Request handling and rendering flow

HTTP request

index.php

1. Create Request from globals
2. Call HttpKernel::handle()
3. Call Response::prepare()
4. Call Response::send()
5. Call HttpKernel::terminate()

HTTP response

Symfony (HttpKernel)

HttpKernel::handle()

1. The kernel.request event
2. Resolve the Controller
3. The kernel.controller event
4. Getting the Controller Arguments
5. Calling the Controller

Is Response?
Yes / No

6. The kernel.view Event

Response

7. The kernel.response Event

Event handling

KernelEvents::REQUEST

Negotiate format

Determine route & controller

KernelEvents::CONTROLLER

Colored overlay arrows: showing the flow of the different types of Controller return values.

Controller returns either:
- Response
- render array
- object with associated kernel.view event subscriber

KernelEvents::VIEW

SELECT_PAGE_DISPLAY_VARIANT

Select page display variant

KernelEvents::RESPONSE

KernelEvents::TERMINATE

Drupal (Controller, MainContentViewSubscriber, main content renderers)

MainContentViewSubscriber

Is render array?
No / Yes

Main content renderer for request format exists?
No / Yes

Initialize the corresponding main content renderer

Call ::renderResponse()

Generate 406 Response

Set Response on Event

Typical flow

main content renderers

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

| format = 'html' | format = 'dialog' | format = 'modal' | format = 'ajax' |

HtmlRenderer

1. prepare() helper

Is a #type 'page' render array?
No / Yes

drupal_render() for main content

Build page display variant

hook_page_attachments()

Calculate title if not provided by main content

2. renderResponse()

Wrap in #type 'html'

hook_page_(top|bottom)()

drupal_render() of entire HTML document

new Response()

DialogRenderer

new AjaxResponse()

drupal_render() for main content

Calculate title if not provided by main content

Get dialog options from Request

Add **OpenDialogCmd**

ModalRenderer

new AjaxResponse()

drupal_render() for main content

Calculate title if not provided by main content

Add **OpenModalDialogCmd**

AjaxRenderer

new AjaxResponse()

drupal_render() for main content

Add **InsertCommand**
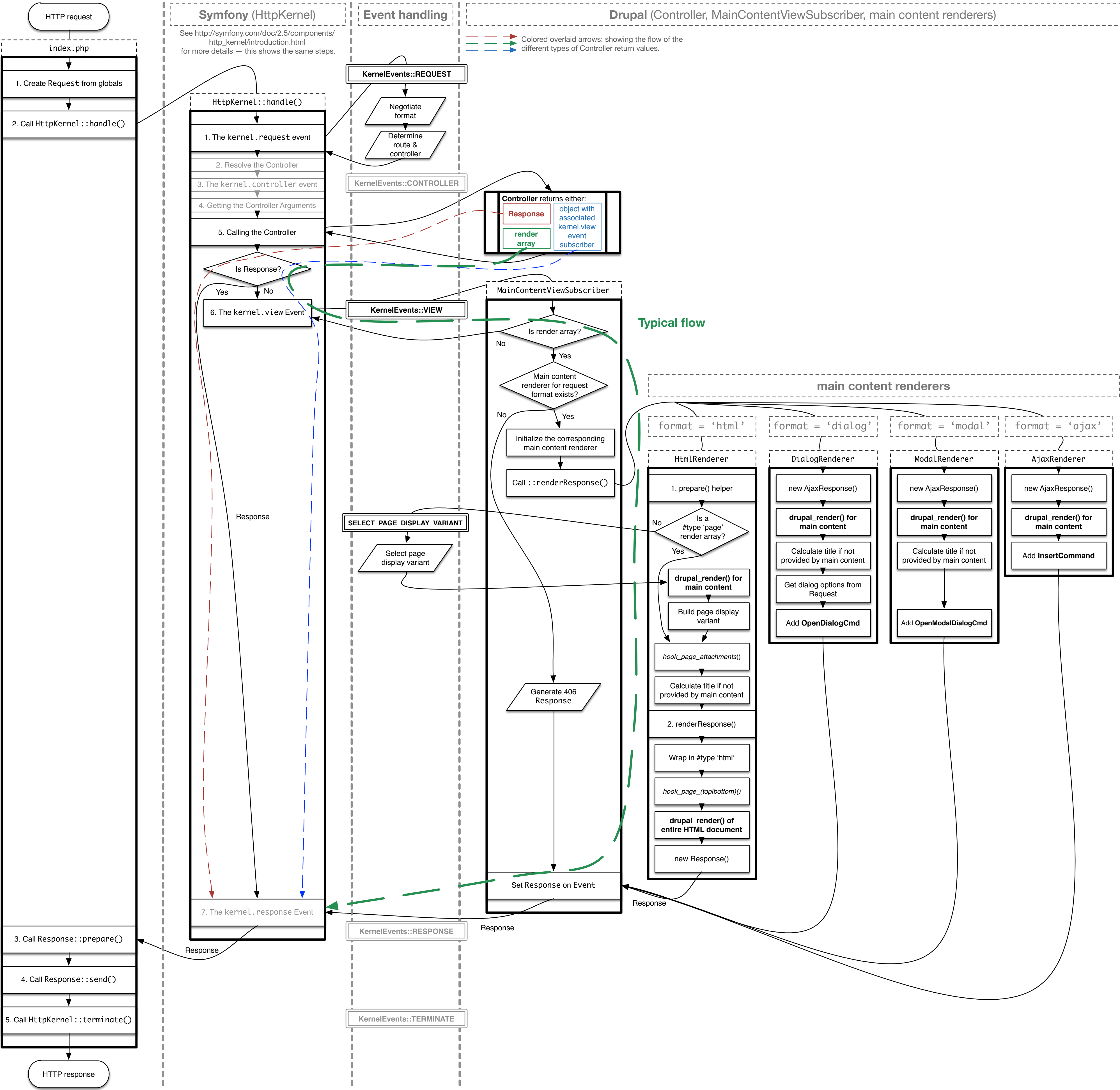
Response

Explanation

Some parts are greyed out because they're of lesser importance: they do happen, but are not crucial for understanding the flow.

Example: **GET /admin.**

Typically, the format will be **html**, but it could also be **ajax, dialog, modal, json, hal_json, and more.**
See: Symfony's Request::setFormat().

Example: **/admin path → system.admin route**
That route has the following attributes:
    _content: '\D\s\C\SystemController::systemAdminMenuBlockPage'
    _title: 'Administration'

_controller is set to '\D\s\C\SystemController::systemAdminMenuBlockPage'

\D\s\C\SystemController::systemAdminMenuBlockPage is called, it returns a render array with no #type.

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 renderer services. These are **tagged with render.main_content_renderer.** Contributed modules can additional renderers.

The flow for all four formats supported in core in their respect is displayed next, but the **explanations below only apply to the HtmlController!**
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 #type 'page' render array.

hook_page_attachments() operates on #type 'page'.

Only accepts render arrays of #type 'page'!

hook_page_top() and hook_page_bottom() operate on #type 'html'.

Renders the 3 parts of the HTML body: *page_top, page, page_bottom* (where *page* corresponds to #type 'page' and hence **page.html.twig**). **Then** renders the entire HTML (i.e. the **html.html.twig** template).

Example: response for the path **/admin.**