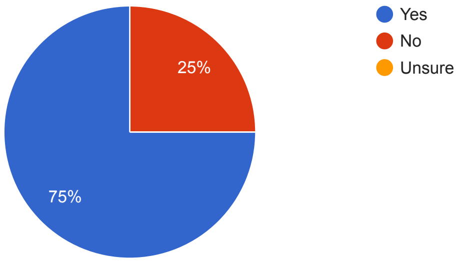
Feedback on current RFC 6906bis draft

4 responses

Concept of a profile

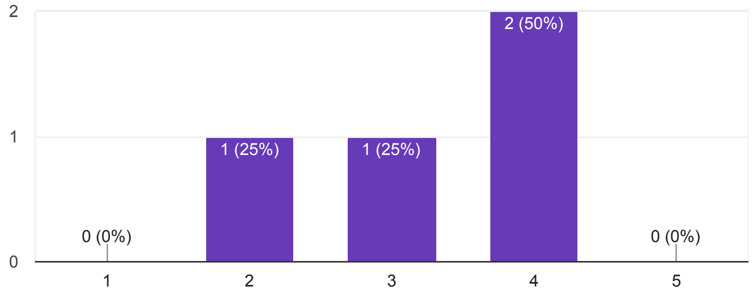
Does RFC 6906bis contain a definition of a “profile”?

4 responses



Is RFC 6906bis clear about what a profile is?

4 responses



To the best of your understanding, what is a profile according to RFC 6906bis?

4 responses

I don't know.

A profile is a documented way to use use the media type.

A set of constraints that go on top of media type constraints. Profiles can also be defined so that they can be used across media types (an example would be profiles as used in ODRL)

A refinement of the media type, to give more semantics and meaning to a more generic media type.

What can/needs to be done to make the definition of a profile more clear?

4 responses

I believe that both *media-type* and *profile* should aim to reach acceptable degrees of interoperability between systems, but the document (in my opinion) is not precise enough.

Perhaps, for the sake of clarity, the document could include 'tests' that exemplify what is the level of interoperability that 'profile' addresses.

For example, I can imagine one for 'mediatype':

(1) Requirement: One can translate between 'media-types': "If the syntax used by the sender is different than the one used by the receiver, then a translation or mapping mechanism can be applied. For example a RDF-Xml serialization can be translated into RDF-Json through a simple transformation. If an accurate translation of syntaxes is possible, then we say that systems can interoperate at syntactical level."

Unfortunately I cannot imagine a similar one for profile, because the document defines it in terms of 'semantics' (being this a very ambiguous term). Perhaps the term 'representation' (that use the same syntax) is more appropriate. Then, a similar test to (1) can be added to 'media-type' and 'profile'.

For example: if I have a profile-A: "Representation with high detail", and profile-B: "A Summary for very small devices"

one could translate *without loosing information* from:

mediatype-A:profile-A <-> mediatype-B:profile-A

mediatype-A:profile-B <-> mediatype-B:profile-B

but not necessarily from:

mediatype-A:profile-A <-> mediatype-B:profile-B

Perhaps *the semiotic ladder* of the frisco report is useful here:
http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/FRISCO-report-1998.pdf

I'd be more assertive about profile URI point some place useful (while leaving it optional). For example, remove "consider to" from the next to last sentence of section 3. "profile maintainers SHOULD make the profile URI dereferencable ..." is better.

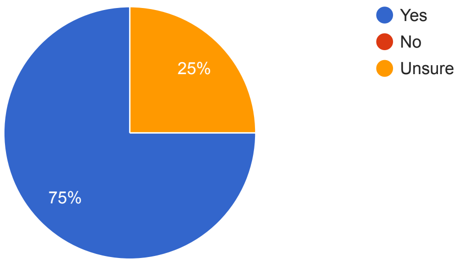
(This is an extra comment since the comment field on the last page doesn't work...) I find the use of the Prefer header problematic since the server can only state that it applied a preference for Profile-A (through Preference-Applied) but not that it didn not apply the preference for Profile-A but used Profile-B instead. That way the client does not know if the absence of the Preference-Applied header means that the server did not understand the preference (e. g. because it does not understand Prefer) or because it understood the preference but could not honour it.

Provide concrete examples

Difference with a media type

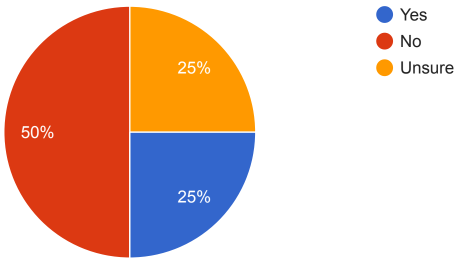
According to RFC 6906bis, is there a difference between a profile and a media type?

4 responses



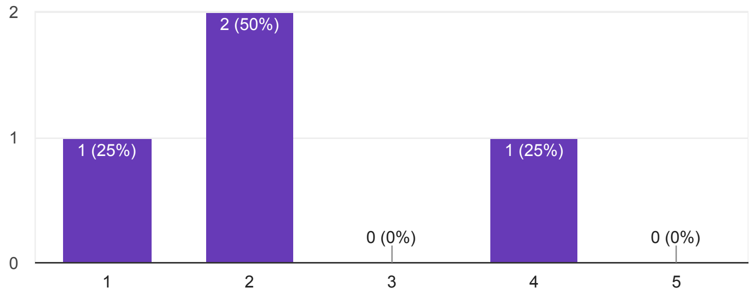
Does RFC 6906bis state what the difference between a profile and a media type is?

4 responses



Is RFC 6906bis clear about what the difference is between a profile and a media type?

4 responses



To the best of your understanding, what is the difference between a

profile and a media type?

4 responses

mediatype is about syntax, media type about represented elements.

HTTP doesn't allow declaring, or negotiating, more than one media type for a response body. Profiles are an attempt to work around this flaw.

A media type generally defines the syntax of the resource. Specialised media types can also specify additional constraints (and are then sometimes served with +... extension, e. g. application/marcxml+xml that is a specialisation -- perhaps even a profile -- of application/xml in the sense that both can be processed by an XML parser). The main difference to me is that a profile MAY be used across media types, so a profile defines namespaces, attributes, elements, vocabularies, element order and cardinality etc. in a way that can be used with several media types. An important case is RDF data served in one profile but in several serialisations.

While profiles and media types can serve the same need, media types are more costly to create and are a required building block for HTTP to even work. For media types to be usable, they need to be stable and new ones should be created at glacial speed. Since profiles just extend or refine media types and are cheap to create, they can be much more volatile and proprietary. Anyone can create a profile at any moment without giving it much thought, which is not the case with a media type.

What can/needs to be done to make the difference with media types more clear?

4 responses

add tests.

I fear that is a lost cause. :) They are, fundamentally, one concept separated by a historical artifact. `http://example.com/podcat-specs.html` is a profile of `application/atom+xml` which is a "profile" of `application/xml` which is a "profile" of `text/plain` which is a "profile" of `application/octet-stream`. Each of those levels "describes additional constraints that can be used to process a resource representation". Each of those levels (except octet stream) could be a "profile".

Given that only superficial differences exist between "profile" and "media type" (eg, one is URI and the other is a registered string) it is impossible to articulate a fundamental distinction.

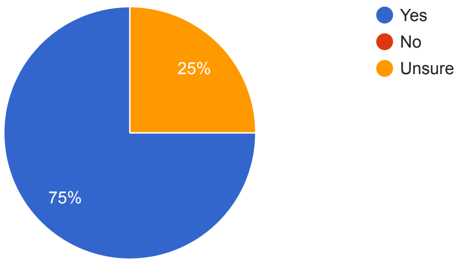
More emphasis on the cross-media type use of profiles.

Provide concrete examples.

Difference with a schema

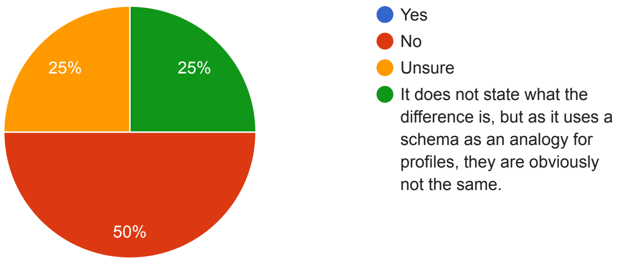
According to RFC 6906bis, is there a difference between a profile and a schema?

4 responses



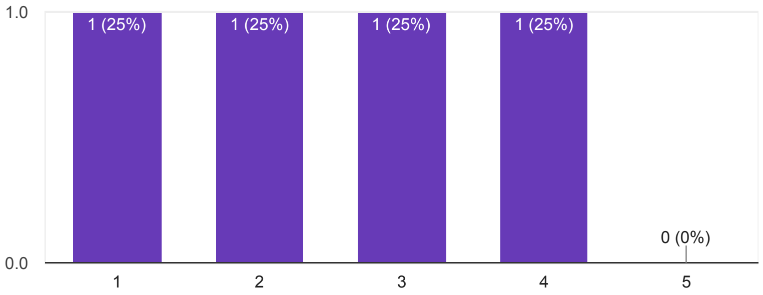
Does RFC 6906bis state what the difference between a profile and a schema is?

4 responses



Is RFC 6906bis clear about what the difference is between a profile and a schema?

4 responses



To the best of your understanding, what is the difference between a

profile and a schema?

4 responses

a schema relates to constraints. a profile are distinct representations in the same syntax (or level of detail)

A profile defines a set of constraints. A schema is a formal definition of a certain class of constraints. Profiles may, but are not required to, use schemas to define their constraints.

A schema is an implementation of a profile using a specific schema language, e. g. JSON-Schema for application/json or application/ld+json, XML schema for application/xml and SHACL or ShEx for RDF serialisations. In that sense, a profile is an abstract construct that can formally specified through a schema.

A profile is more about semantics. meaning and understanding, while a schema is more about validation, data typing and technical details. Sort of like OWL vs. XSD.

What can/needs to be done to make the difference with schemas more clear?

3 responses

to point to public RFC definitions on what is syntax, what is semantics etc.

nothing

Provide more description as well as concrete examples.

Other comments

(Optional) Please enter any other comments you might have on this RFC draft.