

CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 14 January 2021

T. Fossati
arm
J. Jiménez
K. Hartke
Ericsson
13 July 2020

Problem Details For CoAP APIs
draft-ietf-core-problem-details-01

Abstract

This document defines a "problem detail" as a way to carry machine-readable details of errors in a CoAP response to avoid the need to define new error response formats for CoAP APIs. The proposed format is inspired by the Problem Details for HTTP APIs defined in RFC 7807.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at <https://github.com/core-wg/core-problem-details> (<https://github.com/core-wg/core-problem-details>).

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 <https://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 14 January 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://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. Requirements Language	3
2. Basic Problem Details	3
2.1. Examples	4
3. Additional Problem Details	5
3.1. Examples	5
4. Security Considerations	5
5. IANA Considerations	6
6. References	6
6.1. Normative References	6
6.2. Informative References	6
Acknowledgments	6
Authors' Addresses	6

1. Introduction

CoAP [RFC7252] response codes are sometimes not sufficient to convey enough information about an error to be helpful. This specification defines a simple and extensible CoRAL [I-D.ietf-core-coral] vocabulary to suit this purpose. It is designed to be reused by CoAP APIs, which can identify distinct "problem types" specific to their needs. Thus, API clients can be informed of both the high-level error class (using the response code) and the finer-grained details of the problem (using this vocabulary), as shown in Figure 1.

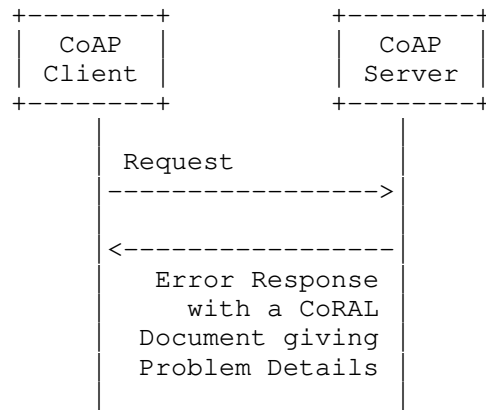


Figure 1: Problem Details

The vocabulary presented is largely inspired by the Problem Details for HTTP APIs defined in [RFC7807].

1.1. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Basic Problem Details

A CoAP Problem Details is a CoRAL document with the following elements:

- * "type" (id) - The problem type. This is a mandatory element.
- * "title" (text) - A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem.
- * "detail" (text) - A human-readable explanation specific to this occurrence of the problem.
- * "instance" (uri) - A URI reference that identifies the specific occurrence of the problem. It may or may not yield further information if dereferenced.

Consumers MUST use "type" as primary identifiers for the problem type; the "title" string is advisory and included only for consumers who are not aware of the semantics of the "type" value.

The "detail" member, if present, ought to focus on helping the client correct the problem, rather than giving debugging information. Consumers SHOULD NOT parse the "detail" member for information; extensions (see Section 3) are more suitable and less error-prone ways to obtain such information.

Note that the "instance" URI reference may be relative; this means that it must be resolved relative to the document's base URI, as per [I-D.ietf-core-coral].

2.1. Examples

This section presents a series of examples of the basic vocabulary in CoRAL textual format (Section 4 of [I-D.ietf-core-coral]). The examples are fictitious. No identification with actual products is intended or should be inferred. All examples involve the same CoAP problem type with semantics of "unknown key id", defined in the fictitious namespace "http://vocabulary.private-api.example".

Note that CoRAL documents are exchanged in CoRAL binary format (Section 3 of [I-D.ietf-core-coral]) in practice. This includes the use of [I-D.ietf-core-href] as an alternative to URIs that is optimized for constrained nodes.

The example in Figure 2 has the most compact representation. It avoids any non-mandatory element. This is suitable for a constrained receiver that happens to have precise knowledge of the semantics associated with the "type".

```
#using pd = <http://example.org/vocabulary/problem-details#>
#using ex = <http://vocabulary.private-api.example/#>
```

```
pd:type          ex:unknown-key-id
```

Figure 2: Minimalist

The example in Figure 3 has all the mandatory as well as the optional elements populated. This format is appropriate for a less constrained receiver (for example, an edge gateway forwarding to a log server that needs to gather as much contextual information as possible, including the problem "headline", details about the error condition, and an error-specific instance URL).

```
#using pd = <http://example.org/vocabulary/problem-details#>
#using ex = <http://vocabulary.private-api.example/#>

pd:type          ex:unknown-key-id
pd:title         "unknown key id"
pd:detail        "Key with id 0x01020304 not registered"
pd:instance      <https://private-api.example/errors/5>
```

Figure 3: Full-Fledged

3. Additional Problem Details

Problem type definitions MAY extend the Problem Details document with additional elements to convey additional, problem-specific information.

Clients consuming problem details MUST ignore any such elements that they do not recognize; this allows problem types to evolve and include additional information in the future.

3.1. Examples

The example in Figure 4 has all the basic elements as well as an additional, type-specific element.

```
#using pd = <http://example.org/vocabulary/problem-details#>
#using ex = <http://vocabulary.private-api.example/#>

pd:type          ex:unknown-key-id
pd:title         "unknown key id"
pd:detail        "Key with id 0x01020304 not registered"
pd:instance      <https://private-api.example/errors/5>
ex:key-id        0x01020304
```

Figure 4: Full Payload and Extensions

4. Security Considerations

Problem Details for CoAP APIs are serialized in the CoRAL binary format. See Section 11 of [RFC7252] for security considerations relating to CoAP. See Section 7 of [I-D.ietf-core-coral] for security considerations relating to CoRAL.

The security and privacy considerations outlined in Section 5 of [RFC7807] apply in full.

5. IANA Considerations

TODO.

6. References

6.1. Normative References

- [I-D.ietf-core-coral] Hartke, K., "The Constrained RESTful Application Language (CoRAL)", Work in Progress, Internet-Draft, draft-ietf-core-coral-03, 9 March 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-core-coral-03.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://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, <<https://www.rfc-editor.org/info/rfc7252>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

6.2. Informative References

- [I-D.ietf-core-href] Hartke, K., "Constrained Resource Identifiers", Work in Progress, Internet-Draft, draft-ietf-core-href-03, 9 March 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-core-href-03.txt>>.
- [RFC7807] Nottingham, M. and E. Wilde, "Problem Details for HTTP APIs", RFC 7807, DOI 10.17487/RFC7807, March 2016, <<https://www.rfc-editor.org/info/rfc7807>>.

Acknowledgments

Mark Nottingham and Erik Wilde, authors of RFC 7807. Carsten Bormann, Jim Schaad, Christian Amsuess for review and comments on this document.

Authors' Addresses

Thomas Fossati
arm

Email: thomas.fossati@arm.com

Jaime Jiménez
Ericsson

Email: jaime@iki.fi

Klaus Hartke
Ericsson

Email: klaus.hartke@ericsson.com