

# API-X Demo/Milestone 1

## Introduction

Welcome to the first milestone of the Fedora API Extensions (API-X) framework. This milestone is the first concrete implementation of API-X, and is accompanied by a set of evaluation tasks comprising a demo that showcases its current capabilities and serves as a basis for soliciting feedback from the community.

The API-X framework [positions itself](#) as HTTP middleware in a repository ecosystem. It serves as a transparent mechanism to expose services that act on repository resources, and/or to mediate the interaction between a client and a repository resources vis à vis services. Existing ad hoc services may be seamlessly discovered and interacted with by clients; likewise, novel services may be authored and incorporated into the ecosystem. The demo includes [services provided by Amherst College](#), and will demonstrate the capability of API-X to incorporate these services into the repository layer.

The API-X [codebase](#) has matured to the point of producing this milestone, but should not be mistaken as production ready just yet. In particular, very little attention has been paid to optimization. We would appreciate if you let us know of any issues or suggestions encountered performing the tasks, especially if they would prevent you deploying API-X in production. Furthermore, the topology of the services distributed in the milestone shouldn't be taken as an endorsement of any particular deployment strategy.

## Demo Environment

The API-X milestone demo uses the [docker](#) environment to package pre-configured instances of Fedora, API-X, and services into lightweight "containers" that behave much as if they were deployed in separate virtual machines. [docker-compose](#) is used to run the demo on an evaluator's local machine. Using docker-compose, the demo environment can be created, paused/resumed, or wiped clean at any point.

Please read the [setup guide](#) for information on how to install and verify Docker on your own machine.

## Providing feedback

Please provide general feedback or questions on the demo in the form of comments on this page.

Bugs and/or feature requests can be submitted [on github](#). We will try to fix any bugs affecting the demo as soon as possible, and incorporate fixes into the docker images so that they can be pulled into your demo environment as soon as they are available. Feature or improvement requests on github are especially welcome.

When performing the evaluation tasks, please keep the following questions in mind when formulating feedback

- Do any aspects of API-X seem particularly useful or not useful for your own use cases?
- Was any step particularly difficult or burdensome, or surprisingly easy? Do you have any suggestions for improvement?
- Do you feel any particular topic should be explored in greater depth in a future milestone demo/evaluation?
- Do you have any unanswered questions?

## Evaluation Tasks

To participate in the demo, please:

- Install the [requirements](#) (e.g. Docker and curl) if not already present
- Read the [getting started](#) guide for using docker-compose to start Fedora, API-X and surrounding services
- [Verify](#) that the demo has started successfully
- Read [running the evaluation](#) before you start for some helpful information to make the process go smoothly.

Once everything is up and running, you can start working through the evaluation tasks. Most (but not all) current API-X features are encountered in one form or another. The tasks do not have to be done linearly, and they do not have to be done all at once.

1. [Resources, URIs, and Proxies](#)
  - a. [Look at a Fedora object](#) via direct and proxy URIs in order to compare and contrast representations
  - b. [Look at proxied binary resources](#)
2. [Service Documents](#)
  - a. [Retrieve and analyze a service document](#)
  - b. [Look for a specific service](#)
3. [Interacting with exposed services](#)
  - a. [A passive resource-scoped service](#) (i.e. just use a GET on a service endpoint to get FITS-extracted metadata)
  - b. [Links within exposed services](#)
  - c. [Query parameters](#) (explore a fascinating image manipulation extension based on [ImageMagick](#))
4. [Loading and deploying extensions](#)
  - a. [A tour of the service registry](#)
  - b. [The loader extension](#)
  - c. [Loading an extension manually](#)
  - d. [Auto-loading extensions](#) (see how services can self-register themselves and the extensions they implement)
5. [Ontologies and Binding](#)
  - a. [Simple binding by rdf:type](#)

- b. [Binding by inference](#)
- c. [Extensions as ontologies](#)
- d. [The ontology registry](#)

## Participation

We need participants!

If anyone is interested in evaluating the demo, feel free to place your name as an participant on this page, or just get started and provide feedback when inspired. We are asking participants to 1) provide feedback on the *content* of the evaluation tasks and guide (e.g. are there tasks we are missing?) and 2) download the demo and execute the tasks, and provide feedback on the experience.

People who have helped participate in the creation or evaluation of the demo include:

- [Elliot Metsger](#) - Johns Hopkins, Software Engineer, API-X stakeholder and developer
- [Ruth Duerr](#) - Ronin Institute for Independent Scholarship, API-X stakeholder, Data Manager/Informaticist, ex-operations manager, ex-archive manager, ex-systems engineer, ex-software engineer & developer
- [Joshua Westgard](#) - University of Maryland Libraries, API-X stakeholder, repository application manager
- [Aaron Birkland](#) - Johns Hopkins, Software Engineer, API-X developer
- [Unknown User \(acoburn\)](#) and [Bethany Seeger](#) - Amherst College, API-X stakeholder and developers of the [Amherst extensions](#) used throughout the demo