Responses

Overview

The CMS application contains a HubzeroHttpResponse instance that all extension output (component, template, etc) is attached to. The response instance allows for customizing the response's HTTP status code, content, and headers. The response instance inherits from the SymfonyComponentHttpFoundationResponse class, providing a variety of methods for building HTTP responses.

Note: For a full list of available Response methods, check out the [Symfony API documentation](#).

Response Object

The creation, setting of content and headers, and sending of the response is handled automatically by the application. But, in some cases, it is beneficial to access and manipulate the response as needed. The response instance may be accessed via its Facade, available in all application types (site, admin, muse, etc), or by retrieving the object directly from the application container.

Facade:

```
Response::header('Content-Type', 'application/json');

echo json_encode($data);
```

Direct access:

```
$response = App::get('response');
$response->header('Content-Type', 'application/json');

echo json_encode($data);
```

Attaching Headers

Most response methods are chainable, allowing for the fluent building of responses. For example, you may use the header method to add a series of headers to the response before sending it back to the user:

```
$response->header('Content-Type', $type)
            ->header('X-Header-One', 'Header Value')
            ->header('X-Header-Two', 'Header Value');
```

Setting Content

To set the content of the response, use the setContent method.Note that the value passed must be of type string.

```
$response->setContent($output);
```

Sending a Response

This will send the set content and headers to the client.

```
$response->send();
```

Redirects

One may also generate redirects by calling the redirect() method on the App. That method accepts three arguments: 1) a URL to predict to, 2) an optional message to display, and 3) an optional message type.

```
App::redirect(
    Route::url('index.php?option=com_support')
);
```

Note that a redirect call is immediate meaning no code immediately after the redirect will be executed.

```
App::redirect(
    Route::url('index.php?option=com_support')
);

// This will not be executed
```

```
die('Hello');
```

The redirect() method is instantiating a new instance of a a HubzeroHttpRedirectResponse class which is a specialized, extended instance of the HubzeroHttpResponse class. If need be, the class can be directly instantiated:

```
$redirect = new HubzeroHttpRedirectResponse($url);
$redirect->setRequest(App::get('request'));
$redirect->send();
```