[DRAFT] Single endPoint JSON Communication

Joe Koop

October 18, 2023

Contents

Single endPoint JSON Communication (SPJC) is a simple application-layer message-passing protocol that uses HTTP (RFC 9110) and is intended for web2 applications. The passing of a message always begins with the client. The server is unable to send messages without being called (the client will have to poll the server). This document shall be accepted as the standard SPJC.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1. Message from a client

A message from a client is an HTTP `SPJC` request, with a JSON body.

1. The request method SHOULD be `SPJC` (see §3 below), but MAY be `POST`
   1. The request method MUST NOT be anything other than `SPJC` or `POST`
2. The body MUST be JSON format
3. The request path SHOULD NOT contain any IDs
4. This request header SHOULD be sent: `Content-Type: application/json`
5. This request header SHOULD be sent: `Cache-Control: no-cache`
6. The server MUST ignore all request headers, except where impractical*, except
   1. The server MAY consider the value of the Authorization header

This document does not impose any limits to the length/size of the request, although keep in mind that servers MAY impose their own limits.

*where impractical: for example, infrastructure headers, like Host

2. Message from a server

A message from a server is an HTTP response, with a JSON body.

1. The HTTP status code SHOULD be 200
2. The HTTP status code MUST be ignored by the client, except
   1. Iff the message from the client was sent via an `SPJC` request, the status code 501 MAY be considered for deciding to repeat the message via a `POST` request
3. The body MUST be JSON format
4. This response header SHOULD be sent: `Content-Type: application/json`
5. This response header SHOULD be sent: `Cache-Control: max-age=0, no-cache, must-revalidate, proxy-revalidate` or equivalent-in-spirit
6. The body (after parsing), SHOULD be an object

1. The object SHOULD have a `success` key/value
   1. The value of `success` SHOULD be a boolean indicating whether or not something went wrong when generating the response
2. When `success` is false, the object SHOULD have a `message` key/value
   1. The value of `message` SHOULD be a string.

3. SPJC Method

SPJC is a non-standard HTTP method. See also RFC 9110 § 16.1. Method Extensibility

- SPJC method is not “safe”
- SPJC method is not “idempotent”
- Responses to SPJC method are not cachable

Example Code

Client helper function

```
/**
 * Send data to the server, and get its response
 *
 * @param path string
 * @param data any
 * @return object|false
 */
async function spjc(path, data) {
    try {
        let response = await fetch(path, {
            method: 'SPJC',
            headers: {
                'Content-Type': 'application/json',
                'Cache-Control': 'no-cache',
            },
            body: JSON.stringify(data),
            redirect: 'manual', // ignore redirections
        });
        let body = await response.json();
        if (typeof body != 'object') return false;
        if (typeof body.success != 'boolean') return false;
        if (body.success != true) return false;
        return body;
    } catch (e) {
        console.error(e);
        return false;
    }
}
```