

New Media Type for Oracle REST Services to Support Specialized Resource Types

ORACLE WHITEPAPER | MARCH 2015





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Contents

Introduction	3
Conventions and Terminology	3
Core terminology	3
Singular Resource	4
Collection Resource	8
Exception Detail Resource	13
Status Resource	14
Query Description Resource	15
create-form Resource	16
edit-form Resource	17
JSON Schema	18
IANA Considerations	28
References	28
Change Log	28

Introduction

This document defines a new media type, `application/vnd.oracle.resource+json`, which can be used by REST services to support the specialized resource types defined in the following table.

Resource Type	Description
Singular	Single entity resource, such as an employee or a purchase order. For more information, see “Singular Resource.”
Collection	List of items, such as employees or purchase orders. See “Collection Resource.”
Exception Detail	Detailed information about a failed request. See “Exception Detail Resource.”
Status	Status of a long running job. See “Status Resource.”
Query description	Query syntax description used by client to build the "q" query parameter. See “Query Description Resource.”
create-form	Template used by the client to get default values when creating a new resource. See “create-form Resource.”
edit-form	Template used by the client to determine which properties are updateable when editing a resource. See “edit-form Resource.”

The `application/vnd.oracle.resource+json` media type:

- » Defines a mandatory parameter named `type` that enables you to specify the specialized resource types.
- » Is a subtype of `application/json`.
- » For each specialized resource type, this document defines when and how it should be used, and its relationship to the other resource types.

Application developers can use this new `application/vnd.oracle.resource+json` media type to consume Oracle REST resources. Once the client knows the media type, the client side can derive much more information than from an `application/json` media type.

Conventions and Terminology

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](#).

Core terminology

Resource

The term "resource" used in this specification refers to a [REST](#) resource.

Paging

Paging allows the client to receive a subset of the items contained in a collection instead of receiving the entire collection at one time.

Paging Size

Paging size defines the number of items in collection that are returned in one paging request.

Singular Resource

A singular resource represents a single entity resource, such as an employee or a purchase order.

For a singular resource, set `type=singular` in the `application/vnd.oracle.resource+json` media type.

Properties

A singular resource supports the "links" property described in this section.

links Property

HATEOAS is one of the most important constraints of REST resources. To support this constraint, each singular resource has a "links" property that defines the hyperlinks to related resources.

The "links" property is an array, and each item in the array is a link object.

Each link object must include the following properties:

- » "rel"— Relation to the target resource.
- » "href"— A [URI](#) or [URI Template](#). If the value is set to URI Template, then the "templated" property must be set to `true`.

A link object may include the following optional properties:

- » "templated"— Boolean flag that specifies whether the "href" property is a URI or URI template. If the property is a URI template, set this value to `true`.
- » "mediaType"— Media type, as defined in [RFC 2046](#), describing the link target.
- » "method"— HTTP method for requesting the target of the link.
- » "profile"— Link to the metadata of the resource, such as [JSON-schema](#), that describes the resource expected when dereferencing the target resource.

The following is an example of a purchase order singular resource that defines a "self" link:

```
{
  "orderId": 123,
  "description": "An office supply offer for a new hire",
  "status": "submitted",
  "links": [
    { "rel": "self", "href": "http://example.com/orders/123" }
  ]
}
```

Query Parameters

When requesting a singular resource via an HTTP GET request, you can specify one or more query parameters to refine the content that is returned.

expand Query Parameter

A resource may be organized hierarchically. For example, a purchase order may consist of a header resource that contains multiple "lines". In this example, when a GET request is issued on the header resource, by default, only the header content is returned with links provided in the "lines" section. The following shows an example of the response under the default behavior:

```
{
  "orderId": 123,
  "description": "An office supply offer for a new hire",
  "status": "submitted",
  "lines": {
    "links": [
      {
        "rel": "self",
        "href": "http://example.com/orders/123/lines"
      }
    ]
  },
  "links": [
    { "rel": "self", "href": "http://example.com/orders/123" }
  ]
}
```

The `expand` query parameter allows the client to expand the child resources so that the contents return with the parent in the same GET request. The query parameter value can be set to one of the following:

- » Name of a single child resource.
- » Comma-separated list of child resource names to specify multiple values.
- » Keyword "all" to expand all child resources.

In addition, you can expand a grandchild resource using a dotted annotation. For example:

`?expand=lines,buyer,buyer.location`.

The following shows an example of the response returned from a GET `/orders/123?expand=lines` request.

```
{
  "orderId": 123,
  "description": "An office supply offer for a new hire",
  "status": "submitted",
  "lines": {
    "items": [
      {
        "lineNumber": 1,
        "productId": 234,
        "quantity": 1,
        "price": 23.5,
        "links": [
```

```

        {
          "rel": "self", "href": "http://example.com/orders/123/lines/1"
        }
      ]
    },
    {
      "lineNumber": 2,
      "productId": 1004,
      "quantity": 2,
      "price": 5.30,
      "links": [
        {
          "rel": "self", "href": "http://example.com/orders/123/lines/2"
        }
      ]
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://example.com/orders/123/lines"
    }
  ]
},
"links": [
  {"rel": "self", "href": "http://example.com/orders/123"}
]
}

```

If a purchase order header has two child resources, lines and buyer, then `GET /orders/123?expand=lines,buyer` expands both resources.

Alternatively, you can specify `?expand=all` to display all child resources, which is equivalent to specifying `?expand=lines,buyer` in this example.

fields Query Parameter

The `fields` query parameter allows the client to select which properties are returned in a GET request. The query parameter value can be set to one of the following:

- » Name of a single property.
- » Comma-separated list of property names to specify multiple values.

In addition, a dotted annotation can be used to project properties into the child resource. In the purchase order example, a `GET /orders/123?fields=orderId,status` request returns the following:

```

{
  "orderId": 123,
  "status": "submitted",
  "links": [
    {"rel": "self", "href": "http://example.com/orders/123"}
  ]
}

```

A GET

/orders/123?expand=lines&fields=orderId,status,lines.price,lines.quantity request returns the following:

```
{
  "orderId": 123,
  "status": "submitted",
  "lines": {
    "items": [
      {
        "quantity": 1,
        "price": 23.5,
        "links": [
          {
            "rel": "self", "href": "http://example.com/orders/123/lines/1"
          }
        ]
      },
      {
        "quantity": 2,
        "price": 5.30,
        "links": [
          {
            "rel": "self", "href": "http://example.com/orders/123/lines/2"
          }
        ]
      }
    ],
    "links": [
      {
        "rel": "self",
        "href": "http://example.com/orders/123/lines"
      }
    ]
  },
  "links": [
    { "rel": "self", "href": "http://example.com/orders/123" }
  ]
}
```

links Query Parameter

The `links` query parameter allows the client to select which links are returned in a GET request. By default, all links are returned.

The query parameter value can be set to one of the following:

- » Name of a `rel` token.
- » Comma-separated list of `rel` tokens to specify multiple values.

For example, GET `/orders?links=self,search` returns the links with `rel=self` or `rel=search`.

Collection Resource

A collection resource represents a list of items, such as a list of employees or purchase orders.

For a collection resource, set `type=collection` in the `application/vnd.oracle.resource+json` media type.

A collection resource extends the singular resource, supporting the "links" property and the same set of query parameters (`expand`, `fields`, and `links`).

In addition, collection resources support properties and query parameters related to querying and paging. The following provides an example of a collection resource named "orders":

```
{
  "items": [
    {
      "orderId": 123,
      "description": "An office supply offer for a new hire",
      "status": "submitted",
      "lines": {
        "links": [
          {"rel": "self", "href": "http://example.com/orders/123/lines"}
        ]
      }
    },
    {
      "orderId": 223,
      "description": "Material required to build a robot",
      "status": "approved",
      "lines": {
        "links": [
          {"rel": "self", "href": "http://example.com/orders/223/lines"}
        ]
      }
    }
  ],
  "links": [
    {"rel": "self", "href": "http://example.com/orders"},
    {"rel": "create", "href": "http://example.com/orders", "method": "POST"}
  ]
}
```

Properties

A collection resource supports the properties described in this section. In addition, a collection resource extends the singular resource, supporting the "links" property, as described in 0 "links Property."

Items Property

The "items" property consists of an array of items in the collection.

hasMore Property

The "hasMore" property is a Boolean value that indicates whether there are more items to be retrieved.

totalResult Property

The "totalResult" property is an integer value that specifies the total number of the resource instances, including instances in the current response, as well as instances that have not been fetched.

limit Property

The "limit" property is an integer value that specifies the actual paging size used by the server when serving a client request. For example, the client may request a paging size value of 100, but the server may adjust the value to 25 for performance reason.

count Property

The "count" property is an integer value that specifies the actual number of items contained in the paging response. This property value must match the number of items in the "items" property.

offset Property

The "offset" property is an integer value that specifies the index of the first item to be returned. The "offset" property index begins at 0 and cannot be negative.

For example:

- » If "offset"=0, the response contains all items starting from the first item in the collection.
- » If "offset"=10 and "count"=20, then the response contains resources from 11 to 30.

Query Parameters

When requesting a collection resource via a GET request, you can specify one or more query parameters to refine the content that is returned. Collection resource query parameters support querying and paging, so that the client can control the portion of items returned in the collection resource.

A collection resource supports the properties described in this section. In addition, a collection resource extends the singular resource, supporting the same query parameters (`expand`, `fields`, and `links`). For more information, see 0 "Query Parameters."

limit Query Parameter

The `limit` query parameter is a positive integer value that specifies the maximum number of items returned by the server. In some cases, the server may override the `limit` value specified to improve performance.

offset Query Parameter

The `offset` query parameter is a non-negative integer value that specifies the index of the first item to be returned. The offset index begins at 0.

For example:

- » `offset=0` returns all items starting from the first item in the collection.
- » `offset=10` returns items starting from order 11.

Simple paging can be achieved by specifying both the `limit` and `offset` query parameters. For example, if there are 100 purchase orders, and the client issues a `GET /orders?offset=10&limit=20` request, the response includes orders from 11 to 30.

Note:

- » There is no implicit ordering if the client only specifies `limit` and `offset`. Therefore, it is recommended that the client specify an order to ensure consistent paging results.
- » For `limit/offset` paging, the client may encounter inconsistencies if the collection resource is updated between paging requests.

fromId Query Parameter

When all the items in the collection resource have unique identifiers and the identifier values are allocated chronologically, then ID-based paging can be used to ensure consistent paging results.

The `fromId` query parameter is an integer value that instructs the server to return items with IDs that are greater than the specified value. If `fromId` is specified, the server returns items in an ascending order based on their ID.

Please note:

- » The `fromId` query parameter can be combined with `limit`.
- » The `fromId` query parameter cannot be combined with `offset`.
- » If both the `fromId` and `toId` query parameters are specified, then `limit` cannot be specified.

If you specify an unsupported combination of query parameters, the request should fail.

told Query Parameter

Like `fromId`, the `toId` query parameter provides ID-based paging that can be used to ensure consistent paging results.

The `toId` query parameter is an integer value that instructs the server to return items with IDs that are less than the specified value. If only `toId` is specified, the server returns items in a descending order based on their ID. If both `fromId` and `toId` are specified, the server returns items in an ascending order based on their ID.

Please note:

- » The `toId` query parameter can be combined with `limit`.
- » The `toId` query parameter cannot be combined with `offset`.

- » If both the `fromId` and `toId` query parameters are specified, then `limit` cannot be specified.

If you specify an unsupported combination of query parameters, the request should fail.

since Query Parameter

When all the items in the collection resource have a timestamp value that indicates when the item was created or updated, then timestamp-based paging can be used to achieve consistent paging results.

The `since` query parameter is a timestamp value that specifies that only items that were created or updated after the specified time be returned. If `since` is specified, the server returns items in an ascending order based on their timestamp.

Please note:

- » The `since` query parameter can be combined with `limit`.
- » The `since` query parameter cannot be combined with `offset`.
- » If both the `until` and `since` query parameters are specified, then `limit` cannot be specified.

If you specify an unsupported combination of query parameters, the request should fail.

until Query Parameter

Like `since`, the `until` query parameter provides timestamp-based paging that can be used to ensure consistent paging results.

The `until` query parameter is a timestamp value that specifies that only items that were created or updated before the specified time be returned. If only `until` is specified, the server returns items in a descending order based on their timestamp. If both `since` and `until` are specified, the server returns items in ascending order based on the timestamp.

Please note:

- » The `until` query parameter can be combined with `limit`.
- » The `until` query parameter cannot be combined with `offset`.
- » If both the `until` and `since` query parameters are provided, then `limit` cannot be specified.

If you specify an unsupported combination of query parameters, the request will fail.

q Query Parameter

The `q` query parameter allows the client to specify a filter (a “where” clause) to restrict the items returned in the collection.

The syntax of the `q` query parameter value differs based on server implementation. The syntax of the query parameter is defined in the `search-form` resource, linked from the collection resource. For more information, see “Query Description Resource.”

totalResults Query Parameter

The `totalResults` query parameter is a Boolean value that specifies whether to return the total number of items that match the `q` query parameter. For example, if the `employees` collection contains employees with `id=1..100`, a `GET /employees?q=id>10&limit=5&totalResults=true` returns the following response:

```
{
  "items": [
    {"id": 11},
    {"id": 12},
    {"id": 13},
    {"id": 14},
    {"id": 15}
  ],
  "totalResults": 90
}
```

orderBy Query Parameter

The `orderBy` query parameter specifies the order of items returned in the response payload. The query parameter value is a comma-separated string of field names, each optionally followed by a colon and `asc` or `desc`. For example, `?orderBy=field1:asc,field2:desc`. If not specified, the server returns items in ascending order.

Pagination Links

Pagination links allow the client to access a specific range of collection resources. There are four types of pagination links:

- `next`—Navigates to the next range in the collection.
- `prev`—Navigates to the previous range in the collection.
- `first`—Navigates to the first set of resources in the collection.
- `last`— Navigates to the first set of resources in the collection.

The `next` and `prev` links should be supported, at a minimum.

For example, `GET /orders?offset=100&limit=2` returns the following:

```
{
  "items": [
    {
      "orderId": 101,
      "description": "An office supply offer for a new hire",
      "status": "submitted",
      "lines": {
        "links": [
          {"rel": "self", "href": "http://example.com/orders/101/lines"}
        ]
      }
    },
    {
      "orderId": 102,
      "description": "Material required to build a robot",

```

```

        "status": "approved",
        "lines":{
            "links": [
                {"rel": "self", "href": "http://example.com/orders/102/lines"}
            ]
        }
    ],
    "links": [
        {"rel": "self", "href":
"http://example.com/orders?offset=100&limit=2"},
        {"rel": "create", "href": "http://example.com/orders", "method":
"POST"},
        {"rel": "next", "href":
"http://example.com/orders?offset=102&limit=2"},
        {"rel": "prev", "href":
"http://example.com/orders?offset=98&limit=2"},
        {"rel": "first", "href":
"http://example.com/orders?offset=0&limit=2"},
        {"rel": "last", "href":
"http://example.com/orders?offset=2000&limit=2"}
    ]
}

```

Exception Detail Resource

The exception detail resource provides detailed information about a failed request. Typically, an exception detail resource is returned when the request fails and, in addition to the HTTP error code and error message, the server needs to provide more information about the failure.

For an exception detail resource, set `type=error` in the `application/vnd.oracle.resource+json` media type.

The exception detail resource extends the problem details described in [Problem Details for HTTP APIs](#) specification, supporting the `type`, `title`, `detail`, `status`, and `instance` properties. In addition, the resource supports the following properties:

- » `"o:errorCode"` (string) – Application-specific error code, which is different than the HTTP error code.
- » `"o:errorPath"` (string) - Path to the exception at the resource or property level.
- » `"o:errorDetails"` (array) - Details of the error consisting of an array of exception detail resources, capturing a hierarchical tree structure.

An example of the exception detail resource is shown below.

```

{
  "type": "http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.1",
  "title": "validation fails for row with key={123}",
  "detail": "There are validation errors for the object",
  "o:errorCode": "JBO-2300",
  "o:errorDetails": [
    {
      "type": "http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.1",

```

```

        "instance": "http://example.com/hr/departments/123",
        "title": "department name cannot be null",
        "o:errorPath": "departments[0]/dname",
        "o:errorCode": "JBO-2400"
    }
  ]
}

```

Status Resource

The status resource captures the status of a long running job.

For a status resource, set `type=status` in the `application/vnd.oracle.resource+json` media type.

A status resource extends the singular resource and returns one or more of the following properties:

- » `"startTime"` - Date-and-time formatted string that specifies the time at which the job started. This property is available only after a job has started.
- » `"endTime"` - Date-and-time formatted string that specifies the time at which the job has ended. This property is available only after a job has ended.
- » `"progress"` – An enum value that specifies the progress of the job. Valid values include: `"succeeded"`, `"failed"`, `"processing"`, `"pending"`, `"aborted"`, `"paused"`, `"blocked"`. This property is mandatory.
- » `"completed"` – A Boolean value that specifies whether the job has been completed. It can be used to determine whether the client needs to keep polling the `"progress"` status.
- » `"requestStatus"` – A numeric value that specifies the HTTP status code of the original job request. This property is available only after the job completes.
- » `"completedPercentage"` A numeric value between 0 and 100 that specifies the percentage that the job is complete.
- » `"intervalToPoll"` A numeric value that specifies the number of milliseconds to wait before rechecking the status of a job.
- » `"error"` A string that specifies the exception detail of the original request. This value is available only when a job has completed and failed.
- » `"links"` – An array of links to the related resources. This property value is inherited from the singular resource. If a server supports the cancelling of a request, then there may be a link with `rel=abort` that client can use to cancel the job.

An example of a status resource is shown below.

```

{
  "startTime": "2014-04-15T01:01:00Z",
  "endTime": "2014-04-15T03:01:00Z",
  "progress": "succeeded",
  "completed": true,
  "requestStatus": 200,
  "completedPercentage": 50,
  "intervalToPoll": 100,
  "links": [

```

```

    {"rel": "self", "href": "http://example.com/jobs/1001"},
    {"rel": "related", "href": "http://example.com/orders/2001"},
    {"rel": "related", "href": "http://example.com/orders/2002"}
  ]
}

```

Interaction Pattern

The status resource is particularly useful for long running or asynchronous jobs. In this scenario, the client may submit a long running job, and the server accepts the job request, typically with a 202 status code, and returns a link to a status resource. The client can poll the status resource link at periodic intervals to determine the job status.

Query Description Resource

The query description resource specifies the query syntax to use to build the `q` query parameter.

For a query description resource, set `type=search-form` in the `application/vnd.oracle.resource+json` media type.

Query description resource properties

Consider the following simple query:

```
(age>18;name co 'Alice') or (age<10 and address pr)
```

In the above query:

- » Age is greater than 18 and name contains Alice, or
- » Age is less than 10 and there is an address.

To communicate the syntax of this simple query, a `search-form` resource may be provided by the server with the following content:

```

{
  "syntax": {
    "operators": {
      "eq": {"code": ["="], "types": ["string", "integer", "boolean"],
        "description": "equal", "example": "age=18"},
      "ne": {"code": ["!="], "types": ["string", "integer", "boolean"],
        "description": "not equal", "example": "age!=18"},
      "gt": {"code": [ ">" ], "types": ["integer", "boolean", "number"],
        "description": "greater than", "example": "age>18"},
      "lt": {"code": [ "<" ], "types": ["integer", "boolean", "number"],
        "description": "less than", "example": "age<18"},
      "ge": {"code": [ ">=" ], "types": ["integer", "boolean", "number"],
        "description": "greater than or equal to", "example": "age>=18"},
      "le": {"code": [ "<=" ], "types": ["integer", "boolean", "number"],
        "description": "less than or equal to", "example": "age<=18"},
      "co": {"code": ["co"], "types": ["string"], "description":
        "contains", "example": "name co 'Alice'"},
      "sw": {"code": ["sw"], "types": ["string"], "description": "starts
        with", "example": "name sw 'Alice'"},
      "ew": {"code": ["ew"], "types": ["string"], "description": "ends
        with", "example": "name ew 'Alice'"},

```



```

    "pr": {"code": ["pr"], "types": ["string", "integer", "boolean",
"number"], "description": "not null", "example": "name pr"}
  },
  "logicalOperators": {
    "and": {"code": [";", "and"], "description": "logical and",
"example": "age>18;name co 'Alice'"},
    "or": {"code": ["or"]},
    "not": {"code": ["not"]}
  },
  "termDelimiter": [" "],
  "precedenceGrouping": "(" ")"
},
"propertyOperators": {
  "ename": {
    "operators": ["eq", "ne", "co", "sw", "ew"]
  },
  "empno": {
    "operators": ["eq", "ne", "lt", "gt"]
  },
  "birthday": {
    "operators": ["eq", "ne", "gt", "lt", "pr"]
  }
}
}
}

```

The "syntax" section describes the query syntax, and includes the following subsections:

- » "operators" - specifies what strings are used for each property operator.
- » "logicalOperators" - describes the strings that can be used for logical operators.
- » "termDelimiter": - specifies the delimiter that is used to separate terms.
- » "precedenceGrouping" - groups expressions to change the standard order of operation.

The "propertyOperators" section defines the resource property specific restrictions. In the above example, only a subset of the property operators can be applied to "ename". For example, the "pr" operator cannot be used for "ename".

Interaction Pattern

Typically, the `search-form` is used in conjunction with a collection resource. If a `search-form` resource is available, then the collection resource should include a link to it using `rel=search-form`. The client can navigate to the `search-form` resource and use that to build the `q` query parameter and then submit filter requests against the collection resource. The `search-form` resource extends the singular resource, supporting the "links" property, as described in 0 "links Property." In the "links" property, the resource may have a link back to the collection resource if the `search-form` is specific to a particular collection resource. In this case, the `rel` of the link should be "search".

create-form Resource

The `create-form` resource is used as a template for clients to obtain default values when creating a new resource.

For a `create-form` resource, set `type=create-form` in the `application/vnd.oracle.resource+json` media type.

create-form Resource Properties

A `create-form` resource extends singular resource, supporting the "links" property, as described in 0 "links Property." The `create-form` resource can be associated with an existing singular or collection resource. The `create-form` resource contains the properties with default values.

For example, a newly created purchase order may have a status property value set to "submitted", as shown below:

```
{
  "status": "submitted",
  "links": [
    {"rel": "self", "href": "http://example.com/orders-create-form"},
    {"rel": "create", "href": "http://example.com/orders"}
  ]
}
```

Interaction Pattern

The `create-form` resource should have a link to the relevant singular or collection resource and `"rel"="create"`. The singular or collection resource should have a link to the `create-form` resource with `rel=create-form`.

edit-form Resource

The `edit-form` resource is used as a template for clients to determine which properties are updateable while editing a resource.

For an `edit-form` resource, set `type=edit-form` in the `application/vnd.oracle.resource+json` media type.

edit-form Resource Properties

An `edit-form` resource extends a singular resource, supporting the "links" property, as described in 0 "links Property." The `edit-form` resource is associated with an existing singular resource and contains the properties that can be modified.

For example, the following provides an example of the `edit-form` resource for an order resource:

```
{
  "description": "An office supply offer for a new hire",
  "status": "submitted",
  "links": [
    {"rel": "self", "href": "http://example.com/orders/102/edit-form"},
    {"rel": "edit", "href": "http://example.com/orders"}
  ]
}
```

Interaction Pattern

The `edit-form` resource should have a link to the related resource with `"rel"="edit"`. The original resource should have a link to the `edit-form` resource with `rel=edit-form`.

JSON Schema

The following is the JSON schema that contains the definition of the specialized resource types:

- » Singular resource (type=singular) - [#/definitions/singularResource](#)
- » Collection resource (type=collection) - [#/definitions/collectionPagedResource](#)
- » Exception detail resource (type=error) - [#/definitions/exceptionDetailType](#)
- » Status resource (type=status) - [#/definitions/statusResource](#)
- » Query description resource (type=search-form) - [#/definitions/searchFormResource](#)
- » create-form resource (type=create-form) - [#/definitions/createFormResource](#)
- » edit-form resource (type=edit-form) - [#/definitions/editFormResource](#)

```
{
  "$schema": "http://json-schema.org/draft-04/hyper-schema#",
  "description": "Describes Oracle-specific meta properties that extends
the official Hypermedia schema.",
  "allOf": [
    {
      "$ref": "http://json-schema.org/draft-04/hyper-schema#"
    }
  ],
  "definitions": {
    "exceptionDetailType": {
      "title": "Error Detail",
      "description": "Complex type that contains error details for a REST
call.",
      "type": "object",
      "properties": {
        "type": {
          "type": "string",
          "description": "Absolute URI [RFC3986] that identifies the
problem type. When dereferenced, it SHOULD provide a human-readable
summary of the problem (for example, using HTML).",
        },
        "title": {
          "type": "string",
          "description": "Short, human-readable summary of the problem.
The summary SHOULD NOT change for subsequent occurrences of the problem,
except for purposes of localization."
        },
        "status": {
          "type": "integer",
          "description": "HTTP status code for this occurrence of the
problem, set by the origin server."
        },
        "detail": {
          "type": "string",
          "description": "Human-readable description specific to this
occurrence of the problem."
        },
        "instance": {
          "type": "string",

```

```

        "description": "Absolute URI that identifies the specific
occurrence of the problem. It may or may not provide additional
information if dereferenced."
    },
    "o:errorCode": {
        "type": "string",
        "description": "Application error code, which is different from
HTTP error code."
    },
    "o:errorPath": {
        "type": "string",
        "description": "Path to the problem at the resource or property
level."
    },
    "o:errorDetails": {
        "description": "Details of the error message, consisting of a
hierarchical tree structure.",
        "type": "array",
        "items": {"$ref": "#/definitions/exceptionDetailType"}
    }
},
"required": ["type", "title"]
},

"instanceLink": {
    "type": "object",
    "description": "Metadata describing link description objects that
MAY appear in the JSON instance representation.",
    "properties": {
        "href": {
            "description": "URI [RFC3986] or URI Template [RFC6570]. If the
value is set to URI Template, then the \"templated\" property must be set
to true.",
            "type": "string"
        },
        "rel": {
            "description": "Name of the link relation that, in addition to
the type property, can be used to retrieve link details. For example,
href or profile.",
            "type": "string"
        },
        "templated": {
            "description": "Boolean flag that specifies that \"href\"
property is a URI or URI Template. If the property is a URI template, set
this value to true. By default, this value is false.",
            "type": "boolean"
        },
        "mediaType": {
            "description": "Media type, as defined by RFC 2046, describing
the link target.",
            "type": "string"
        },
        "method": {
            "description": "HTTP method for requesting the target of the
link.",
            "type": "string"
        }
    }
}

```

```

    },
    "profile":
    {
        "description": "Link to the metadata of the resource, such as
JSON-schema, that describes the resource expected when dereferencing the
target resource..",
        "type": "string"
    }
},
"required": ["href", "rel"]
},

"expandQueryParam": {
    "description": "Comma-delimited string of child resource names that
you want to expand so that the contents return with the parent in the
same GET request.",
    "type": "string"
},
"fieldsQueryParam": {
    "description": "Comma-delimited string of field names that you
wanted returned in a GET request.",
    "type": "string"
},
"limitQueryParam": {
    "type": "integer",
    "description": "Positive integer value that specifies the maximum
number of items returned by the server."
},
"offsetQueryParam": {
    "type": "integer",
    "description": "Non-negative integer values that specifies the
index of the first item to be returned. The offset index begins at 0. By
default, the offset is 0, which returns all items starting from the first
item in the collection.",
    "minimum":0,
    "default":0
},
"fromIdQueryParam": {
    "type": "integer",
    "description": "Integer value that instructs the server to return
items with IDs that are greater than the specified value."
},
"toIdQueryParam": {
    "type": "integer",
    "description": "Integer value that instructs the server to return
items with IDs that are less than the specified value."
},
"sinceQueryParam": {
    "type": "string",
    "format": "date-time",
    "description": "Timestamp value that specifies that only items that
were created or updated after the specified time be returned."
},
"untilQueryParam": {
    "type": "string",
    "format": "date-time",
    "description": "Timestamp value that specifies that only items that
were created or updated before the specified time be returned."
},

```

```

    "qQueryParam": {
      "type": "string",
      "description": "Filter (a \"where\" clause) to restrict the items
returned in the collection. By default, no filtering is applied."
    },
    "totalResultsQueryParam": {
      "description": "Boolean value that specifies whether to calculate
the totalResults property. By default, this value is set to false
indicating that totalResults is not calculated.",
      "type": "boolean",
      "default": false
    },
    "orderByQueryParam": {
      "description": "Comma-separated string of field names, each
optionally followed by asc or desc, that specifies the order of items
returned in the response payload." ,
      "type": "string"
    },
    "linksQueryParam": {
      "type": "string",
      "description": "Comma-separated list of relation types that are
returned in a GET request. By default, all links are returned."
    },
    "totalResultsProp": {
      "type": "integer",
      "description": "Total count of the resource instances, including
both the instances in the current range and the instances on the server
that satisfy the request."
    },
    "limitProp": {
      "type": "integer",
      "description": "Actual paging size used by the server."
    },
    "countProp": {
      "type": "integer",
      "description": "Number of resource instances returned in the
current range."
    },
    "hasMoreProp": {
      "type": "boolean",
      "description": "Boolean value that is set to true if more resources
are available on the server than the subset returned in current page."
    },
    "offsetProp": {
      "type": "integer",
      "description": "Offset value used in the current page. "
    },
    "queryParams": {
      "description": "Metadata describing common schema properties of an
API (for example, Query Parameters). Each property can be individually
referred to by its 'id' property, using JSON References
(http://tools.ietf.org/html/draft-pbryan-zyp-json-ref-00).",
      "type": "object",
      "properties":
      {
        "expand": {
          "$ref": "#/definitions/expandQueryParam"
        }
      }
    }
  }

```

```

    },
    "fields": {
      "$ref": "#/definitions/fieldsQueryParam"
    },
    "limit": {
      "$ref": "#/definitions/limitQueryParam"
    },
    "offset": {
      "$ref": "#/definitions/offsetQueryParam"
    },
    "fromId": {
      "$ref": "#/definitions/fromIdQueryParam"
    },
    "toId": {
      "$ref": "#/definitions/toIdQueryParam"
    },
    "since": {
      "$ref": "#/definitions/sinceQueryParam"
    },
    "until": {
      "$ref": "#/definitions/untilQueryParam"
    },
    "q": {
      "$ref": "#/definitions/qQueryParam"
    },
    "totalResults": {
      "$ref": "#/definitions/totalResultsQueryParam"
    },
    "orderBy": {
      "$ref": "#/definitions/orderByQueryParam"
    },
    "links": {
      "$ref": "#/definitions/linksQueryParam"
    }
  }
},
"singularResource": {
  "title": "Singular Resource",
  "description": "Oracle base singular resource schema definition.",
  "type": "object",
  "properties": {
    "links": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/instanceLink"
      }
    }
  }
},
"links": [
  {
    "href": "#",
    "rel": "describedby"
  }
]
},
"collectionBaseResource": {
  "allOf": [

```

```

    {
      "$ref": "#/definitions/singularResource"
    }
  ],
  "title": "Base Collection Resource",
  "description": "Oracle base collection resource schema definition.
",
  "properties": {
    "items": {
      "type": "array",
      "items": {
        "type": "object"
      }
    }
  }
},
"collectionPagedResource": {
  "allOf": [
    {
      "$ref": "#/definitions/collectionBaseResource"
    }
  ],
  "title": "Collection Paging Resource",
  "description": "Oracle base collection resource schema definition.
",
  "properties": {
    "hasMore": {
      "$ref": "#/definitions/hasMoreProp"
    },
    "totalResults": {
      "$ref": "#/definitions/totalResultsProp"
    },
    "limit": {
      "$ref": "#/definitions/limitProp"
    },
    "count": {
      "$ref": "#/definitions/countProp"
    },
    "offset": {
      "$ref": "#/definitions/offsetProp"
    }
  },
  "links": [
    {
      "href": "",
      "rel": "canonical",
      "schema": {
        {
          "$ref": "#/definitions/queryParams"
        }
      }
    },
    {
      "href": "{next}",
      "rel": "prev"
    },
    {
      "href": "{prev}",
      "rel": "next"
    }
  ],

```



```

    {
      "href": "{first}",
      "rel": "first"
    },
    {
      "href": "{last}",
      "rel": "last"
    }
  ]
},

"statusResource": {
  "title": "Job Status",
  "description": "Captures the status of a long running job.",
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/singularResource"}
  ],
  "properties": {
    "startTime": {
      "type": "string",
      "format": "datetime",
      "description": "Date-and-time formatted string that specifies
the time at which the job started. This property is available only after
a job has started."
    },
    "endTime": {
      "type": "string",
      "format": "datetime",
      "description": "Date-and-time formatted string that specifies
the time at which the job has ended. This property is available only
after a job has ended."
    },
    "progress": {
      "enum": ["succeeded", "failed", "processing",
"pending", "aborted", "paused", "blocked"],
      "description": "The current progress of the job. These values
indicate that the job has ended: 'succeeded', 'failed', 'aborted'. The
value 'blocked' means that the job requires action, such as waiting for a
human to approve something. The values that indicate the job is in
process are: 'pending', 'processing', 'paused'"
    },
    "completed": {
      "type": "boolean",
      "description": "Boolean value that specifies whether the job
has been completed."
    },
    "requestStatus": {
      "type": "number",
      "description": "Numeric value that specifies the HTTP status
code of the original job request. This is not equivalent to the polling
request status. This property is available only after the job completes."
    },
    "message": {
      "type": "string",
      "description": "Human-readable message that describes the
current processing status."
    },
    "completedPercentage": {

```

```

        "type": "number",
        "description": "Numeric value between 0 and 100 that specifies
the percentage that the job is complete."
    },
    "intervalToPoll": {
        "type": "number",
        "description": "Numeric value that specifies the number of
milliseconds to wait before rechecking the status of a job."
    },
    "error": {
        "$ref": "#/definitions/exceptionDetailType",
        "description": "String that specifies the exception detail of
the original request. This value is available only when a job has
completed and failed."
    },
    "links": {
        "type": "array",
        "item": {
            "$ref": "#/definitions/link"
        },
        "description": "Array of links to the related resources. This
property value is inherited from the singular resource. If a server
supports the cancelling of a request, then there may be a link with
rel=abort that client can use to cancel the job."
    }
},
"required": ["progress"]
},

"opType": {
    "type": "object",
    "properties": {
        "code": {
            "$ref": "http://json-schema.org/draft-
04/schema#/definitions/stringArray",
            "description": "Strings used in this custom query format to
represent an operator."
        },
        "types": {
            "type": "array",
            "items": {
                "$ref": "http://json-schema.org/draft-
04/schema#/definitions/simpleTypes"
            },
            "description": "Types of properties that are applicable for
this operator. For example, startWith applies to string only."
        },
        "description": {
            "type": "string",
            "description": "Description of the function of the operator."
        },
        "example": {
            "type": "string",
            "description": "Example of how to use the operator."
        }
    }
},
"searchFormResource": {
    "title": "Search form resource",

```

```

    "description": "Description of the query syntax.",
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/singularResource"}
    ],
    "properties": {
      "syntax": {
        "type": "object",
        "properties": {
          "operators": {
            "type": "object",
            "description": "Operators supported by the custom query
format.",
            "properties": {
              "eq": {
                "$ref": "#/definitions/opType",
                "description": "equal"
              },
              "ne": {
                "$ref": "#/definitions/opType",
                "description": "not equal"
              },
              "gt": {
                "$ref": "#/definitions/opType",
                "description": "greater"
              },
              "lt": {
                "$ref": "#/definitions/opType",
                "description": "less than"
              },
              "ge": {
                "$ref": "#/definitions/opType",
                "description": "greater or equal"
              },
              "le": {
                "$ref": "#/definitions/opType",
                "description": "less than or equal"
              },
              "co": {
                "$ref": "#/definitions/opType",
                "description": "contains"
              },
              "sw": {
                "$ref": "#/definitions/opType",
                "description": "startWith"
              },
              "ew": {
                "$ref": "#/definitions/opType",
                "description": "end with"
              },
              "pr": {
                "$ref": "#/definitions/opType",
                "description": "is not null"
              }
            }
          }
        }
      },
      "logicalOperators": {
        "type": "object",
        "description": "Logical operators supported.",

```

```

    "properties": {
      "and": {
        "$ref": "#/definitions/opType",
        "description": "logical and"
      },
      "or": {
        "$ref": "#/definitions/opType",
        "description": "logical or"
      },
      "not": {
        "$ref": "#/definitions/opType",
        "description": "logical not"
      }
    }
  },
  "termDelimiter": {
    "$ref": "http://json-schema.org/draft-04/schema#/definitions/stringArray",
    "description": "Strings that can be used to delimit items."
  },
  "precedenceGrouping": {
    "type": "string",
    "description": "Strings that can be used to group terms."
  }
},
"propertyOperators": {
  "type": "object",
  "description": "Operators that can be applied for a particular
property"
},
"required": ["syntax"]
},

"createFormResource": {
  "title": "A create-form resource",
  "description": "Template used to create a new resource. The
resource contains properties with default values.",
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/singularResource"}
  ]
},

"editFormResource": {
  "title": "An edit-form resource",
  "description": "Template used to update a resource. The resource
contains properties that can be modified.",
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/singularResource"}
  ]
}
}
}

```

IANA Considerations

The proposed MIME media type for REST resource is defined as follows:

- » type name: application
- » subtype name: vnd.oracle.resource+json
- » required parameters: type

References

Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, March 2012.

Informative References

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [json-schema] Court, G. and K. Zyp, "[JSON Schema, draft 4](#)", August 2013,
- [REST] Fielding, R., "[Representational State Transfer \(REST\)](#)", 2000,
- [ProblemDetail] Nottingham, M. and E. Wilde, "[Problem Details for HTTP APIs](#)", 2014,

Change Log




Version	Description	Date	Author
Draft-0	Initial Draft	Oct, 2014	Ning Dong
0.1	Added a few values in enum of "progress" property of Status resource, also changed success to succeeded	Mar, 2015	Ning Dong



Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0315