

56. 日志处理

学习要点：

1. 日志处理

本节课我们来学习一下日志处理的功能，包括配置日志和写入日志等。

一. 日志处理

1. 日志处理的操作由 `Log` 类完成，它记录着所有程序中运行的错误记录；
2. 在 `config` 目录下的 `log.php` 配置文件，用于设置日志信息；
3. 我们在 `runtime` 目录下后一个 `log` 文件夹，里面按照日期排好了每月的日志；
4. 使用 `record()` 方法，记录一条测试日志；

```
Log::record('测试日志！');
```
5. 我们在 `log` 日志文件夹里找到最新生成的日志，可以看到生成的日志信息；
6. 系统提供了不同日志级别，默认 `info` 级别，从低到高排列如下：
7. `debug/info/notice/warning/error/critical/alert/emergency`；
8. 一般记录就是 `info` 信息，我们也可以指定我们的信息级别；

```
Log::record('测试日志！', 'error');
```
9. 系统还提供了关闭写入的功能，在配置文件中关闭，或者使用 `::close()` 方法；

```
Log::close();
```

```
'close' => false,
```
10. 系统发生异常后，会自动写入 `error` 日志，如果你想手动也可以；

```
try {  
    echo 0/0;  
} catch (ErrorException $e)  
{  
    echo '发生错误：'. $e->getMessage();  
    Log::record('被除数不得为零', 'error');  
}
```
11. 对于各种日志级别，系统提供了一些快捷方式和助手函数，比如：

```
Log::error('错误日志！');           //Log::record('错误日志！', 'error')  
Log::info('信息日志！');           //Log::record('信息日志！', 'info')  
trace('错误日志！', 'error');  
trace('信息日志！', 'info');
```
12. 如果你开启的是调试模式，那每次加载都会写入内存的运行数据，关闭就不写入；
13. 系统默认并不记录 HTTP 异常，因为这种异常容易遭受攻击不断写入日志；
14. 在配置文件 `log.php` 中，可以设置路径，来更改日志文件的存储位置；

```
Env::get('app_path') . '../runtime/logs/';
```

15. 在配置文件 `log.php` 中，可以设置限定日志文件的级别，不属于的无法写入；

```
'level' => ['info'],
```

16. 在配置文件 `log.php` 中，添加转换为 `json` 格式，部署模式不知；

```
'json' => true
```

17. 使用 `::getLog()` 方法，可以获取写入到内存中的日志；

```
$logs = Log::getLog();  
dump($logs);
```

18. 使用 `::clear()` 方法，可以清理掉内存中的日志；

```
Log::clear();
```

19. 在配置文件 `log.php` 中，可以设置以单独文件存储的方式；

```
'single' => true,
```

20. 在配置文件 `log.php` 中，设置 `max_files` 最大文件数量，会按日期文件存储；

21. 一旦超过指定的数量，早期的会被清理掉，验证时，更改服务器时间，设置如下：

```
'max_files' => 2,
```