
`Request::except(['id', 'name']);`
`Request::except(['id'=>1, 'name'=>'默认值']);`
`Request::except(['id', 'name'], 'post');`
14. 使用变量修饰符，可以将参数强制转换成指定的类型；
15. `/s`(字符串)、`/d`(整型)、`/b`(布尔)、`/a`(数组)、`/f`(浮点)；
`Request::param('id/d');`

二. 助手函数

1. 为了简化操作，`Request` 对象提供了助手函数；
`input('?get.id');` //判断 get 下的 id 是否存在
`input('?post.name');` //判断 post 下的 name 是否存在
`input('param.name');` //获取 param 下的 name 值
`input('param.name', '默认值');` //默认值
`input('param.name', '', 'htmlspecialchars');` //过滤器
`input('param.id/d');` //设置强制转换