46．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：

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

4. Request 支持的所有变量类型方法，如下表：

param	获取当前请求的变量
get	获取 $_GET 变量
post	获取 $_POST 变量
put	获取 PUT 变量
delete	获取 DELETE 变量
session	获取 $_SESSION 变量
cookie	获取 $_COOKIE 变量
request	获取 $_REQUEST 变量
server	获取 $_SERVER 变量
env	获取 $_ENV 变量
route	获取 路由（包括PATHINFO）变量
file	获取 $_FILES 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
dump(Request::param('name'));
//获取所有请求的变量，以数组形式，过滤
dump(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
dump(Request::param(false));
//获取所有请求的变量，包含上传变量，过滤
dump(Request::param(true));
```

6. 如果使用依赖注入的方式，可以将变量作为对象的属性进行调用；

```
public function read(\think\Request $request)
{
    return $request->name;
}
```

7. 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；

```
public function edit($id)
{
    dump(Request::param());
    dump(Request::route());              // 路由请求不支持 get 变量
    dump(Request::get());                // get 变量不支持路由请求
}
```

8. 注意：除了::server()和::env()方法外，其它方法传递的变量名区分大小写；
9. 因为::server()和::env()属于系统变量，会强制转换为大写后获取值；
10. 如果获取不到值，支持请求的变量设置一个默认值；

```
dump(Request::param('name', 'nodata'));
```

11. 对于变量的过滤，在全局设置一个过滤函数，也可以单独对某个变量过滤；

```
'default_filter'        => 'htmlspecialchars',
Request::param('name', '', 'htmlspecialchars');
Request::param('name', '', 'strtoupper');
```

12. 使用 only()方法，可以获取指定的变量，也可以设置默认值；

```
dump(Request::only('id,name'));
dump(Request::only(['id','name']));
dump(Request::only(['id'=>1,'name'=>'nodata']));
```

13. 使用 only()方法，默认是 param 变量，可以在第二参数设置 GET、POST 等；

```
dump(Request::only(['id','name'], 'post'));
```

14. 相反的 except()方法，就是排除指定的变量；

```
dump(Request::except('id,name'));
dump(Request::except(['id','name']));
dump(Request::except(['id'=>1,'name'=>'nodata']));
dump(Request::except(['id','name'], 'post'));
```

15. 使用变量修饰符，可以将参数强制转换成指定的类型；
16. /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；

```
dump(input('?get.id'));          //判断 get 下的 id 是否存在
dump(input('?post.name'));       //判断 post 下的 name 是否存在
dump(input('param.name'));       //获取 param 下的 name 值
dump(input('param.name', 'nodata'));                    //默认值
dump(input('param.name', '', 'htmlspecialchars'));      //过滤器
dump(input('param.id/d'));       //设置强制转换
```