

CoRE Working Group  
Internet-Draft  
Intended status: Informational  
Expires: February 23, 2018

P. van der Stok  
consultant  
K. Hartke  
Universitaet Bremen TZI  
August 22, 2017

The 'Pending' Response Code for the Constrained Application Protocol  
(CoAP)  
draft-hartke-core-pending-01

#### Abstract

This document proposes a new CoAP response code, 2.\_\_ Pending. A CoAP server can use this response code to signal that it has accepted the request but has not yet started processing it or that processing the request will take longer than a client is typically willing to wait for a response. A 2.\_\_ response can include status information and indicate a location where the result will become available.

#### Note

The string "2.\_\_" is a placeholder for the CoAP response code that will be assigned by IANA on completion of this document.

#### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 23, 2018.

#### Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1.](#) Introduction . . . . . [2](#)
- [1.1.](#) Terminology . . . . . [3](#)
- [2.](#) 2.\_\_ Pending . . . . . [3](#)
- [2.1.](#) Observing Resources . . . . . [4](#)
- [3.](#) Security Considerations . . . . . [4](#)
- [4.](#) IANA Considerations . . . . . [5](#)
- [5.](#) References . . . . . [5](#)
- [5.1.](#) Normative References . . . . . [5](#)
- [5.2.](#) Informative References . . . . . [6](#)
- Authors' Addresses . . . . . [6](#)

[1.](#) Introduction

The Constrained Application Protocol (CoAP) [[RFC7252](#)] is a request/response protocol not unlike HTTP. CoAP defines no upper bound for the time between a request and the resulting response. For example, a CoAP-over-UDP server is expected to return an empty Acknowledgement to the client if it cannot provide a response right away, but there is no limit on the time when the server should return the Separate Response.

In particular in the case of requests with long processing times, a CoAP client faces the problem that it cannot easily determine how long it should wait for the response and whether the CoAP server is actually still processing the request. Long processing times occur, for example, when requests need manual intervention to authorize their processing, or when they perform a long sequence of remote actions. An example for this is the "possibly long" authorization request specified in EST-coaps [[I-D.vanderstok-ace-coap-est](#)].

This document proposes a new CoAP response code, 2.\_\_ Pending. The semantics of this response code are modelled after the HTTP [[RFC7231](#)] 202 (Accepted) status code:

The 202 (Accepted) status code indicates that the request has been accepted for processing, but the processing has not been

completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place. [...] The representation sent with this response ought to describe the request's current status and point to (or embed) a status monitor that can provide the user with an estimate of when the request will be fulfilled.

The 2.\_\_ (Pending) response code adapts this status to CoAP. The 2.\_\_ (Pending) response code is not meant for overload cases, which are better handled by the 5.03 (Service Unavailable) response code.

### 1.1. Terminology

Readers are expected to be familiar with the terms and concepts described in [[RFC7252](#)] and [[RFC7641](#)].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

## 2. 2.\_\_ Pending

A 2.\_\_ (Pending) response in reply to a GET request indicates that the target resource exists but a representation of the resource is not available yet. The Max-Age Option indicates after what time a client should retry its GET request to retrieve the representation. The client MAY observe the resource as defined in [[RFC7641](#)] to be notified when the representation becomes available (see [Section 2.1](#)).

A 2.\_\_ (Pending) response in reply to a POST request indicates that the result of processing the request is not available yet, for example, because the server needs more time to process the request than a client is typically willing to wait for a response. The server MAY specify a location using the Location-\* options where the result will become available. If the server does not specify a location, the result will become available at the target resource of the POST request. To receive the result, the client MAY poll or observe the resource at the specified location using the GET request method. The Max-Age Option indicates how long the client should wait before making the GET request.

A 2.\_\_ (Pending) response MAY contain a payload that represents the progress of processing the original request or any other status information. The content format of this representation is specified by the Content-Format Option.

A 2.\_\_ (Pending) response is cacheable, but cannot be validated. If it contains Location-\* options, it invalidates any cached response for the resource at the specified location; otherwise, it invalidates any cached response for the target resource of the request.

As a consequence of being cacheable, a 2.\_\_ (Pending) response in reply to a POST request makes the POST method temporarily idempotent: until Max-Age expires, any POST request with the same cache-key -- be it from the same client or any another client -- can yield the same 2.\_\_ (Pending) response. (This is the same behavior as for 4.xx and 5.xx error responses in reply to POST requests.)

2.1. Observing Resources

When a client registers to observe a resource [RFC7641] for which no representation is available yet, the server MAY send one or more 2.\_\_ (Pending) notifications before sending the first 2.05 (Content) or 2.03 (Valid) notification. The possible resulting sequence of notifications is shown in Figure 1.

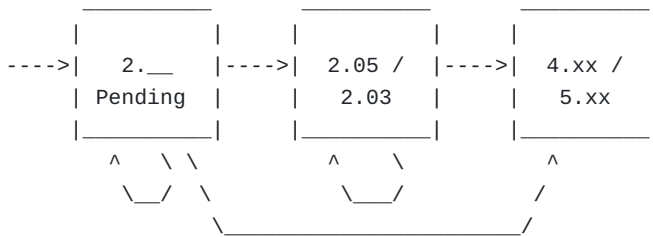


Figure 1: Sequence of Notifications

Unless the server is unwilling to add the client to the list of observers, each 2.\_\_ (Pending) notification MUST include an Observe Option with a sequence number as specified in [RFC7641]. Otherwise, the registration request falls back to a normal GET request.

3. Security Considerations

This section analyses the possible threats related to 2.\_\_ (Pending) responses. It is meant to inform protocol and application developers about the security limitations of the response code as described in this document.

A 2.\_\_ (Pending) response is subject to the same general security considerations as all CoAP responses as described in Section 11 of [RFC7252]. Specifically, the security considerations for the response code are closest to those of the Observe Option as stated in

[Section 7 of \[RFC7641\]](#), because the server stores additional state over an extended period.

2.\_\_ (Pending) responses are secured following the recommendations for the existing CoAP response codes as specified in [Section 9 of \[RFC7252\]](#). When additional security techniques are standardized for CoAP (e.g., based on object security), these are then also available for securing the responses.

#### 4. IANA Considerations

This document adds the 2.\_\_ (Pending) response code to the "CoAP Response Codes" registry.

```

+-----+-----+-----+
| Code | Description | Reference |
+-----+-----+-----+
| 2.__ | Pending      | [RFCXXXX] |
+-----+-----+-----+

```

Table 1: New CoAP Response Codes

[[IANA: Please assign a code point in the range 2.06-2.30.]] [[RFC Editor: Please replace every occurrence of "2.\_\_" in this document with the assigned code point and remove this paragraph before publication.]]

#### 5. References

##### 5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), DOI 10.17487/RFC7252, June 2014, <<http://www.rfc-editor.org/info/rfc7252>>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", [RFC 7641](#), DOI 10.17487/RFC7641, September 2015, <<http://www.rfc-editor.org/info/rfc7641>>.

## [5.2.](#) Informative References

[I-D.vanderstok-ace-coap-est]

Kumar, S., Stok, P., Kampanakis, P., Furuhed, M., and S. Raza, "EST over secure CoAP (EST-coaps)", [draft-vanderstok-ace-coap-est-02](#) (work in progress), June 2017.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [RFC 7231](#), DOI 10.17487/RFC7231, June 2014, <<http://www.rfc-editor.org/info/rfc7231>>.

### Authors' Addresses

Peter van der Stok  
consultant

Phone: +31-492474673 (Netherlands), +33-966015248 (France)  
Email: [consultancy@vanderstok.org](mailto:consultancy@vanderstok.org)  
URI: [www.vanderstok.org](http://www.vanderstok.org)

Klaus Hartke  
Universitaet Bremen TZI  
Postfach 330440  
Bremen D-28359  
Germany

Phone: +49-421-218-63905  
Email: [hartke@tzi.org](mailto:hartke@tzi.org)