2015-12-18 - Fedora API Extensions Meeting

Dial In Details

Date: Friday December 18, 11am EST (-5 UTC)

- Dial-in Number: (712) 775-7035
    - Participant Code: 479307#
    - International numbers: Conference Call Information
    - You may also call in using the VoIP dialer from a web browser, or Android/iOS apps
- IRC:
    - Join the #fcrepo chat room via Freenode Web IRC (enter a unique nick)
    - Or point your IRC client to #fcrepo on irc.freenode.net

Meeting Goals

1. Review/prioritize API-X requirements
2. Agree on first concrete development steps

Attendees

- Aaron Birkland
- Unknown User (acoburn)
- Stefano Cossu
- Elliot Metsger
- Joshua Westgard
- Andrew Woods
- Bethany Seeger
- Nick Ruest

Agenda

1. Discuss/prioritize initial list of High-level Requirements
    a. Pick a list of requirements that could potentially drive first sprint(s)
    b. See comments on high-level requirements page:
        i. Are deployment-related issues appropriate as requirements, or are they goals?
            1. If they are goals, do we have other items that are also goals, but not requirements?
        ii. Is it appropriate for an API-X extension to use native java APIs such as fcrepo-kernel-api, or should they use standardized and codified fedora HTTP APIs?
2. Discuss list of potential Proof of Concept ideas. Do we want to pursue one or more of these as a team? Individually?
3. Rough implementation timeline

Minutes

Agenda Item 1: Review initial list of High Level Requirements

A.B. Review design considerations that were extracted from last meeting's requirements.

A.W./E.M./A.C. - changes are in-line with what was discussed

A.B. Review the revised list of high-level requirements for category appropriateness and phrasing.

A.W. 1a. clarify "deploy" as the ability to deploy versus being required to deploy. Requirement to interact with the framework using HTTP?

A.C. HTTP might not be appropriate for interacting with the framework in a clustered environment. But having extensions being invoked by HTTP sounds reasonable.

A.B. Proposed adding another requirement that extensions shall be invoked by HTTP requests.

A.C. Proposed language that extensions are exposed as HTTP endpoints.

E.M. 2a. commented that the current phrasing places the requirement on the extension; rephrase to place requirement on the framework for clarity

A.B. Proposes to collapse items in #2 (Extensions) into #1 (Infrastructure), and remove "Extensions" as a category. Rename #1 to "Infrastructure". #2 is now Patterns.

J.W. Question necessity of phrasing "outside world" in the bullet under 1a.

A.B. Revised bullet removed text.

A.B. Review item 2, now Patterns.

E.M./A.B. proposed HTTP RFC compatibility as a design consideration *ACTION*

A.C. Clarified that the items in #2 Patterns are not exclusive

A.B. None of these requirements are exclusive, they should be considered minimum requirements

A.B. *ACTION* to add this clarifying language to the high level requirements.

A.B. Review item 3, now Service Discovery & Binding

E.M. Suggested wording change, "For a specific object in the repository it shall be possible to..." for 3b, 3c, 3d.

All: Concur.

A.B. Proposed keeping 3a-f, but tweak for readability.

S.C. Should extension metadata or documentation be a requirement?

A.B. Sees that encapsulated/implicit as a part of extension configuration

A.B. *ACTION* Add as a design consideration extension metadata/configuration needed, including link to documentation

A.C. Sees 3e, f as significant requirements; how extensions register themselves with the framework and select which objects they act on

A.C. Existing language is vague, but simple. Could we, for example, define the "payload" of information that would be exchanged between and extension and the framework for registration/deployment/de-registration?

A.C. Goal to standardize how an extension registers with the framework

A.B. Propose to add a requirement to #1 Infrastructure that requires a specification for the interaction between an extension and the framework with regard to service discovery and binding.

All: Concur.

A.B. Added high level requirement 1e.

Agenda Item 2: Proof-of-Concept

A.B. Simple proofs of concept that are chosen because they interact with the framework in different ways

S.C. Validation PoC would be simpler if deletion were not a consideration.

A.B. Is approach of using the PoC to drive development of the framework useful, without digging into the details of each PoC (time constraints).

All: Concur.

Agenda Item 3: Implementation timeline

A.B. JHU till now has been working on API-X on margin time.

A.B. Formal time for API-X being allocated late January, early Feb.

A.B. Proposed scheduling a sprint for mid-January/beginning of February.

A.B. Model the sprint after an exemplar like a Web AC sprint.

A.W. Proposed detailing the goals of the sprint, what would be done, and then this group can discuss

E.M. Suggested a high-level roadmap would be useful

A.B. Proposed assembling the details/goals of the first sprint and then discussing them at the next meeting

J.W. Agreed with making the spring details public. To the degree that UMD and API-X goals align, they can contribute resources.

A.B. Proposed next meeting Jan 8, 2pm EST.

N.R. can't make that time but indicated we shouldn't let that stop us from meeting.

S.C. volunteered to create a diagram that shows where API-X sits in a Fedora-based ecosystem/infrastructure.

A.B. Concurred. Suggested some wiki gardening needs to take place, especially with regard to presenting a unified or single voice of the API-X working group that reflects the decisions made thus far.