Draft
OCCI-WG

Ralf Nyrén
Florian Feldhaus, GWDG

February 25, 2011
Updated: April 13, 2015

Open Cloud Computing Interface - JSON Rendering

Status of this Document

This document is a draft providing information to the community regarding the specification of the Open Cloud Computing Interface.

Abstract

This document, part of a document series, produced by the OCCI working group within the Open Grid Forum (OGF), provides a high-level definition of a Protocol and API. The document is based upon previously gathered requirements and focuses on the scope of important capabilities required to support modern service offerings.

Contents

1 Introduction **3**

2 Notational Conventions **3**

3 OCCI JSON Rendering **4**

3.1 Entity Instance Rendering 4

3.1.1 Resource Instance Rendering 4

3.1.2 Link Instance Rendering 5

3.2 Category Instance Rendering 5

3.2.1 Kind Instance Rendering 5

3.2.2 Mixin Instance Rendering 6

3.2.3 Action Instance Rendering 6

3.3 Entity Collection Rendering 7

3.3.1 Resource Collection Rendering 7

3.3.2 Link Collection Rendering 7

3.4 Category Collection Rendering 7

3.4.1 Kind Collection Rendering 7

3.4.2 Mixin Collection Rendering 7

3.4.3 Action Collection Rendering 8

3.5 Attributes Rendering 8

3.5.1 Attribute Description Rendering 8

4 Security Considerations **8**

5 Glossary **9**

6 Contributors **9**

7 Intellectual Property Statement **10**

8 Disclaimer **10**

9 Full Copyright Notice **10**

1 Introduction

The Open Cloud Computing Interface (OCCI) is a RESTful Protocol and API for all kinds of management tasks. OCCI was originally initiated to create a remote management API for IaaS[1] model-based services, allowing for the development of interoperable tools for common tasks including deployment, autonomic scaling and monitoring. It has since evolved into a flexible API with a strong focus on interoperability while still offering a high degree of extensibility. The current release of the Open Cloud Computing Interface is suitable to serve many other models in addition to IaaS, including PaaS and SaaS.

In order to be modular and extensible the current OCCI specification is released as a suite of complimentary documents, which together form the complete specification. The documents are divided into four categories consisting of the OCCI Core, the OCCI Protocols, the OCCI Renderings and the OCCI Extensions.

- The OCCI Core specification consists of a single document defining the OCCI Core Model. The OCCI Core Model can be interacted through *renderings* (including associated behaviours) and expanded through *extensions*.
- The OCCI Protocol specifications consist of multiple documents each describing how the model can be interacted with over a particular protocol (e.g. HTTP, AMQP etc.). Multiple protocols can interact with the same instance of the OCCI Core Model.
- The OCCI Rendering specifications consist of multiple documents each describing a particular rendering of the OCCI Core Model. Multiple renderings can interact with the same instance of the OCCI Core Model and will automatically support any additions to the model which follow the extension rules defined in OCCI Core.
- The OCCI Extension specifications consist of multiple documents each describing a particular extension of the OCCI Core Model. The extension documents describe additions to the OCCI Core Model defined within the OCCI specification suite.

The current specification consists of seven documents. This specification describes version 1.2 of OCCI and is backward compatible with 1.1. Future releases of OCCI may include additional protocol, rendering and extension specifications. The specifications to be implemented (MUST, SHOULD, MAY) are detailed in the table below.

Table 1. What OCCI specifications must be implemented for the specific version.

Document	OCCI 1.1	OCCI 1.2
Core Model	MUST	MUST
Infrastructure Model	SHOULD	SHOULD
Platform Model	MAY	MAY
SLA Model	MAY	MAY
HTTP Protocol	MUST	MUST
Text Rendering	MUST	MUST
JSON Rendering	MAY	MUST

2 Notational Conventions

All these parts and the information within are mandatory for implementors (unless otherwise specified). 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].

[1] Infrastructure as a Service

3 OCCI JSON Rendering

The OCCI JSON Rendering specifies a rendering of OCCI instance types in the JSON data interchange format as defined in [2].

The rendering can be used to render OCCI instances independently of the protocol being used. Thus messages can be delivered by e.g. the HTTP protocol as specified in [3].

The following media-type MUST be used for the OCCI JSON Rendering:

`application/occi+json`

The OCCI JSON Rendering consists of a JSON object containing information on the OCCI Core instances OCCI Kind, OCCI Mixin, OCCI Action, OCCI Link and OCCI Resource. The rendering also include a JSON object to invoke the operation identified by OCCI Actions. The rendering of each OCCI Core instance will be described in the following sections.

3.1 Entity Instance Rendering

Entity instances MUST be rendered as JSON hashmaps.

3.1.1 Resource Instance Rendering

The OCCI Resource Instance Rendering consists of a JSON object as shown in the following example. Section **??** contains a detailed example. Table 2 defines the object members.

```
{
    "kind": String,
    "mixins": Array,
    "attributes": Object,
    "actions": Array,
    "id": String,
    "links": Array,
    "summary": String,
    "title": String,
}
```

Table 2. OCCI Resource instance rendered with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
kind	String	Type identifier	immutable	1
mixins	Array of Strings	List of type identifiers of associated OCCI Mixins	mutable	0..*
attributes	Object	Instance Attributes (see 3.5.1)	mutable	0..*
actions	Array of Strings	List of type identifiers of OCCI Actions applicable to the OCCI Resource instance	mutable	0..*
id	String	ID of the OCCI Resource	immutable	1
links	Array of Strings	List of URIs of OCCI Links	mutable	0..*
summary	String	Summary text of resource	mutable	0..1
title	String	Title of resource	mutable	0..1

3.1.1.1 Action Invocation Rendering The OCCI Action Invocation Rendering identifies an invocable operation on a OCCI Resource or OCCI Link instance. To trigger such an operation the OCCI Action Invocation Rendering is required.

The OCCI Action Invocation Rendering consists of a top-level JSON object as shown in the following example. Section **??** contains a detailed example. Table 3 defines the object members.

```
{
    "action": String,
    "attributes": Object
}
```

Table 3. An OCCI Action invocation is rendered with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
action	String	Type identifier	immutable	1
attributes	Object	Instance attributes (see 3.5.1)	mutable	0..*

3.1.2 Link Instance Rendering

The OCCI Link Instance Rendering consists of a JSON object as shown in the following example. Section **??** contains a detailed example. Table 4 defines the object members.

```
{
    "kind": String,
    "mixins": Array,
    "attributes": Object,
    "actions": Array,
    "id": String,
    "source": String,
    "target": String,
    "rel": String,
    "title": String
}
```

Table 4. OCCI Link instances are rendered with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
kind	String	Type identifier	immutable	1
mixins	Array of Strings	List of type identifiers of associated OCCI Mixins	mutable	0..*
attributes	Object	Instance attributes (see 3.5.1)	mutable	0..*
actions	Array of Strings	List of type identifiers of OCCI Action Categories applicable to the OCCI Link instance	mutable	0..*
id	String	ID of the OCCI Link	immutable	1
source	String	URI of the source OCCI Resource. If only one OCCI Resource is rendered in the same collection, this OCCI Resource is the source of the OCCI Link if this entry is omitted	immutable	0..1
target	String	URI of the target Resource	immutable	1
rel	string	Type identifier of the target Resource, to be supplied if the target is an OCCI Resource.	immutable	0..1
title	String	title of the Link	mutable	0..1

3.2 Category Instance Rendering

Category instances MUST be rendered as JSON hashmaps.

3.2.1 Kind Instance Rendering

The OCCI Kind Instance Rendering consists of a JSON object as shown in the following example. Section **??** contains a detailed example. Table 5 defines the top-level object members.

```
{
    "term": String,
    "scheme": String,
    "title": String,
    "attributes": Object,
    "actions": Array,
    "parent": Array,
    "location": String
}
```

Table 5. OCCI Kind instances are rendered with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
term	String	Unique identifier within the categorisation scheme	immutable	1
scheme	String	Categorisation scheme	immutable	1
title	String	Title of the OCCI Kind	immutable	0..1
attributes	Object	Attribute description, see 8	immutable	0..*
parent	String	OCCI Kind type identifier of the related "parent" Kind instance	immutable	0..1
actions	Array of Strings	List of OCCI Action type identifiers	immutable	0..*
location	string	Transport protocol specific URI bound to the OCCI Kind instance. MUST be supplied for the OCCI Kinds of all OCCI Entities except OCCI Entity itself	immutable	0..1

3.2.2 Mixin Instance Rendering

The OCCI Mixin Instance Rendering consists of a JSON object as shown in the following example. Section **??** contains a detailed example. Table 6 defines the top-level object members.

```
{
    "term": String,
    "scheme": String,
    "title": String,
    "attributes": Object,
    "actions": Array,
    "depends": Array,
    "applies": Array,
    "location": String
}
```

Table 6. OCCI Mixin instances are rendered with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
term	String	Unique identifier within the categorisation scheme	immutable	1
scheme	String	Categorisation scheme	immutable	1
title	String	Title of the OCCI Mixin	immutable	0..1
attributes	Object	Attribute description, see 8	immutable	0..*
depends	Array of Strings	List of type identifiers of the dependent Mixin instances	immutable	0..*
applies	Array of Strings	List of OCCI Kind type identifiers this OCCI Mixin can be applied to		
actions	Array of Strings	List of OCCI Action type identifiers	immutable	0..*
location	String	Transport protocol specific URI bound to the OCCI Mixin instance	immutable	1

3.2.3 Action Instance Rendering

The OCCI Action Instance Rendering consists of a JSON object as shown in the following example. Table 7 defines the top-level object members.

Table 7. OCCI Actions are rendered inside the top-level JSON object with name *actions* as an array of JSON Objects with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
term	String	Unique type identifier within the categorisation scheme	immutable	1
scheme	String	Categorisation scheme	immutable	1
title	String	Title of the OCCI Action	immutable	0..1
attributes	Object	Attribute description, see 8	immutable	0..*

```
        {
            "term": String,
            "scheme": String,
            "title": String,
            "attributes": Object,
        }
```

3.3 Entity Collection Rendering

Collections of Entity instances MUST be rendered as JSON arrays. The content of that array is a set of entity instance renderings.

That array MUST be a member of a JSON hashmap that is associated with the relevant key name specific to the type of Entity collection being rendered.

3.3.1 Resource Collection Rendering

The JSON hashmap key-name associated with the array of resource instances MUST be resources.

```
{
    "resources": []
}
```

3.3.2 Link Collection Rendering

The JSON hashmap key-name associated with the array of link instances MUST be links.

```
{
    "links": []
}
```

3.4 Category Collection Rendering

Collections of Category instances MUST be rendered as JSON arrays. The content of that array is a set of Category instance renderings.

That array MUST be a member of a JSON hashmap that is associated with the relevant key name specific to the type of Category collection being rendered.

3.4.1 Kind Collection Rendering

The JSON hashmap key-name associated with the array of kind instances MUST be kinds.

```
{
    "kinds": []
}
```

3.4.2 Mixin Collection Rendering

The JSON hashmap key-name associated with the array of mixin instances MUST be mixins.

```
{
    "mixins": []
}
```

3.4.3 Action Collection Rendering

The JSON hashmap key-name associated with the array of action instances MUST be actions.

```
{
    "actions": []
}
```

Collections of Category instances are rendered as JSON arrays.

3.5 Attributes Rendering

Attribute names consist of alphanumeric characters separated by dots. The dots define a namespace hierarchy. This hierarchy is reflected by stacked JSON objects as shown in the following example. The last object contains either a Number, String or Boolean value or, when used within a category, an Object following the Attribute Description Rendering (see 3.5.1).

```
{
    "one": {
        "two": {
            "three": Number|String|Boolean|Object
        }
    }
}
```

3.5.1 Attribute Description Rendering

Attribute Descriptions are rendered as JSON objects as defined in table 8

Table 8. All properties of the Attribute definition are optional, but may contain defaults which MUST be used if the Attribute is not present in the instantiated OCCI Entity.

Object member	JSON type	Description	Default
mutable	Boolean	Defines if the Attribute is mutable after initialization	false
required	Boolean	Defines if the Attribute MUST be specified at instantiation of the OCCI Entity	false
type	String	Type of the Attribute. MUST be either string, number or boolean.	string
default	String, Number or Boolean	Attribute default. MUST be the same type as defined in the type property and MUST be used if the Attribute is not present in the instantiated OCCI Entity	
description	String	Description of the attribute	

```
{
    "mutable": Boolean,
    "required": Boolean,
    "type": String,
    "default": String | Number | Boolean,
    "description": String
}
```

4 Security Considerations

OCCI does not require that an authentication mechanism be used nor does it require that client to service communications are secured. It does RECOMMEND that an authentication mechanism be used and that where appropriate, communications are encrypted using HTTP over TLS. The authentication mechanisms that MAY be used with OCCI are those that can be used with HTTP and TLS. For further discussion see the appropiate section in [3].

5 Glossary

Term	Description
Action	An OCCI base type. Represents an invocable operation on a Entity sub-type instance or collection thereof.
Attribute	A type in the OCCI Core Model. Describes the name and properties of attributes found in Entity types.
Category	A type in the OCCI Core Model and the basis of the OCCI type identification mechanism. The parent type of Kind.
capabilities	In the context of Entity sub-types **capabilities** refer to the Attributes and Actions exposed by an **entity instance**.
Collection	A set of Entity sub-type instances all associated to a particular Kind or Mixin instance.
Entity	An OCCI base type. The parent type of Resource and Link.
entity instance	An instance of a sub-type of Entity but not an instance of the Entity type itself. The OCCI model defines two sub-types of Entity, the Resource type and the Link type. However, the term *entity instance* is defined to include any instance of a sub-type of Resource or Link as well.
Kind	A type in the OCCI Core Model. A core component of the OCCI classification system.
Link	An OCCI base type. A Link instance associates one Resource instance with another.
Mixin	A type in the OCCI Core Model. A core component of the OCCI classification system.
mix-in	An instance of the Mixin type associated with an *entity instance*. The "mix-in" concept as used by OCCI *only* applies to instances, never to Entity types.
OCCI	Open Cloud Computing Interface.
OGF	Open Grid Forum.
Resource	An OCCI base type. The parent type for all domain-specific Resource sub-types.
resource instance	See *entity instance*. This term is considered obsolete.
tag	A Mixin instance with no attributes or actions defined. Used for taxonomic organisation of entity instances
template	A Mixin instance which if associated at instance creation-time pre-populate certain attributes.
type	One of the types defined by the OCCI Core Model. The Core Model types are Category, Attribute, Kind, Mixin, Action, Entity, Resource and Link.
concrete type/sub-type	A concrete type/sub-type is a type that can be instantiated.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
URN	Uniform Resource Name.

6 Contributors

We would like to thank the following people who contributed to this document:

Name	Affiliation	Contact
Michael Behrens	R2AD	behrens.cloud at r2ad.com
Mark Carlson	Toshiba	mark at carlson.net
Augusto Ciuffoletti	University of Pisa	augusto.ciuffoletti at gmail.com
Andy Edmonds	ICCLab, ZHAW	edmo at zhaw.ch
Sam Johnston	Google	samj at samj.net
Gary Mazzaferro	Independent	garymazzaferro at gmail.com
Thijs Metsch	Intel	thijs.metsch at intel.com
Ralf Nyrén	Independent	ralf at nyren.net
Alexander Papaspyrou	Adesso	alexander at papaspyrou.name
Boris Parák	CESNET	parak at cesnet.cz
Alexis Richardson	Weaveworks	alexis.richardson at gmail.com
Shlomo Swidler	Orchestratus	shlomo.swidler at orchestratus.com
Florian Feldhaus	NetApp	florian.feldhaus at gmail.com

Next to these individual contributions we value the contributions from the OCCI working group.

7 Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

8 Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

9 Full Copyright Notice

References

[1] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119 (Best Current Practice), Internet Engineering Task Force, Mar. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2119.txt

[2] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," RFC 4627 (Informational), Internet Engineering Task Force, Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4627.txt

[3] R. Nyren, T. Metsch, and A. Edmonds, "Open Cloud Computing Interface – HTTP Protocol," Draft, March 2015. [Online]. Available: TBD