Client Functional Spec

Release 0.7

Table of Contents

1 Schema Management 2

1.1 General 2

1.2 Schema deploy directory location 2

2 Extension Management 3

3 API Reference 4

3.1 General 4

3.2 Methods 4

3.2.1 /event 4

3.3 /session 5

3.3.1 GET /session/:id 5

3.3.2 PUT /session/:id 7

4 Future 8

# Concepts

## Connections

Connections are made to a schema name by passing it to the VariantClient.getConnection() method. Connections are pseudo

# External Code

## Client-Side User Hook Listeners

# API

## General

## Top Level Methods

### getConnection(): VariantSession

## Methods On the Connection Object

### getSession(): VariantSession

Trigger a user-defined event.

|  |  |  |
| --- | --- | --- |
| Response Headers | | |
| Content-Type | text/plain; charset=utf-8 | |
| Request Body | | |
| {  "sid": <String>,  "name": <String>,  "value": <String?>,  "ts": <Number?=NOW>,  "params": <Array?> [  {  "name": <String>,  "value": <String?>  },  ...  ],  } | | |
| session-id | | Current variant session ID. |
| event-name | | The name of the event. Any string. |
| event-value | | Value of the event. Any string. |
| event-create-date | | Event creation date, as Epoch time. Optional. If not given, defaults to now. |
| event-parameters | | A map of event parameters. Optional. |
| Response Headers | | |
|  | | |
| Response body | | |
| None. | | |
| Response Codes | | |
| 200 OK | | Event triggered. |
| 400 BAD\_REQUEST | | Invalid request body. One of a number of user errors was detected in the request body, e.g. a missing required parameter. An additional message will be provided in the HTTP Status header. |
| 403 FORBIDDEN | | Invalid session ID. The session ID provided is not known. |

## /session

|  |  |
| --- | --- |
| GET /session/:id | |
| Get session by ID.   |  |  |  | | --- | --- | --- | | Headers | | | | Content-Type | text/plain; charset=utf-8 | | | Request Body | | | | None | | | | Response body | | | | {  "sid": session-id::String,  "ts": timestamp::Number,  "schid": schema-id::String,  "req": request-def::Object ?,  "states": [  {  "state": state-name::String,  "count": visit-count::Number,  },  ...  ],  "tests": [test-name::String,...]  "disqualTests": [test-name::String,...]  }  request-def ::=  {  "state": state-name::String,  "status": status::String,  "comm": is-commited::Boolean,  "params": [  {  "key": param-name::String,  "val": param-value::String  },  ...  ],  "exps": experience-list::List[String]  } | | | | session-id | | Current variant session ID. | | timestamp | | Session creation timestamp, as the Unix Epoch time, i.e. the number of milliseconds since January 1, 1970, 00:00:00 GMT | | schema-id | | Variant schema ID in effect when this session was created. | | state-name | | The name of a visited state. | | visit-count | | Number of times this state has been visited by this session. | | test-name | | The name of a traversed test | | is-qualified | | Is this session qualified for this test. | |  | | | | Response Headers | | | |  | | | | Response Codes | | | | 200 OK | | Session found and returned. | | 404 NOT\_FOUND | | No session with that ID on server. |  PUT /session/:id | |
| Save or replace user session by session ID. Idempotent. Body is not parsed but saved in the session store under the ID. Only parsed if required instantiation on Server. |  |

|  |  |  |
| --- | --- | --- |
| Headers | | |
| Content-Type | text/plain; charset=utf-8 | |
| Request Body | | |
| See response body of GET /session/:id | | |
| Response body | | |
| None. | | |
| Response Codes | | |
| 200 OK | | Session created or replaced. |
| 400 BAD\_REQUEST | | Invalid request body. One of a number of user errors was detected in the request body, e.g. a missing required parameter. An additional message will be provided in the HTTP Status header. |

# Future

* Connection.getSession() methods are inconsistent in that they are idempotent if the session exists, but not if it doesn’t because session ID tracker is reinitialized for each call.