Concept - Configurable cache constraints on renderables

Your Rating: ☆☆☆☆☆   Results: ★★★☆☆ 119 rates

Introduction

> ⓘ **Draft for ?**
>
> While rendering a page the renderables and their rendering models should be able to influence how the page is cached.
>
> ⬆ **MAGNOLIA-3902** - Configurable cache constraints on renderables `OPEN`

Requirements

- Caching should be configurable on renderable definitions
- Caching should be changable in RenderingModel and in template script, this should override what's on the renderable definition

Implementation

- Renderers will look at renderable definitions and set cache headers

Implications on Browser Cache Policy

Browser cache policy will never set an expiration later than that requested via cache headers

Problems, thoughts, notes

- It is safe to set headers after the response has been committed, they will be ignored. So with caching turned off these additional headers have no effect
- I believe in the context that "constraints" originally meant, essentially cache-control headers (e.g Concept - Cache Header Negotiation)
- However, another thing to keep in mind is that with Concept - Cache arbitrary objects we could be using the cache to store cached components /fragments, and we'll probably want those to avoid different kinds of cache keys (simple composites: (workspace,uuid), but perhaps stuff like (workspace,uuid,username) or any other combination of traits (workspace,uuid,country,...

- We shouldn't introduce cache-specific logic in the rendering module - not without extracting rendering/templating out of main and fixing possibly dependency issues at the very least.
- I (GJ) am not convinced in the value of configurability for those headers - if something's not cacheable, it likely means there is already some code doing some uncacheable things with content. That code might as well set headers on the response themselves. Of course that particular model might use bits of configuration (from itself, from the definition, from some module, ...). Of course, if we turn that into simple configuration items (cacheable boolean, perhaps a TTL, then yes; anything more complex will bring the complexity of cache-control in configuration, which is probably not going to help)