# 简介

Sessions 和 cookies 允许数据在多次请求中保持， 在纯 PHP 中，可以分别使用全局变量 $\_SESSION 和 $\_COOKIE 来访问，Yii 将 session 和 cookie 封装成对象并增加一些功能， 可通过面向对象方式访问它们。

# Sessions

和 [请求](https://www.yiichina.com/doc/guide/2.0/runtime-requests) 和 [响应](https://www.yiichina.com/doc/guide/2.0/runtime-responses)类似， 默认可通过为 [yii\web\Session](https://www.yiichina.com/doc/api/2.0/yii-web-session) 实例的 session [应用组件](https://www.yiichina.com/doc/guide/2.0/structure-application-components) 来访问 sessions。

## 开启和关闭 Sessions

可使用以下代码来开启和关闭 session。



## 访问 Session 数据

可使用如下方式访问session中的数据：



当 session 数据为数组时，session 组件会限制你直接修改数据中的单元项， 例如：



使用变通方法来解决这个问题：



## 自定义 Session 存储

[yii\web\Session](https://www.yiichina.com/doc/api/2.0/yii-web-session) 类默认存储 session 数据为文件到服务器上， Yii 提供以下 session 类实现不同的 session 存储方式：

* [yii\web\DbSession](https://www.yiichina.com/doc/api/2.0/yii-web-dbsession)：存储 session 数据在数据表中
* [yii\web\CacheSession](https://www.yiichina.com/doc/api/2.0/yii-web-cachesession)：存储 session 数据到缓存中，缓存和配置中的[缓存组件](https://www.yiichina.com/doc/guide/2.0/caching-data" \l "cache-components)相关
* yii\redis\Session：存储 session 数据到以 [redis](http://redis.io/) 作为存储媒介中
* yii\mongodb\Session：存储 session 数据到 [MongoDB](http://www.mongodb.org/)。

所有这些 session 类支持相同的 API 方法集，因此， 切换到不同的 session 存储介质不需要修改项目使用 session 的代码。

****注意：****如果通过$\_SESSION访问使用自定义存储介质的session， 需要确保session已经用[yii\web\Session::open()](https://www.yiichina.com/doc/api/2.0/yii-web-session" \l "open()-detail) 开启， 这是因为在该方法中注册自定义session存储处理器。

## Flash 数据

Flash 数据是一种特别的 session 数据，它一旦在某个请求中设置后， 只会在下次请求中有效，然后该数据就会自动被删除。

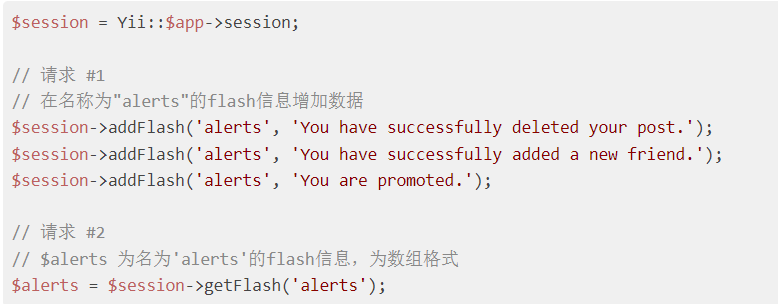
常用于实现只需显示给终端用户一次的信息， 如用户提交一个表单后显示确认信息。

可通过 session 应用组件设置或访问 session，例如：

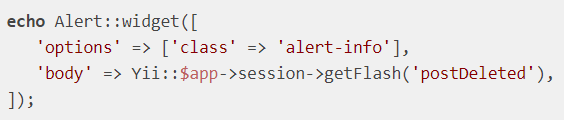


和普通 session 数据类似，可将任意数据存储为 flash 数据。

当调用 [yii\web\Session::setFlash()](https://www.yiichina.com/doc/api/2.0/yii-web-session" \l "setFlash()-detail) 时, 会自动覆盖相同名的已存在的任何数据， 为将数据追加到已存在的相同名 flash 中，可改为调用 [yii\web\Session::addFlash()](https://www.yiichina.com/doc/api/2.0/yii-web-session" \l "addFlash()-detail)。 例如:



****提示：****要显示 Flash 消息，您可以通过以下方式使用 yii\bootstrap\Alert 小部件：



# Cookies

Yii 使用 [yii\web\Cookie](https://www.yiichina.com/doc/api/2.0/yii-web-cookie) 对象来代表每个 cookie， [yii\web\Request](https://www.yiichina.com/doc/api/2.0/yii-web-request) 和 [yii\web\Response](https://www.yiichina.com/doc/api/2.0/yii-web-response) 通过名为 'cookies' 的属性维护一个 cookie 集合，前者的 cookie 集合代表请求提交的 cookies， 后者的 cookie 集合表示发送给用户的 cookies。

控制器是直接处理请求和响应的部分。 因此, 应当在控制器中读取和发送 cookie 。 (译者注：意思在控制器中处理 cookie 是安全的。)

## 读取 Cookies

当前请求的 cookie 信息可通过如下代码获取：



## 发送 Cookies

可使用如下代码发送 cookie 到终端用户：



## Cookie 验证

在上两节中，当通过 request 和 response 组件读取和发送 cookie 时， 你会喜欢扩展的 cookie 验证的保障安全功能，它能 使 cookie 不被客户端修改。该功能通过给每个 cookie 签发一个哈希字符串来告知服务端 cookie 是否在客户端被修改， 如果被修改，通过 request 组件的 [cookie collection](https://www.yiichina.com/doc/api/2.0/yii-web-request" \l "$cookies-detail) cookie 集合访问不到该 cookie。

Cookie 验证默认启用，可以设置 [yii\web\Request::$enableCookieValidation](https://www.yiichina.com/doc/api/2.0/yii-web-request" \l "$enableCookieValidation-detail) 属性为 false 来禁用它， 尽管如此，我们强烈建议启用它。

****注意：****直接通过 $\_COOKIE 和 setcookie() 读取和发送的 Cookie 不会被验证。

当使用 cookie 验证时，必须指定 [yii\web\Request::$cookieValidationKey](https://www.yiichina.com/doc/api/2.0/yii-web-request" \l "$cookieValidationKey-detail)，它是用来生成上述的哈希值， 可通过在应用配置中配置 request 组件。

