GWD-R
OCCI-WG

Ralf Nyrén
Florian Feldhaus, GWDG
February 25, 2011
Updated: February 7, 2013

Open Cloud Computing Interface - JSON Rendering

Status of this Document

This document provides information to the community regarding the specification of the Open Cloud Computing Interface. Distribution is unlimited.

Copyright Notice

Trademarks

OCCI is a trademark of the Open Grid Forum.

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.

Comments

Contents

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 three categories consisting of the OCCI Core, 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 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. They do not require changes to the HTTP Rendering specifications as of this version of the specification.

The current specification consists of three documents. This specification describes version 1.1 of OCCI. Future releases of OCCI may include additional rendering and extension specifications. The documents of the current OCCI specification suite are:

OCCI Core describes the formal definition of the the OCCI Core Model [1].

OCCI HTTP Rendering defines how to interact with the OCCI Core Model using the RESTful OCCI API [2]. The document defines how the OCCI Core Model can be communicated and thus serialised using the HTTP protocol.

OCCI Infrastructure contains the definition of the OCCI Infrastructure extension for the IaaS domain [3]. The document extends the OCCI Core Model with additional Entity sub-types and their associated attributes and actions.

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 [4].

3 OCCI JSON Rendering

The OCCI JSON Rendering specifies a rendering of OCCI instance types in the JSON data interchange format as defined in [5].

[1] Infrastructure as a Service

The Rendering can be used to render OCCI instances independently of the transport mechanism being used. Thus messages can be delivered by e.g. the HTTP protocol as specified in [2] or by using text files with the .json file extension as defined in [5].

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

```
application/occi+json
```

4 JSON Format

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.

4.1 Attributes

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 examle. The last object contains either a Number, String or Boolean value or, when used within a category, an Object following the Attribute Description Format (see 4.8).

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

4.2 Resource Instance Format

The OCCI Resource Instance Format consists of a JSON object as shown in the following example. Section 5.1 contains a detailed example. Table 1 defines the object members.

```
{
    "resources": [
        {
            "kind": String,
            "mixins": Array,
            "attributes": Object,
            "actions": Array,
            "id": String,
            "links": Array
        }
    ]
}
```

Table 1. OCCI Resource instances are rendered inside the top-level JSON object with name *resources* as an array of JSON objects 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 4.8)	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..*

4.3 Link Instance Format

The OCCI Link Instance Format consists of a JSON object as shown in the following example. Section 5.2 contains a detailed example. Table 2 defines the object members.

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

Table 2. OCCI Link instances are rendered inside the top-level JSON object with name *links* as an array of JSON objects 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 4.8)	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

4.4 Action Invocation Format

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

The OCCI Action Invocation Format consists of a top-level JSON object as shown in the following example. Section 5.3 contains a detailed example. Table 3 defines the object members.

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

Table 3. An OCCI Action invocation is rendered as top-level JSON object with the following entries:

Object member	JSON type	Description	Mutability	Multiplicity
action	String	Type identifier	immutable	1
attributes	Object	Instance attributes (see 4.8)	mutable	0..*

4.5 Kind Instance Format

The OCCI Kind Instance Format consists of a JSON object as shown in the following example. Section 5.4 contains a detailed example. Table 4 defines the top-level object members.

Table 4. OCCI Kind instances are rendered inside the top-level JSON object with name *kinds* as an array of JSON objects 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 7	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

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

4.6 Mixin Instance Format

The OCCI Mixin Instance Format consists of a JSON object as shown in the following example. Section 5.5 contains a detailed example. Table 5 defines the top-level object members.

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

4.7 Action Instance Format

The OCCI Action Instance Format consists of a JSON object as shown in the following example. Table 6 defines the top-level object members.

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

Table 5. OCCI Mixin instances are rendered inside the top-level JSON object with name *mixins* as an array of JSON objects 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 7	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

Table 6. 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 7	immutable	0..*

```
    ]
}
```

4.8 Attribute Description Format

OCCI Attribute Descriptions are rendered as JSON objects as defined in table 7

Table 7. All properties of the OCCI 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
}
```

5 Detailed Examples

5.1 Resource Instance Format Example

```
{
    "resources": [
        {
            "kind": "http://schemas.ogf.org/occi/infrastructure#compute",
            "mixins": [
                "http://example.com/occi/templates#my_os_mixin"
            ],
            "attributes": {
                "occi": {
                    "compute": {
                        "speed": 2,
                        "memory": 4,
                        "cores": 2
                    }
                },
                "com": {
                    "example": {
                        "occi": {
                            "templates": {
                                "my_os_mixin": {
                                    "my_attribute": "my_value"
                                }
                            }
                        }
                    }
                }
            },
            "actions": [
                "http://schemas.ogf.org/occi/infrastructure/compute/action#start"
            ],
            "id": "urn:uuid:996ad860-2a9a-504f-8861-aeafd0b2ae29",
            "links": [
                "/storage/22fe83ae-a20f-54fc-b436-cec85c94c5e8"
            ]
        }
    ]
}
```

5.2 Link Instance Format Example

```
{
    "links": [
        {
            "kind": "http://schemas.ogf.org/occi/infrastructure#networkinterface",
            "mixins": [
                "http://schemas.ogf.org/occi/infrastructure/networkinterface#ipnetworkinterface"
            ],
            "attributes": {
                "occi": {
                    "infrastructure": {
                        "networkinterface": {
                            "interface": "eth0",
                            "mac": "00:80:41:ae:fd:7e",
                            "address": "192.168.0.100",
                            "gateway": "192.168.0.1",
                            "allocation": "dynamic"
                        }
                    }
                }
            },
            "actions": [
                "http://schemas.ogf.org/occi/infrastructure/networkinterface/action#up"
            ],
            "id": "urn:uuid:22fe83ae-a20f-54fc-b436-cec85c94c5e8",
            "target": "/network/b7d55bf4-7057-5113-85c8-141871bf7635",
            "source": "/compute/996ad860-2a9a-504f-8861-aeafd0b2ae29"
        }
    ]
}
```

5.3 Action Invocation Format Example

```
{
    "action": "http://schemas.ogf.org/occi/infrastructure/compute/action#stop",
    "attributes": {
        "method": "graceful"
    }
}
```

5.4 Kind Format Example

```
{
    "kinds": [
        {
            "term": "compute",
            "scheme": "http://schemas.ogf.org/occi/infrastructure#",
            "title": "Compute Resource",
            "parent": "http://schemas.ogf.org/occi/core#resource",
            "attributes": {
                "occi": {
                    "compute": {
                        "hostname": {
                            "mutable": true,
                            "required": false,
                            "type": "string",
                            "description": "Hostname of the compute resource"
                        },
                        "state": {
                            "mutable": false,
                            "required": false,
                            "type": "string",
                            "default": "inactive",
                            "description": "State the compute resource is in"
                        }
                    }
                }
            },
            "actions": [
                "http://schemas.ogf.org/occi/infrastructure/compute/action#start",
                "http://schemas.ogf.org/occi/infrastructure/compute/action#stop",
                "http://schemas.ogf.org/occi/infrastructure/compute/action#restart",
                "http://schemas.ogf.org/occi/infrastructure/compute/action#suspend"
            ],
            "location": "/compute/"
        }
    ]
}
```

5.5 Mixin Format Example

```
{
    "mixins": [
        {
            "term": "medium",
            "scheme": "http://example.com/template/resource#",
            "depends": [
                "http://schemas.ogf.org/occi/infrastructure#resource_tpl"
            ],
            "applies": [
                "http://schemas.ogf.org/occi/infrastructure#compute"
            ],
            "attributes": {
                "occi": {
                    "compute": {
                        "speed": {
                            "type": "number",
                            "default": 2.8
                        }
```

```
                        }
                    }
                },
                "title": "Medium VM",
                "location": "/template/resource/medium/"
            }
        ]
}
```

5.6 Action Format Example

```
{
    "actions": [
        {
            "term": "stop",
            "scheme": "http://schemas.ogf.org/occi/infrastructure/compute/action#",
            "title": "Stop Compute instance",
            "attributes": {
                "method": {
                    "mutable": true,
                    "required": false,
                    "type": "string",
                    "default": "poweroff"
                }
            }
        }
    ]
}
```

6 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 OCCI Attributes and OCCI Actions exposed by an **entity instance**.
Client	An OCCI client.
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.
model attribute	An internal attribute of a the Core Model which is *not* client discoverable.
OCCI	Open Cloud Computing Interface.
OCCI base type	One of Entity, Resource, Link or Action.
OCCI Action	see Action.
OCCI Attribute	A client discoverable attribute identified by an instance of the Attribute type. Examples are occi.core.title and occi.core.summary.
OCCI Category	see Category.
OCCI Entity	see Entity.
OCCI Kind	see Kind.
OCCI Link	see Link.
OCCI Mixin	see Mixin.
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.
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.

7 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	Oracle	mark.carlson at oracle.com
Andy Edmonds	Intel - SLA@SOI project	andy at edmonds.be
Sam Johnston	Google	samj at samj.net
Gary Mazzaferro	OCCI Counselour - AlloyCloud, Inc.	garymazzaferro at gmail.com
Thijs Metsch	Platform Computing, Sun Microsystems	tmetsch at platform.com
Ralf Nyrén	Aurenav	ralf at nyren.net
Alexander Papaspyrou	TU Dortmund University	alexander.papaspyrou at tu-dortmund.de
Alexis Richardson	RabbitMQ	alexis at rabbitmq.com
Shlomo Swidler	Orchestratus	shlomo.swidler at orchestratus.com
Florian Feldhaus	GWDG	florian.feldhaus at gwdg.de

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

8 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.

9 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.

10 Full Copyright Notice

References

[1] R. Nyrén, A. Edmonds, A. Papaspyrou, and T. Metsch, "Open Cloud Computing Interface – Core," GFD-P-R.183, April 2011. [Online]. Available: http://ogf.org/documents/GFD.183.pdf

[2] T. Metsch and A. Edmonds, "Open Cloud Computing Interface – HTTP Rendering," GFD-P-R.185, April 2011. [Online]. Available: http://ogf.org/documents/GFD.185.pdf

[3] ——, "Open Cloud Computing Interface – Infrastructure," GFD-P-R.184, April 2011. [Online]. Available: http://ogf.org/documents/GFD.184.pdf

[4] 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

[5] 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