

# To Spec Or Not To Spec

Yet Another Consideration When Choosing a Media Type

Scott Smith  
REST Fest US 2016

# About Me

- Senior Developer at NPR
- NPR Story API
- Public Media Platform (PMP) API (<http://publicmediaplatform.org>)
- <https://twitter.com/scottsmith130>
- <https://github.com/scottsmith130>
- <https://linkedin.com/in/scottsmith130>

# Inspired By True Events

- Media type specifications can be too specific or not specific enough for your API
- Choosing a spec with too many constraints can provide stability for clients but could make adapting the API difficult
- Choosing a spec with too few constraints can provide flexibility for the API but could make client support harder

# Collection.Doc+JSON (CDOC)

- <http://cdoc.io>
- Customized for the PMP API
- Permissions logic part of the spec
- What if we need different permissions logic?
  - Change the spec –
    - Change the permissions logic, or split the permissions logic out into its own extension spec
    - What if we need to change something again later?
  - Use a profile –
    - Can't be used to change the spec, only to add to it
    - Can be used with a more extensible media type, but may require refactoring code to support a new media type

# Hypertext Application Language (HAL)

- [http://stateless.co/hal\\_specification.html](http://stateless.co/hal_specification.html)
- Generic hypermedia controls for JSON documents
- Links can be any valid URI
- What if we have a need to use a URI fragment?
  - HAL spec says to look at RFC3986
  - RFC3986 says resolution of fragments is up to the media type
  - Server is free to use URI fragments in links, but clients may have no idea what to do with them
  - Change the spec – not easy to get changes into the HAL spec
  - Use a profile – only clients that know about the profile will benefit

# Hypertext Application Language (HAL) (cont.)

- Clients are not required to use embedded resources
- What if we have a need to embed resources that don't exist externally?
  - If there is no link associated with the embedded resource, clients may ignore the resource altogether
  - Use a URI fragment for the link – clients may not know what to do with it
  - Change the spec – not easy to get changes into the HAL spec
  - Use a profile – only clients that know about the profile will benefit

# Some Takeaways

- Customized media types are sometimes appropriate, but they can lead to a lot of rework later if changes are needed and the media type is not extensible where it needs to be
- Generic media types are much easier to extend, but clients must still be specific enough to understand the extensions
- If in doubt, start generically and extend
  - Clients/Servers must still be written to handle new media types or new profiles
  - Profiles may not require as much overhead up front, and can always be converted to a media type later if needed
- Ultimately, your media type choice and extension support may depend on how wide your client base is and how much adaptability you need
  - If you control all your clients, the sky is the limit
  - If you need to support clients that can only handle what is defined by the media type, the choice may be more difficult

# Questions?

- <https://twitter.com/scottsmith130>
- <https://github.com/scottsmith130>
- <https://linkedin.com/in/scottsmith130>