Dying Documentation

Analysis report, generated on 2018-03-19T21:47:46.204Z[Etc/UTC]

Table of content

Sample text in `intro.md`

1. Bootstrapping

Uri: features/bootstrapping.feature

- **As** an external application
- **I want** to call a REST API
- **So that** I can interact with this application

Description:

1.1. the web application is responding


Given the API runs


When I visit the root path


Then I get any response

1.2. the web application is healthy


Given the API runs


When I visit the health check


Then the status code is 200

2. Event recording

Uri: features/event.feature

- **As** an monitored device
- **I want** to record events
- **So that** they can be digested into reports later

Description: A device goes through various states, such as from being used to being idle. The new state is sent to the application.

The application does not need to know the device in advance before accepting the events. Not all events received by the application should be recorded. If two consecutive events from the same device describe the same state, then that event is discarded.

2.1. the first event of a device is recorded

- Given there are initially 0 events for device A
- And there are initially 2 events for device B
- When I send an event for device A with state some_random_state
- Then there are 1 events for device A
- And there are 2 events for device B
- And the first event for device A has state some_random_state

2.2. the same event sent twice in a row is only recorded once

- Given there are initially 10 events for device A
- When I send an event for device A with state some_random_state
- And I send another event for device A with state some_random_state
- Then there are 11 events for device A
- And the last event for device A has state some_random_state

3. Opera browser support

Uri: features/support-opera.feature

- **As** a html web frontend running on Opera Mobile
- **I want** to call the API with cross-origin requests
- **So that** that mobile users are not affected

Description: Opera Mobile appears to be more restrictive on CORS support. Unlike Chrome which appears to accept asterisks, Opera seems very picky on the Access-Control-Allow-Origin, Access-Control-Request-Method and Access-Control-Allow-Headers.

3.1. the preflight OPTIONS request contains tailored headers for Opera

- Given the API runs
- When I perform an OPTIONS request to the health check with Origin header set to https://test.example.com
- Then the Access-Control-Allow-Origin header is https://test.example.com
- And the Access-Control-Request-Method header contains all of GET,POST,PUT,PATCH,DELETE,OPTIONS,HEAD,TRACE
- And the Access-Control-Allow-Headers header contains all of Authorization,X-Module

Sample Steps


Given a passed step


And a failed step


When a pending step


But an undefined step


Then a skipped step