Muck acceptance tests v2

Lars Wirzenius / The Ick project

Contents

Chapter 1

Introduction

Muck is a persistent in-memory JSON store with an HTTP API and advanced access control using signed JWT access tokens. This document presents its automated acceptance tests, using a (for now hypothetical) language similar to the Gherkin langauge implemented by Cucumber.

Chapter 2

A happy path scenario

This scenario does some basic resource management via the Muck API.

Start Muck. This also sets up access to it for the user by getting an access token, which will be used for all requests.

given a running Muck

Check server status.

then there are *0* resources in Muck

Create a simple resource. Remember its id.

given I am *tomjon*
when I create a resource `{"foo": "bar"}`
then there is *1* resource in Muck
and remember the resource id as *ID*
and rememer the resource revision as *REV1*

Retrieve the resource.

when I fetch resource *ID*
then I get `{"foo": "bar"}`
and it is mine
and it has revision *REV1*

Make sure another user can't retreive, update, or delete the resource.

given I am *verence*
when I fetch resource *ID*
then it doesn't exist

when I update *ID*, revision *REV1*, with `{"foo": "somethingelse"}`
then it doesn't exist

when I delete *ID*
then it doesn't exist

Update the resource.

given I am *tomjon*
when I update *ID*, revision *wrong*, with `{"foo": "somethingelse"}`

then it doesnt't work

when I update *ID*, revision *REV1*, with `{"foo": "somethingelse"}`
then it works
and remember the resource revision as *REV2*

Check the resource has been updated.

when I fetch resource *ID*
then I get `{"foo": "somethingelse"}`
and it is mine
and it has revision *REV2*

Restart Muck. The resource should still exist.

when Muck is restarted
and I fetch resource *ID*
then I get `{"foo": "somethingelse"}`
and it is mine
and it has revision *REV2*

Search for the resource. First with a condition that is no longer true.

when I search for *foo* being *bar*
then there are no matches

Now search for the correct value.

when I search for *foo* being *somethingelse*
then I get only resource *ID*

Delete the resource.

when I delete *ID*
then it works

when I fetch resource *ID*
then it doesn't exist

Restart Muck again. The resource should not exist.

when Muck is restarted
when I fetch resource *ID*
then it doesn't exist

All done.