# 响应

当一个应用在处理完一个[请求](https://www.yiichina.com/doc/guide/2.0/runtime-requests)后, 这个应用会生成一个 [response](https://www.yiichina.com/doc/api/2.0/yii-web-response) 响应对象并把这个响应对象发送给终端用户 这个响应对象包含的信息有 HTTP 状态码，HTTP 头和主体内容等, 从本质上说，网页应用开发最终的目标就是根据不同的请求去构建这些响应对象。

# 状态码

构建响应要做的第一件事就是声明请求是否被成功处理，我们可通过设置 [yii\web\Response::$statusCode](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$statusCode-detail) 这个属性来做到这一点，该属性接受一个有效的 [HTTP 状态码](https://tools.ietf.org/html/rfc2616" \l "section-10)。例如，表明该请求已被成功处理， 可以设置状态码为 200，如下所示：

Yii::$app->response->statusCode = 200;

如果需要指定请求失败，可抛出对应的 HTTP 异常，如下所示：

**throw** **new** \yii\web\NotFoundHttpException;

当[错误处理器](https://www.yiichina.com/doc/guide/2.0/runtime-handling-errors) 捕获到一个异常，会从异常中提取状态码并赋值到响应， 对于上述的 [yii\web\NotFoundHttpException](https://www.yiichina.com/doc/api/2.0/yii-web-notfoundhttpexception) 对应 HTTP 404 状态码， 以下为 Yii 预定义的 HTTP 异常：



如果想抛出的异常不在如上列表中，可创建一个 [yii\web\HttpException](https://www.yiichina.com/doc/api/2.0/yii-web-httpexception) 异常， 带上状态码抛出，如下：

**throw** **new** \yii\web\HttpException(402);

# HTTP 头部

可在 response 组件中操控 [header collection](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$headers-detail) 来发送 HTTP 头部信息， 例如：



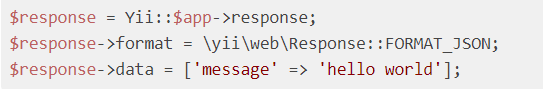
# 响应主体

大多是响应应有一个主体存放你想要显示给终端用户的内容。

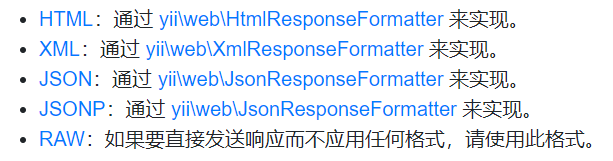
如果已有格式化好的主体字符串，可赋值到响应的 [yii\web\Response::$content](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$content-detail) 属性， 例如：

Yii::$app->response->content = 'hello world!';

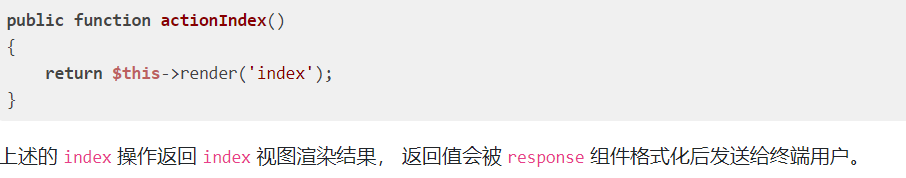
如果在发送给终端用户之前需要格式化，应设置 [format](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$format-detail) 和 [data](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$data-detail) 属性，[format](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$format-detail) 属性指定 [data](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$data-detail)中数据格式化后的样式，例如：



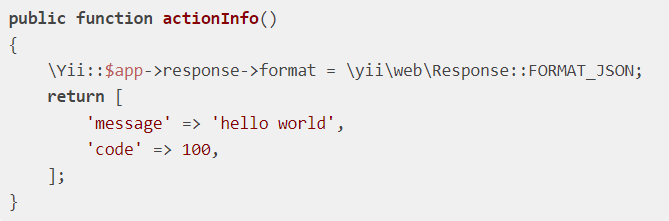
Yii支持以下可直接使用的格式，每个实现了[formatter](https://www.yiichina.com/doc/api/2.0/yii-web-responseformatterinterface) 类， 可自定义这些格式器或通过配置 [yii\web\Response::$formatters](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$formatters-detail) 属性来增加格式器。



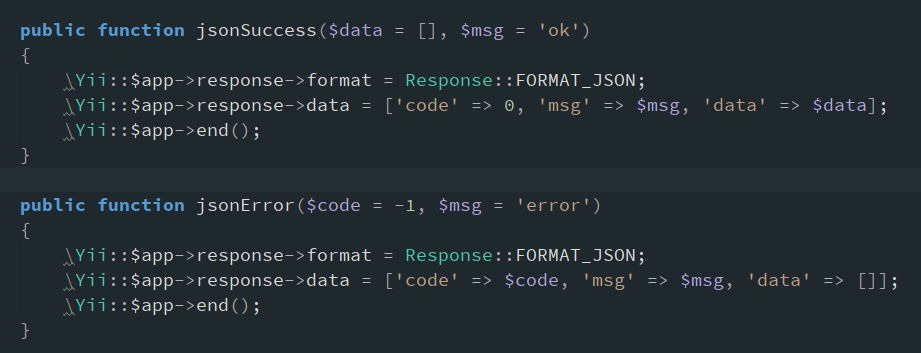
上述响应主体可明确地被设置，但是在大多数情况下是通过[操作](https://www.yiichina.com/doc/guide/2.0/structure-controllers) 方法的返回值隐式地设置，常用场景如下所示：



因为响应格式默认为 [HTML](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "FORMAT_HTML-detail)，只需要在操作方法中返回一个字符串， 如果想使用其他响应格式，应在返回数据前先设置格式， 例如：



比如api需要返回json格式，则可以在BaseController中封装如下方法：



# 浏览器跳转

浏览器跳转依赖于发送一个 Location HTTP 头，因为该功能通常被使用， Yii提供对它提供了特别的支持。

可调用[yii\web\Response::redirect()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "redirect()-detail) 方法将用户浏览器跳转到一个 URL 地址，该方法设置合适的 带指定 URL 的 Location 头并返回它自己为响应对象，在操作的方法中， 可调用缩写版 [yii\web\Controller::redirect()](https://www.yiichina.com/doc/api/2.0/yii-web-controller" \l "redirect()-detail)，例如：



除了动作方法外，可直接调用[yii\web\Response::redirect()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "redirect()-detail) 再调用 [yii\web\Response::send()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "send()-detail) 方法来确保没有其他内容追加到响应中。

\Yii::$app->response->redirect('http://example.com/new', 301)->send();

如果当前请求为 AJAX 请求，发送一个 Location 头不会自动使浏览器跳转，为解决这个问题， [yii\web\Response::redirect()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "redirect()-detail) 方法设置一个值为要跳转的URL的 X-Redirect 头， 在客户端可编写 JavaScript 代码读取该头部值然后让浏览器跳转对应的 URL。

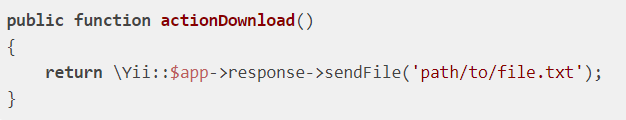
****信息：****Yii 配备了一个 yii.js JavaScript 文件提供常用 JavaScript 功能， 包括基于 X-Redirect 头的浏览器跳转， 因此，如果你使用该 JavaScript 文件（通过 [yii\web\YiiAsset](https://www.yiichina.com/doc/api/2.0/yii-web-yiiasset) 资源包注册）， 就不需要编写 AJAX 跳转的代码。

# 发送（下载）文件

和浏览器跳转类似，文件发送是另一个依赖指定 HTTP 头的功能， Yii 提供方法集合来支持各种文件发送需求，它们对 HTTP 头都有内置的支持。

* [yii\web\Response::sendFile()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "sendFile()-detail)：发送一个已存在的文件到客户端
* [yii\web\Response::sendContentAsFile()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "sendContentAsFile()-detail)：发送一个文本字符串作为文件到客户端
* [yii\web\Response::sendStreamAsFile()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "sendStreamAsFile()-detail)：发送一个已存在的文件流作为文件到客户端

这些方法都将响应对象作为返回值，如果要发送的文件非常大，应考虑使用 [yii\web\Response::sendStreamAsFile()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "sendStreamAsFile()-detail) 因为它更节约内存， 以下示例显示在控制器操作中如何发送文件：



如果不是在操作方法中调用文件发送方法，在后面还应调用 [yii\web\Response::send()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "send()-detail) 没有其他内容追加到响应中。

\Yii::$app->response->sendFile('path/to/file.txt')->send();

# 发送响应

在 [yii\web\Response::send()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "send()-detail) 方法调用前响应中的内容不会发送给用户， 该方法默认在 [yii\base\Application::run()](https://www.yiichina.com/doc/api/2.0/yii-base-application" \l "run()-detail) 结尾自动调用，尽管如此，可以明确调用该方法强制立即发送响应。

[yii\web\Response::send()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "send()-detail) 方法使用以下步骤来发送响应：

1. 触发 [yii\web\Response::EVENT\_BEFORE\_SEND](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "EVENT_BEFORE_SEND-detail) 事件。
2. 调用 [yii\web\Response::prepare()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "prepare()-detail) 来格式化 [response data](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$data-detail) 为 [response content](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "$content-detail)。
3. 触发 [yii\web\Response::EVENT\_AFTER\_PREPARE](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "EVENT_AFTER_PREPARE-detail) 事件。
4. 调用 [yii\web\Response::sendHeaders()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "sendHeaders()-detail) 来发送注册的HTTP头
5. 调用 [yii\web\Response::sendContent()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "sendContent()-detail) 来发送响应主体内容
6. 触发 [yii\web\Response::EVENT\_AFTER\_SEND](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "EVENT_AFTER_SEND-detail) 事件。

一旦 [yii\web\Response::send()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "send()-detail) 方法被执行后，其他地方调用该方法会被忽略， 这意味着一旦响应发出后，就不能再追加其他内容。

如你所见 [yii\web\Response::send()](https://www.yiichina.com/doc/api/2.0/yii-web-response" \l "send()-detail) 触发了几个实用的事件， 通过响应这些事件可调整或包装响应。