Request handling and rendering flow

HTTP request

1. Create Request from globals

2. Call HttpKernel::handle()

3. The kernel::request event

HttpKernel::handle()

4. Call

5. Call

HttpKernel::terminate()

5. Call

Response::prepare()

Response::send()

1. The kernel::request event

HttpKernel::handle()

2. The kernel::controller event

KernelEvents::REQUEST

3. The kernel::response event

KernelEvents::RESPONSE

4. The kernel::view event

KernelEvents::VIEW

5. Calling the Controller

Is Response?

No

Yes

KernelEvents::VIEW

MainContentViewSubscriber looks at the available main content renderers. These are tagged with render.main_content_renderer. Contributed modules can add additional renderers.

The flow for all four formats supported in core in their respect is the same. The explanations below only apply to the MainContentViewSubscriber.

If an unsupported format was negotiated, a 406 response is generated.

Example: system/admin, system/admin.html.twig

If anything else (e.g. an object), then it's up to another VIEW event subscriber to turn it into a Response.

MainContentViewSubscriber only handles render arrays!

Example: response for the path /admin

MainContentViewSubscriber looks at the available main content renderers. These are tagged with render.main_content_renderer. Contributed modules can add additional renderers.

The flow for all four formats supported in core in their respect is the same. The explanations below only apply to the MainContentViewSubscriber.

If an unsupported format was negotiated, a 406 response is generated.

Example: BlockPageVariant selected.

The main content must already be rendered, because it might set the page title.

The result of a built page display variant is a Drupal 'page' render array.

Only accepts render arrays of type 'page'!

hook_page_attachments() operates on Drupal 'page'.

renders the entire HTML (i.e. the display variant) as a Drupal 'page' render array.

hook_page_top() and hook_page_bottom() operate on Drupal HTML.

Renders the 3 parts of the HTML: body/page_top, body/page_bottom (the page corresponds to body/page) and hook_page_html/exception (the exception) Then renders the entire HTML (i.e. the $htmlTwigTemplate)

Example: system/admin

Typically, the format will be html, but it could also be json, dialog, modal, json-hal, json, and more.

See Symfony's 'Request::setFormat()'.

Example: /admin path → system/admin/adminBlockPage

That route has the following attributes:

controller: "Drupal\SystemController\systemAdminMenuBlockPage"

_file: 'Administration'

_controller is set to "Drupal\SystemController\systemAdminMenuBlockPage"

MainContentViewSubscriber only handles render arrays!

If it’s anything else (e.g. an object), then it’s up to another VIEW event subscriber to turn it into a Response.

MainContentViewSubscriber looks at the available main content renderers. These are tagged with render.main_content_renderer. Contributed modules can add additional renderers.

The flow for all four formats supported in core in their respect is the same. The explanations below only apply to the MainContentViewSubscriber.

If an unsupported format was negotiated, a 406 response is generated.

Example: BlockPageVariant selected.

The main content must already be rendered, because it might set the page title.

The result of a built page display variant is a Drupal 'page' render array.

Only accepts render arrays of type 'page'!

hook_page_attachments() operates on Drupal 'page'.

renders the entire HTML (i.e. the display variant) as a Drupal 'page' render array.

hook_page_top() and hook_page_bottom() operate on Drupal HTML.

Renders the 3 parts of the HTML: body/page_top, body/page_bottom (the page corresponds to body/page) and hook_page_html/exception (the exception) Then renders the entire HTML (i.e. the $htmlTwigTemplate)

Example: system/admin

Typically, the format will be html, but it could also be json, dialog, modal, json-hal, json, and more.

See Symfony's 'Request::setFormat()'.

Example: /admin path → system/admin/adminBlockPage

That route has the following attributes:

controller: "Drupal\SystemController\systemAdminMenuBlockPage"

_file: 'Administration'

_controller is set to "Drupal\SystemController\systemAdminMenuBlockPage"

MainContentViewSubscriber only handles render arrays!

If it’s anything else (e.g. an object), then it’s up to another VIEW event subscriber to turn it into a Response.

MainContentViewSubscriber looks at the available main content renderers. These are tagged with render.main_content_renderer. Contributed modules can add additional renderers.

The flow for all four formats supported in core in their respect is the same. The explanations below only apply to the MainContentViewSubscriber.

If an unsupported format was negotiated, a 406 response is generated.

Example: BlockPageVariant selected.

The main content must already be rendered, because it might set the page title.

The result of a built page display variant is a Drupal 'page' render array.

Only accepts render arrays of type 'page'!

hook_page_attachments() operates on Drupal 'page'.

renders the entire HTML (i.e. the display variant) as a Drupal 'page' render array.

hook_page_top() and hook_page_bottom() operate on Drupal HTML.

Renders the 3 parts of the HTML: body/page_top, body/page_bottom (the page corresponds to body/page) and hook_page_html/exception (the exception) Then renders the entire HTML (i.e. the $htmlTwigTemplate)

Example: system/admin

Typically, the format will be html, but it could also be json, dialog, modal, json-hal, json, and more.

See Symfony's 'Request::setFormat()'.

Example: /admin path → system/admin/adminBlockPage

That route has the following attributes:

controller: "Drupal\SystemController\systemAdminMenuBlockPage"

_file: 'Administration'

_controller is set to "Drupal\SystemController\systemAdminMenuBlockPage"

MainContentViewSubscriber only handles render arrays!

If it’s anything else (e.g. an object), then it’s up to another VIEW event subscriber to turn it into a Response.

MainContentViewSubscriber looks at the available main content renderers. These are tagged with render.main_content_renderer. Contributed modules can add additional renderers.

The flow for all four formats supported in core in their respect is the same. The explanations below only apply to the MainContentViewSubscriber.

If an unsupported format was negotiated, a 406 response is generated.

Example: BlockPageVariant selected.

The main content must already be rendered, because it might set the page title.

The result of a built page display variant is a Drupal 'page' render array.

Only accepts render arrays of type 'page'!

hook_page_attachments() operates on Drupal 'page'.

renders the entire HTML (i.e. the display variant) as a Drupal 'page' render array.

hook_page_top() and hook_page_bottom() operate on Drupal HTML.

Renders the 3 parts of the HTML: body/page_top, body/page_bottom (the page corresponds to body/page) and hook_page_html/exception (the exception) Then renders the entire HTML (i.e. the $htmlTwigTemplate)