OSB Ops API Specification

Summary

OSB is the protocol for provisioning and binding resources to a service. The OSB specification does not define how a resource owner can perform operations on the provisioned resource.

This document's goal is to introduce a protocol for performing operations on provisioned resources.

Operations include:

- backup and restore of databases
- scaling up and down a compute cluster
- purging a queue or bucket
- etc.

In this document, we will use the OSB terminology going forward.

Specification

The specification relies on OpenAPI to specify the operations that can be invoked on a service instance via the service broker.

This specification introduces extension paths to the OSB spec, examples:

- `operation_instances`
- `job_instances`

Executing an Operation

Operation Route

The OSB specifies conventions for the provisioning and binding routes. The binding routes are in the context of a specific service instance being bound to.

We will use the same convention and introduce "operations" path segment for all the service instance operations to reside. However, this scheme does not imply this API is bound to OSB, but compliments it and allows non-OSB services to implement only operations API.

```
/v2/service_instances/{instance_id}/operation_instances/{operation_name}
```

Where the `instance_id` is the unique identifier for the service instance and the `operation_name` is the unique identifier for the operation to be executed against that specific service instance.

Operation Request

Operation Request Body must valid per the "Parameters" section.

Operation Status Codes

When an operation is received, the service should respond with one of the following status codes

Status Code	Description	Response
200 OK	Operation completed successfully. See Response Body.	See **Operation Response**.
202 ACCEPTED	Operation will complete asynchronously.	See **Scheduling a Job**.
400 BAD REQUEST	Operation cannot be executed with provided input.	See **Errors.**
402 UNAUTHORIZED	Operation requires valid Authorization header.	See **Authorization.**
403 FORBIDDEN	Operation not permitted for user.	See **Authorization.**
404 NOT FOUND	Operation not found.	See **Errors.**
405 METHOD NOT ALLOWED	Operation does not support the Request Method.	See **Errors.**

Operation Response

The operation response will yield a JSON object that validates against the "result" schema:

```
https://openservicebrokerapi.org/operations.json#/definitions/result
```

JSON Schema for the "result" schema (as YAML):

```
defintions:
 result:
  description: |
   Represents result of an operation or job.
   The result may reference an external resource via `href` or may
   embed the result and specify the schema via `schema`
  oneOf:
    - description: References an external resource
   type: object
   properties:
    href:
     description: URL to the result of a job as an external resource
     type: string
     format: uri
   additionalProperties: false
   required:
     - href
    - description: Embeds the results an provides a schema for the
results
   type: object
   properties:
    schema:
     description: Unique identifier for the job
     type: string
     format: uri
   additionalProperties: true
   required:
     - schema
```

Example:

```
HTTP/1.1 200 OK

{
 "result": {
  "schema": "/v2/operations#components/usage_results",
  "size": 100,
  "capacity": 1000
 }
}
```

Scheduling Jobs

A Job is a long-running operation. When a service provides jobs, it should also implement scheduling, monitoring and manipulating of a job.

Schedule Job Route

When an operation is long running, it should treat the operation as a job request and provide ability for the caller to monitor the job.

```
POST
/v2/service_instances/{instance_id}/operation_instances/{operation_name}
```

Schedule Job Request

Schedule Job Request Body must valid per the "Parameters" section.

Schedule Job Status Codes

When a service receives a request to create a job, it will response with one of the status codes:

Status Code	Description	Response
202 ACCEPTED	Job was created. See Response body.	See **Schedule Job Response**.
400 BAD REQUEST	Job cannot be executed with provided input.	See **Errors**.
402 UNAUTHORIZED	Job requires valid Authorization header.	See **Authorization**.
403 FORBIDDEN	Job not permitted for user.	See **Authorization**.
404 NOT FOUND	Job not found.	See **Errors**.
405 METHOD NOT ALLOWED	Only `POST` is supported.	See **Errors**.

Schedule Job Response

The monitor job response will yield a JSON object that validates against the "task" schema:

```
https://openservicebrokerapi.org/operations.json#/definitions/task
```

JSON Schema for the "task" schema (as YAML):

```
defintions:
 task:
  description: Represents a job that was accepted and how to check for the state of the job
  type: object
  properties:
   href:
    description: URL to monitor the state of the job
    type: string
    format: url
   id:
    description: Unique identifier for the job
    type: string
```

Example:

```
HTTP/1.1 202 ACCEPTED
Location: /v2/service_instances/{instance_id}/job_instances/{job_id}

{
 "task": {
  "href": "/v2/service_instances/{instance_id}/job_instances/{job_id}",
  "id": "{job_id}"
 }
}
```

Monitoring a Job

When a job is scheduled, it is often important to be able to check the status of the job and get the result when it completes. The scheduling of the job, when successful, will return a URL to the be used for monitoring the job.

This section specifies how the resource at will behave.

Monitor Job Route

Jobs can be monitored at the route:

```
GET /v2/service_instances/{instance_id}/job_instances/{job_id}
```

Monitor Job Status Codes

The response will be the status either of:

Status Code	Description	Response
200 OK	Status is available. See Response Body.	See **Monitor Job Response**.
402 UNAUTHORIZED	Job requires valid Authorization header.	See **Authorization**.
403 FORBIDDEN	Job not permitted for user.	See **Authorization**.
404 NOT FOUND	Job not found.	See **Errors**.
405 METHOD NOT ALLOWED	Only `GET` is supported.	See **Errors**.

Monitor Job Response

The monitor job response will yield a JSON object that validates against the "job" schema:

```
https://openservicebrokerapi.org/operations.json#/definitions/job
```

JSON Schema for the "job" schema (as YAML):

```
defintions:
 job:
  description: Represents a job that was accepted and how to check for
the state of the job
```

```
  type: object
  properties:
   id:
    description: Unique identifier for the job
    type: string
   state:
    description: Stateof the job
    enum:
              - SCHEDULED
              - STARTED
             - COMPLETED
             - ABORTED
   status:
    description: Status of the job
    enum:
      - UNDETERMINED
      - INPROGRESS
      - SUCCESS
      - FAILURE
   operation:
    description: Details about the operation that created the job
    type: object
    properties:
     id:
      description: The ID of the operation
      type: string
     url:
      description: The URL of the operation
      type: string
      format: uri
     parameters:
      description: The parameters of the operation
      $ref:
https://openservicebrokerapi.org/operations.json#/definitions/parameters
   result:
    description: URL to get the result of the job.
    $ref:
https://openservicebrokerapi.org/operations.json#/definitions/result
   errors:
    description: Errors that occurred during the job.
    $ref: #/definitions/errors
   messages:
    description: Messages emitted during the job.
    $ref: #/definitions/messages
   createdBy:
    description: StaffId of user who initiated the job
    type: string
   createdOn:
    description: The date-time when the job was created
    type: string
    format: date-time
   updatedBy:
    description: StaffId of user who updated the job. Will be same as
```

```
createdBy except for the abort/cancel case (???)
    type: string
   updatedOn:
    description: The date-time when the job state was last updated
    type: string
    format: date-time
  required:
    - id
    - state
    - status
    - operation
    - createdBy
    - createdOn
    - updatedBy
    - updatedOn
```

Examples:

Example 1. Job for scaling a compute cluster is scheduled.

```
HTTP/1.1 200 OK

{
 "job": {
  "id": "{job_id}"
  "state": "SCHEDULED",
  "status": "UNDETERMINED",
  "operation": {
   "id": "{operation_name},
   "url":
"/v2/service_instances/{instance_id}/operation_instances/scale",
   "parameters": {
    "min": 2,
    "max": 10,
    "current": 5
   }
  },
  "createdBy": "cstivers",
  "createdOn": "2018-01-01T00:00:00Z",
  "updatedBy": "cstivers",
  "updatedOn": "2018-01-01T00:01:00Z",
 }
}
```

Example 2. Job for scaling a compute cluster is in progress.

```
HTTP/1.1 200 OK

{
 "job": {
  "id": "{job_id}"
  "state": "STARTED",
  "status": "INPROGRESS",
  "operation": {
   "id": "{operation_name},
   "url":
"/v2/service_instances/{instance_id}/operation_instances/scale",
   "parameters": {
    "min": 2,
    "max": 10,
    "current": 5
   }
  },
  "messages": [
   {"message": "Updated ASG configuration"},
   {"message": "Scaling from 1 to 5"}
  ],
  "createdBy": "cstivers",
  "createdOn": "2018-01-01T00:00:00Z",
  "updatedBy": "cstivers",
  "updatedOn": "2018-01-01T00:01:00Z",
 }
}
```

Example 3. Job for scaling a compute cluster has completed successfully.

```
HTTP/1.1 200 OK

{
 "job": {
  "id": "{job_id}"
  "state": "COMPLETED",
  "status": "SUCCESS",
  "operation": {
   "id": "scale",
   "url":
"/v2/service_instances/{instance_id}/operation_instances/scale",
   "parameters": {
    "min": 2,
    "max": 10,
    "current": 5
   }
  },
  "messages": [
   {"message": "Updated ASG configuration", "timestamp":
"2018-01-01T00:00:00Z"},
   {"message": "Scaling from 1 to 5", "timestamp":
"2018-01-01T00:00:30Z"},
   {"message": "Operation completed", "timestamp":
"2018-01-01T00:01:00Z"}
  ],
  "result": {
   "href":
"/v2/service_instances/{instance_id}/job_instances/{job_id}/result"
  },
  "createdBy": "cstivers",
  "createdOn": "2018-01-01T00:00:00Z",
  "updatedBy": "cstivers",
  "updatedOn": "2018-01-01T00:01:00Z"
 }
}
```

Cancelling a Job

Cancel Job Route

```
DELETE /v2/service_instances/{instance_id}/job_instances/{job_id}
```

Monitor Job Status Codes

The response will be the status either of:

Status Code	Description	Response

200 OK	Job has been cancelled.	–none–
402 UNAUTHORIZED	Missing Valid Authorization header.	See **Authorization**.
403 FORBIDDEN	User not authorized to cancel the job.	See **Authorization**.
404 NOT FOUND	Job does not exist.	See **Errors**.
405 METHOD NOT ALLOWED	Only if `DELETE` is not supported, then the Job cannot be cancelled.	See **Errors**.

Discovery

If a Service Broker has operations available for the services it provisions, it will expose an operations endpoint (`/v2/operations`), which will serve an OpenAPI v3.x compliant specification.

Discovery Route

```
GET /v2/operations
```

Discovery Status Codes

Status Code	Description	Response
200 OK	MUST return the operations descriptor for the Service Broker	–none–
404 Not Found	Only if the operations are not provided	–none–

Discover Response

The response body will be an OpenAPI document that specifies each of the exposed operations.

For details on the specification, see https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md

Dicsover Response Example

The following is an example of exposing operations for a database, including:

- backup – initiate a backup job
- backup_job – check the status of a backup job

Specification:

```
openapi: "3.0.0"
info:
  version: 2.13
  title: Database Operations
  contact:
    name: Database Team
    email: database@atlassian.com
servers:
  - description: prod
    url: https://database.us-west-2.prod.atl-paas.net/osb/database/
  - description: staging
    url: https://database.us-west-2.staging.atl-paas.net/osb/database/
  - description: dev
```

```
    url: https://database.us-west-2.dev.atl-paas.net/osb/database/
paths:
  /v2/service_instances/{instance_id}/operation_instances/info:
    get:
      operationName: info
      description: |
        Get details about the database instance.
      parameters:
        - name: instance_id
          description: The database instance to be backed up
          in: path
          required: true
          schema:
            type: string
      responses:
        '200':
          description: Get info about the database instance
          content:
            application/json:
              schema:
                $ref: '#/components/info'
        default:
          description: unexpected error
          content:
            application/json:
              schema:
                $ref:
'https://openservicebrokerapi.org/operations.json#/definitions/errors'
  /v2/service_instances/{instance_id}/operation_instances/backup:
    post:
      operationName: backup
      description: |
        Backup all the data in the database.
  This results in a job being created and scheduled to run.
      parameters:
        - name: instance_id
          description: The database instance to be backed up
          in: path
          required: true
          schema:
            type: string
      responses:
        '202':
          description: Backup request for {instance_id} has be accepted
and is processing
          content:
            application/json:
              schema:
                $ref:
'https://openservicebrokerapi.org/operations.json#/definitions/task'
        default:
          description: unexpected error
          content:
            application/json:
              schema:
                $ref:
'https://openservicebrokerapi.org/operations.json#/definitions/errors'
  /v2/service_instances/{instance_id}/job_instances/{job_id}:
    get:
      operationName: monitor_backup
      description: |
        Monitor the status of a backup job.
      parameters:
        - name: instance_id
          description: Unique identifier for the database
          in: path
          required: true
          schema:
            type: string
        - name: job_id
          description: Unique identifier for the backup job
          in: path
          required: true
          schema:
            type: string
      responses:
        '200':
          description: Backup job exists and status is returned.
          content:
            application/json:
              schema:
                $ref:
'https://openservicebrokerapi.org/operations.json#/definitions/job'
        default:
          description: unexpected error
          content:
            application/json:
              schema:
                $ref:
'https://openservicebrokerapi.org/operations.json#/definitions/errors'
components:
  info:
    type: object
      properties:
        name:
          description: Database name
          type: string
        ipv4:
          description: IPv4 Address
          type: string
        port:
          description: Port
          type: int
        capacity:
          description: Database Capacity Info
          type: object
```

```
properties:
  total:
    description: total storage capacity allocated
    type: number
```

```
            used:
              description: used storage capacity
              type: number
```

Parameters

Operation parameters will be defined as a flat JSON object. The object will be flat, because it allows for easier integration with tools and user interfaces.

The URI for the Parameters schema is:

```
https://openservicebrokerapi.org/operations.json#/definitions/parameters
```

"parameters" JSON Schema (as YAML):

```
definitions:
 parameters:
  description: Represents parameters for an operation.
  type: object
  patternProperties:
   '^[a-zA-Z]([a-zA-Z0-9_\-]?[a-zA-Z0-9]+)*$':
    oneOf:
      - type: boolean
      - type: number
      - type: string
      - type: array
       items:
       oneOf:
         - type: boolean
         - type: number
         - type: string
```

Example parameters for scaling compute resources:

```
{
 "parameters": {
  "scale_min": 2,
  "scale_max": 10,
  "scale_current": 5
 }
}
```

Errors

For ease of integration with tooling and interfaces, a standard Error format will be utilized by operations to communicate the details for the error.

The URI for the Errors schema is:

```
https://openservicebrokerapi.org/operations.json#/definitions/errors
```

"errors" JSON Schema (as YAML):

```
definitions:
 errors:
  description: A collection of errors which were raised for the
operation
  type: array
  items:
   $ref: '#/definitions/error'
 error:
  description: Respresents a message emitted during the operation.
  type: object
  properties:
   error:
    description: An operation specific error identifier
    type: integer
   description:
    description: Description of the error
    type: string
   timestamp:
    description: The timestamp of the error
    type: string
    format: date-time
  required:
   - message
```

Error Example:

```
{
 "errors": [
  {"error": "OpenedTheGatesOffHell", "description": "You unleashed the
beast!"},
  {"error": "FireFireFire", "description": "The house is on fire!"}
 ]
}
```

Messages

For ease of integration with tooling and interfaces, a standard message format will be utilized by operations to communicate the additional information to the end-user.

The URI for the "messages" schema is:

```
https://openservicebrokerapi.org/operations.json#/definitions/messages
```

The JSON Schema for "messages" (as YAML):

```
definitions:
 messages:
  description: A collection of errors which were raised for the
operation
  type: array
  items:
   $ref: '#/definitions/message'
 message:
  description: Respresents a message emitted during the operation.
  type: object
  properties:
   message:
    description: Message that was emitted
    type: string
   timestamp:
    description: The timestamp of the message
    type: string
    format: date-time
  required:
   - message
```

Error Example:

```
{
 "messages": [
  {"message": "You unleashed the beast!"},
  {"message": "The house is on fire!"}
 ]
}
```

Authorization

– TBD –

Related Material

Open Service Broker API (OSB)

> Protocol for how a service broker will be registered, and called upon to provision services and bind provisioned services to other services.

OpenAPI Specification

> The OpenAPI Specification (OAS) defines a standard, programming language-agnostic interface description for REST APIs, which allows both humans and computers to discover and understand the capabilities of a service without requiring access to source code, additional documentation, or inspection of network traffic. When properly defined via OpenAPI, a consumer can understand and interact with the remote service with a minimal amount of implementation logic. Similar to what interface descriptions have done for lower-level programming, the OpenAPI Specification removes guesswork in calling a service.

JSON Schema

JSON Schema is a vocabulary that allows you to annotate and validate JSON documents.