Muck acceptance tests

Lars Wirzenius / The Ick project

Contents

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.

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.

> **given** I am *tomjon*
> **when** I make request *GET /status*
> **then** status code is *200*
> **and** response body is `{"resources":0}`

Create a simple resource. Remember its id.

> **when** I make request *POST /res* with body `{"foo":"bar"}`
> **then** status code is *201*
> **and** response has header *"Muck-Owner: tomjon"*
> **and** resource id is remembered as *ID*
> **and** resource revision is remembered as *REV1*
>
> **when** I make request *GET /status*
> **then** status code is *200*

> **and** response body is `{"resources":1}`

Retrieve the resource.

> **when** I make request *GET /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *200*
> **and** response body is `{ "foo": "bar" }`
> **and** response has header *"Muck-Id: ${ID}"*
> **and** response has header *"Muck-Revision: ${REV1}"*
> **and** response has header *"Muck-Owner: tomjon"*

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

> **given** I am *verence*
> **when** I make request *GET /res with header "Muck-Id: ${ID}"*
> **then** status code is *404*
>
> **when** I make request *PUT /res*
> with header *"Muck-Id: ${ID}"* and
> header *"Muck-Revision: ${REV1}"* and
> body `{"foo": "foobar"}`
>
> **then** status code is *404*
>
> **when** I make request *DELETE /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *404*

Update the resource.

> **given** I am *tomjon*
> **when** I make request *PUT /res* with header *"Muck-Id: ${ID}"* and
> header *"Muck-Revision: wrong"* and
> body `{"foo": "foobar"}`
> **then** status code is *400*
>
> **when** I make request *PUT /res* with header *"Muck-Id: ${ID}"* and
> header *"Muck-Revision: ${REV1}"* and
> body `{"foo": "foobar"}`
> **then** status code is *200*
> **and** resource revision is remembered as *REV2*

Check the resource has been updated.

> **when** I make request *GET /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *200*
> **and** response body is `{"foo": "foobar"}`
> **and** response has header *"Muck-Id: ${ID}"*
> **and** response has header *"Muck-Revision: ${REV2}"*
> **and** response has header *"Muck-Owner: tomjon"*

Restart Muck. The resource should still exist.

> **when** Muck is restarted
> **and** I make request *GET /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *200*
> **and** response body is `{"foo":"foobar"}`
> **and** response has header *"Muck-Id: ${ID}"*
> **and** response has header *"Muck-Revision: ${REV2}"*
> **and** response has header *"Muck-Owner: tomjon"*

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

> **when** I make request *GET /search* with body
> `{"cond": [{"where": "data", "field": "foo", "pattern": "bar","op": "=="}]}`
> **then** status code is *200*
> **and** response body is *{"resources": []}*

Now search for the correct value.

> **when** user *tomjon* makes request *GET /search* with body
> `{"cond": [{"where":"data","field":"foo","pattern":"foobar","op":"=="}]}`
> **then** status code is *200*
> **and** response body is `{"resources": ["${ID}"]}`

Delete the resource.

> **when** user *tomjon* makes request *DELETE /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *200*
>
> **when** user *tomjon* makes request *GET /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *404*

Restart Muck again. The resource should not exist.

> **when** Muck is restarted
> **and** user *tomjon* makes request *GET /res* with header *"Muck-Id: ${ID}"*
> **then** status code is *404*

All done.