HttpCache

Content Cache

eZ Publish uses Symfony HttpCache to manage content "view" cache with the expiration model.

An additional **X-Location-Id** header is added in the response for identification (for details see cache purge document).

Configuration

```
ezpublish:
    system:
        my_siteaccess:
            content:
                view_cache: true      # Activates HttpCache for content
                ttl_cache: true       # Activates expiration based HttpCache for
content (very fast)
                default_ttl: 60       # Number of seconds an Http response cache is
valid (if ttl_cache is true, and if no custom s-maxage is set)
```

Cache and Expiration Configuration for error pages

It is normal to want to set the `default_ttl` setting above high to have a high cache hit ratio on your installation. As the system takes care of purges, the cache rarely becomes stale.

However, a few redirect and error pages are served via the ContentView system, and if you do set a high `default_ttl`, you should make sure to set those pages to a much lower ttl to avoid issues caused by that. For this you can use the FOSHttpCacheBundle matching rules feature to specify a different ttl time:

```
fos_http_cache:
    cache_control:
        rules:
            # Make sure cacheable (fresh) responses from eZ Platform which are
errors/redirects gets lower ttl than default_ttl
            -
                match:
                    match_response: "response.isFresh() && ( response.isServerError()
|| response.isClientError() || response.isRedirect() )"
                headers:
                    overwrite: true
                    cache_control:
                        max_age: 5
                        s_maxage: 20
```

Similarly, if you want to apply performance tuning to avoid crawlers affecting the setup too much, you can also set up caching of generic 404's and similar error pages in the following way:

```
fos_http_cache:
    cache_control:
        rules:
            # Example of performance tuning, force TTL on 404 pages to avoid crawlers,
etc., taking too much load
            # Should not be set too high, as cached 404's can cause issues for future
routes, URL aliases, wildcards, etc.
            -
                match:
                    match_response: "!response.isFresh() && response.isNotFound()"
                headers:
                    overwrite: true
                    cache_control:
                        public: true
                        max_age: 0
                        s_maxage: 20
```

Making your controller content cache aware

Sometimes you need that your controller's cache expires in the same time than a specific content (i.e. ESI sub-requests with `render` twig helper, for a menu for instance). To be able to do that, you just need to add **X-Location-Id** header to the response object:

```
use Symfony\Component\HttpFoundation\Response;

// In a controller
// "Connects" the response to location #123 and sets a max age (TTL) of 1 hour.
$response = new Response();
$response->headers->set( 'X-Location-Id', 123 );
$response->setSharedMaxAge( 3600 );
```

Making your controller user cache aware

If the content you're rendering depends on the user permissions, then an additional header must be set for this as explained in Context aware HTTP cache

- Purge