Media Type Representation

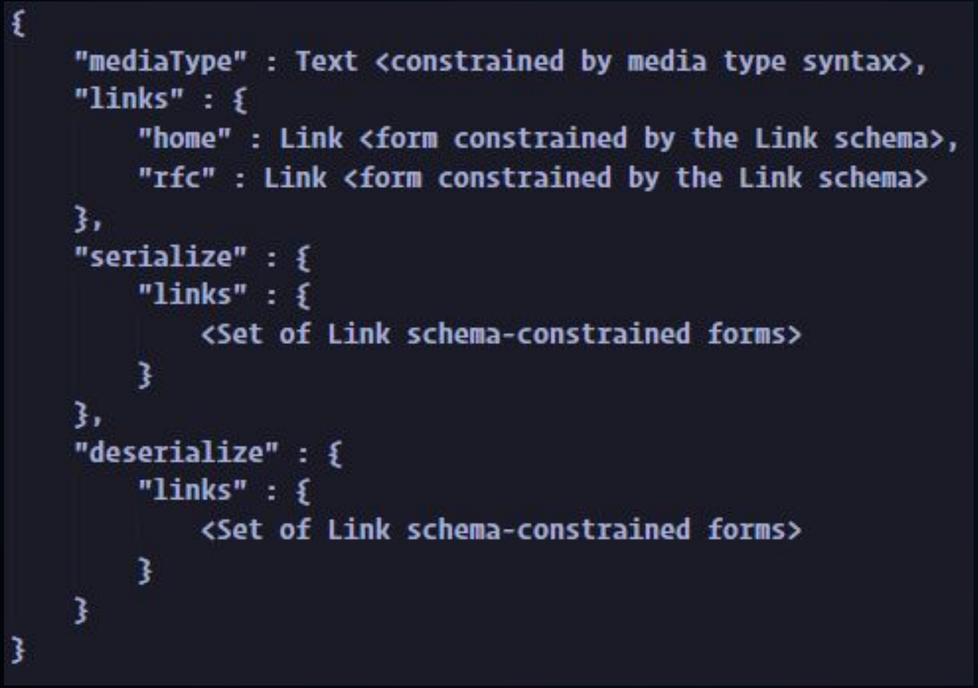
Rule: A consistent form should be used to represent media type formats

- application/wrml; format=URI; schema=URI

```
application/wrml;
        format="http://api.formats.wrml.org/application/json”;
        schema="http://api.schemas.wrml.org/common/Format”
application/wrml;
        format="http://api.formats.wrml.org/application/json”
        schema="http://api.schemas.wrml.org/soccer/Player”;
application/wrml;
        format="http://foo.bar.baz/uberformat”;
        schema="http://foo.bar.baz/uberschema”;
```

Format

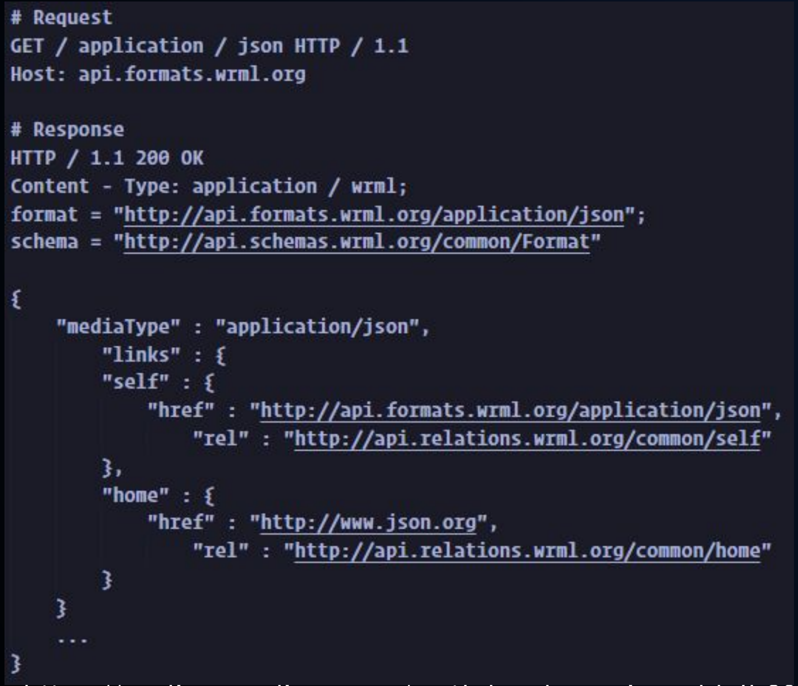
- When represented using JSON, a Format has the following consistent structure:



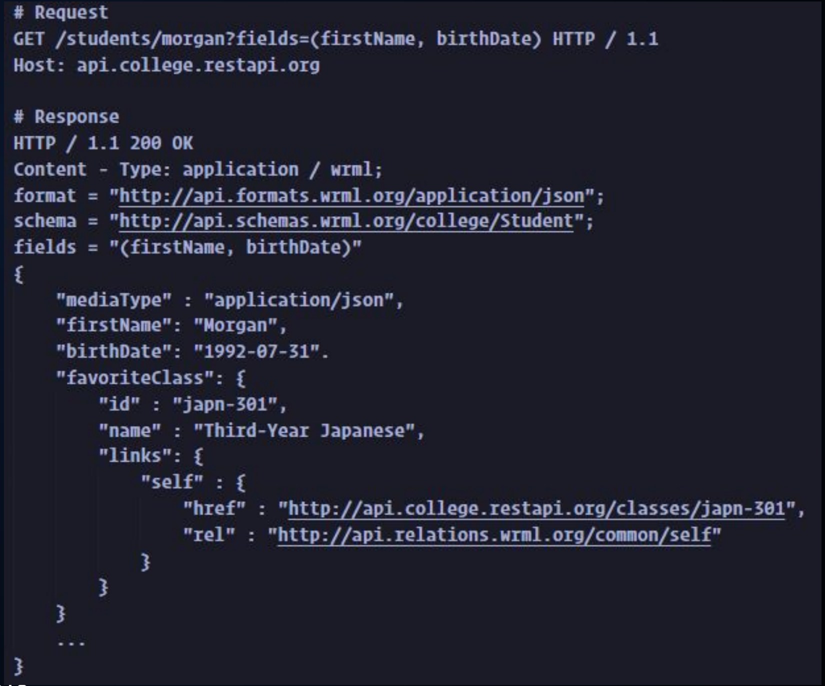
```
{
    "mediaType" : Text <constrained by media type syntax>,
    "links" : {
        "home" : Link <form constrained by the Link schema>,
        "rfc" : Link <form constrained by the Link schema>
    },
    "serialize" : {
        "links" : {
            <Set of Link schema-constrained forms>
        }
    },
    "deserialize" : {
        "links" : {
            <Set of Link schema-constrained forms>
        }
    }
}
```

Format Examples

- Below is an example of an HTTP request and response for the JSON format document's representation



```
# Request
GET / application / json HTTP / 1.1
Host: api.formats.wrml.org

# Response
HTTP / 1.1 200 OK
Content - Type: application / wrml;
format = "http://api.formats.wrml.org/application/json";
schema = "http://api.schemas.wrml.org/common/Format"

{
    "mediaType" : "application/json",
        "links" : {
        "self" : {
            "href" : "http://api.formats.wrml.org/application/json",
                "rel" : "http://api.relations.wrml.org/common/self"
        },
        "home" : {
            "href" : "http://www.json.org",
                "rel" : "http://api.relations.wrml.org/common/home"
        }
    }
    ...
}
```

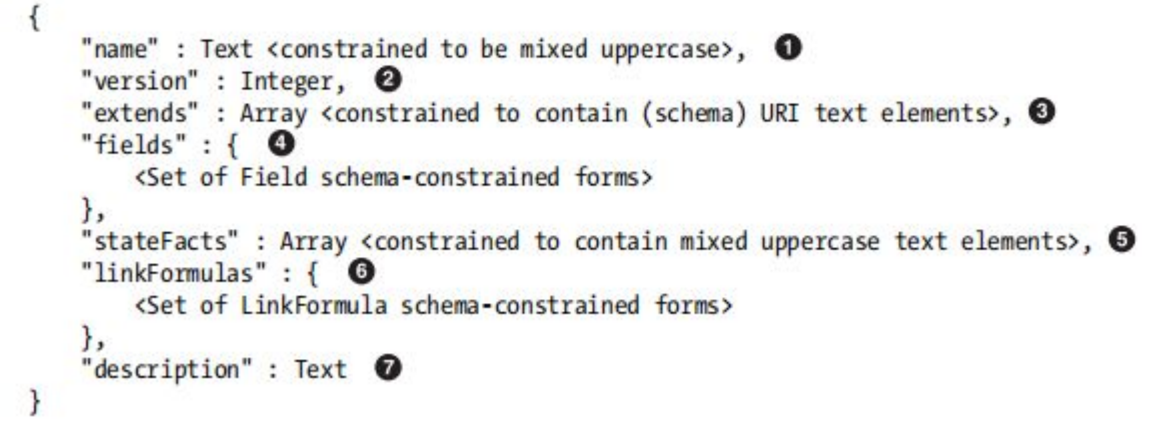


```
# Request
GET /students/morgan?fields=(firstName, birthDate) HTTP / 1.1
Host: api.college.restapi.org

# Response
HTTP / 1.1 200 OK
Content - Type: application / wrml;
format = "http://api.formats.wrml.org/application/json";
schema = "http://api.schemas.wrml.org/college/Student";
fields = "(firstName, birthDate)"
{
    "mediaType" : "application/json",
    "firstName": "Morgan",
    "birthDate": "1992-07-31".
    "favoriteClass": {
        "id" : "japn-301",
        "name" : "Third-Year Japanese",
        "links": {
            "self" : {
                "href" : "http://api.college.restapi.org/classes/japn-301",
                "rel" : "http://api.relations.wrml.org/common/self"
            }
        }
    }
    ...
}
```

https://cagline.medium.com/restful-web-services-ddafb8019f2e

Rule: A consistent form should be used to represent media type schemas

- Schemas provide contractual resource type definitions, which are a crucial component of the interface that binds a REST API and its clients together
- A representational form, or just form for short, consists of the fields and links as detailed by the “blueprints” of its associated schema document
- When represented using JSON, a Schema has the following consistent form

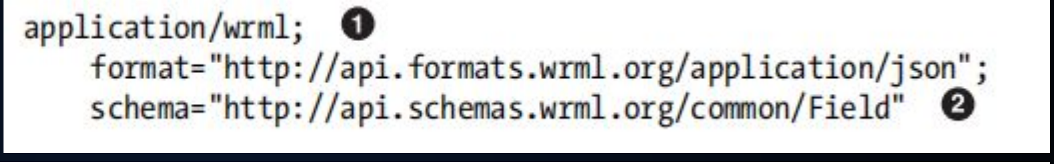


```
{
    "name" : Text <constrained to be mixed uppercase>, ❶
    "version" : Integer, ❷
    "extends" : Array <constrained to contain (schema) URI text elements>, ❸
    "fields" : { ❹
        <Set of Field schema-constrained forms>
    },
    "stateFacts" : Array <constrained to contain mixed uppercase text elements>, ❺
    "linkFormulas" : { ❻
        <Set of LinkFormula schema-constrained forms>
    },
    "description" : Text ❼
}
```

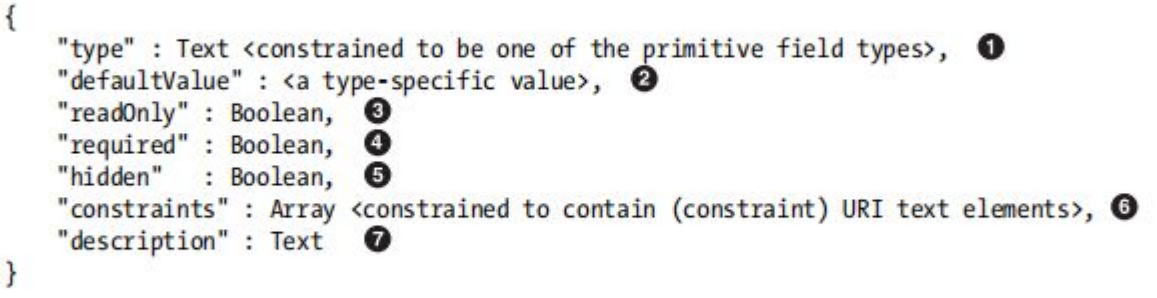
Field Representation

Each field’s value may be one of the following types:

- Boolean
- Choise(similat to enum)
- DateTime
- Double
- Integer
- List
- Schema
- Text
- Null



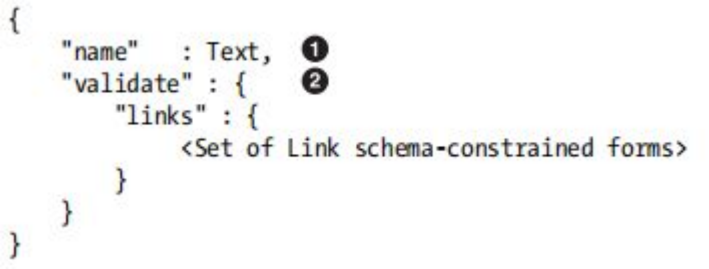
```
application/wrml;  ❶
    format="http://api.formats.wrml.org/application/json";
    schema="http://api.schemas.wrml.org/common/Field"  ❷
```



```
{
   "type" : Text <constrained to be one of the primitive field types>,  ❶
   "defaultValue" : <a type-specific value>,  ❷
   "readOnly" : Boolean,  ❸
   "required" : Boolean,  ❹
   "hidden"   : Boolean,  ❺
   "constraints" : Array <constrained to contain (constraint) URI text elements>, ❻
   "description" : Text  ❼
}
```

Constraint Representation

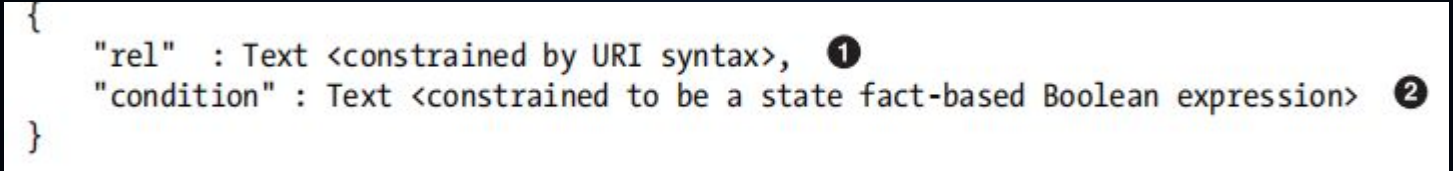
- A schema field’s constraints value lists the URIs of the constraints that are applied to a form’s associated field value



```
{
   "name"   : Text,  ❶
   "validate" : {    ❷
      "links" : {
           <Set of Link schema-constrained forms>
      }
   }
}
```

Link Formula Representation

- A schema link formula equates the availability of a state-sensitive link in a response message body’s form with a Boolean expression that uses the schema’s state facts as operands



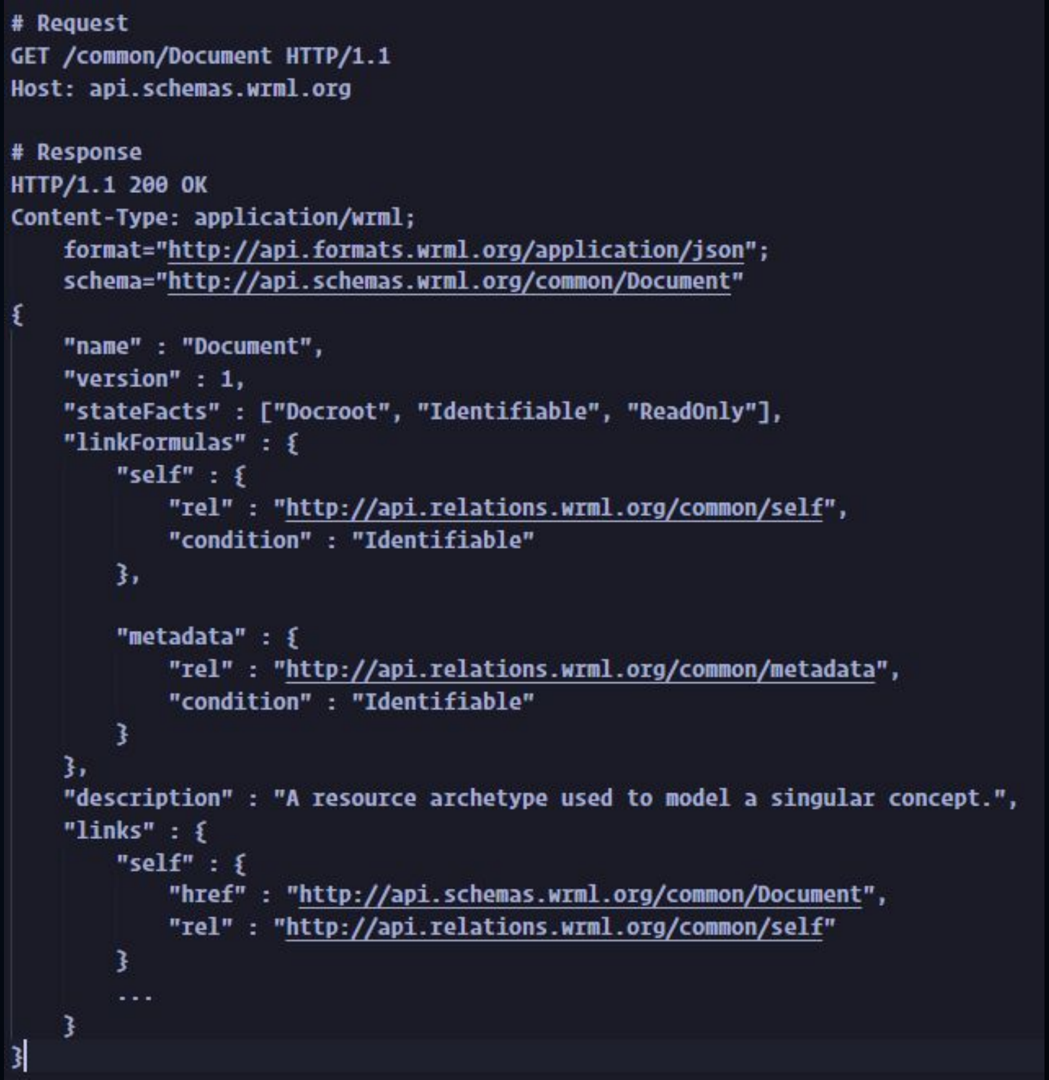
```
{
    "rel"   : Text <constrained by URI syntax>,  ❶
    "condition" : Text <constrained to be a state fact-based Boolean expression>  ❷
}
```

Document Schema Representation

- Document is the base form for all resource types.

EXAMPLE

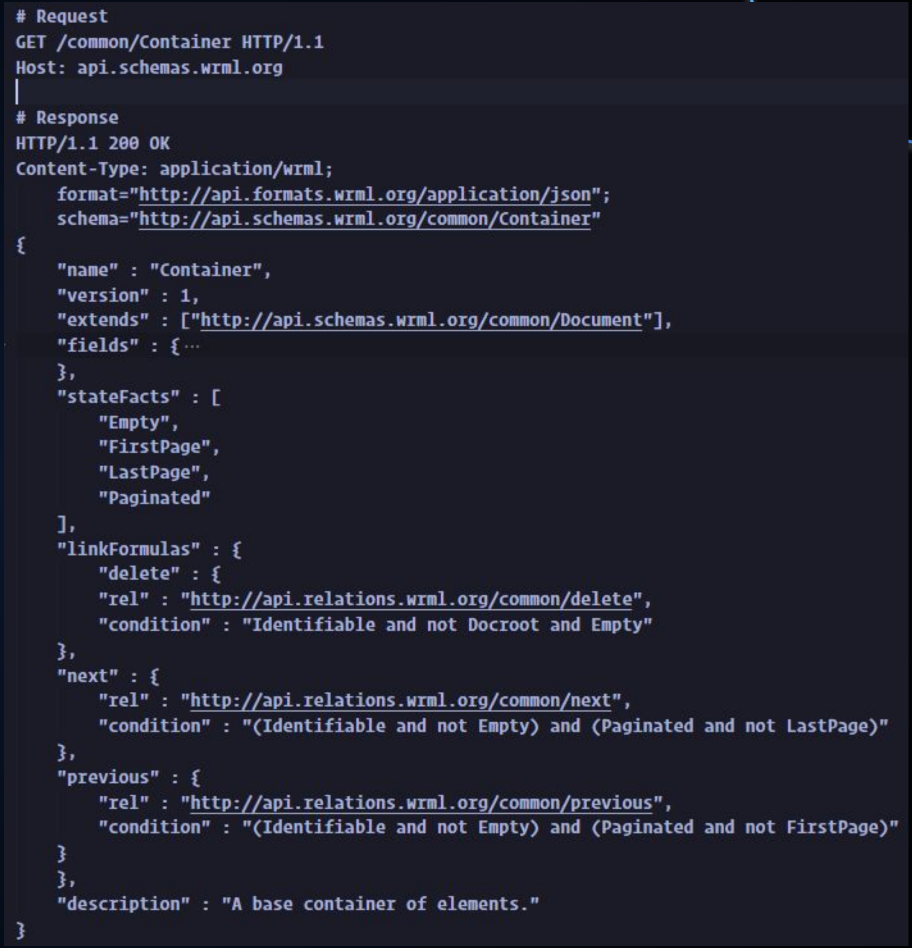
Example



```
# Request
GET /common/Document HTTP/1.1
Host: api.schemas.wrml.org

# Response
HTTP/1.1 200 OK
Content-Type: application/wrml;
    format="http://api.formats.wrml.org/application/json";
    schema="http://api.schemas.wrml.org/common/Document"
{
    "name" : "Document",
    "version" : 1,
    "stateFacts" : ["Docroot", "Identifiable", "ReadOnly"],
    "linkFormulas" : {
        "self" : {
            "rel" : "http://api.relations.wrml.org/common/self",
            "condition" : "Identifiable"
        },

        "metadata" : {
            "rel" : "http://api.relations.wrml.org/common/metadata",
            "condition" : "Identifiable"
        }
    },
    "description" : "A resource archetype used to model a singular concept.",
    "links" : {
        "self" : {
            "href" : "http://api.schemas.wrml.org/common/Document",
            "rel" : "http://api.relations.wrml.org/common/self"
        }
        ...
    }
}
```

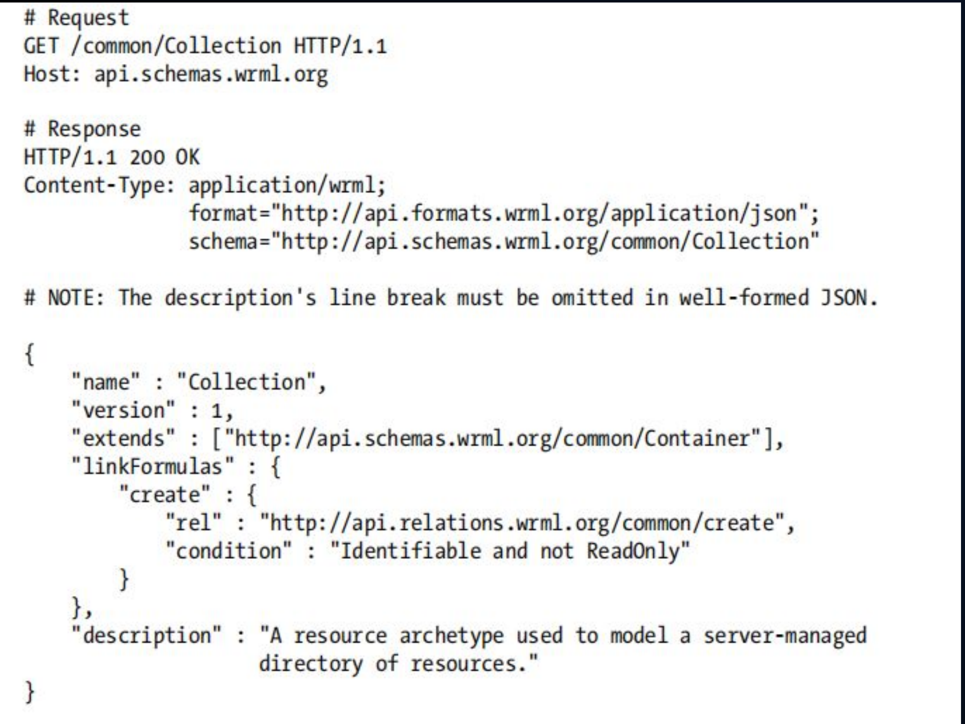
Container Schema Representation

- Collection models a server managed directory of resources and a Store is a client-managed resource repository



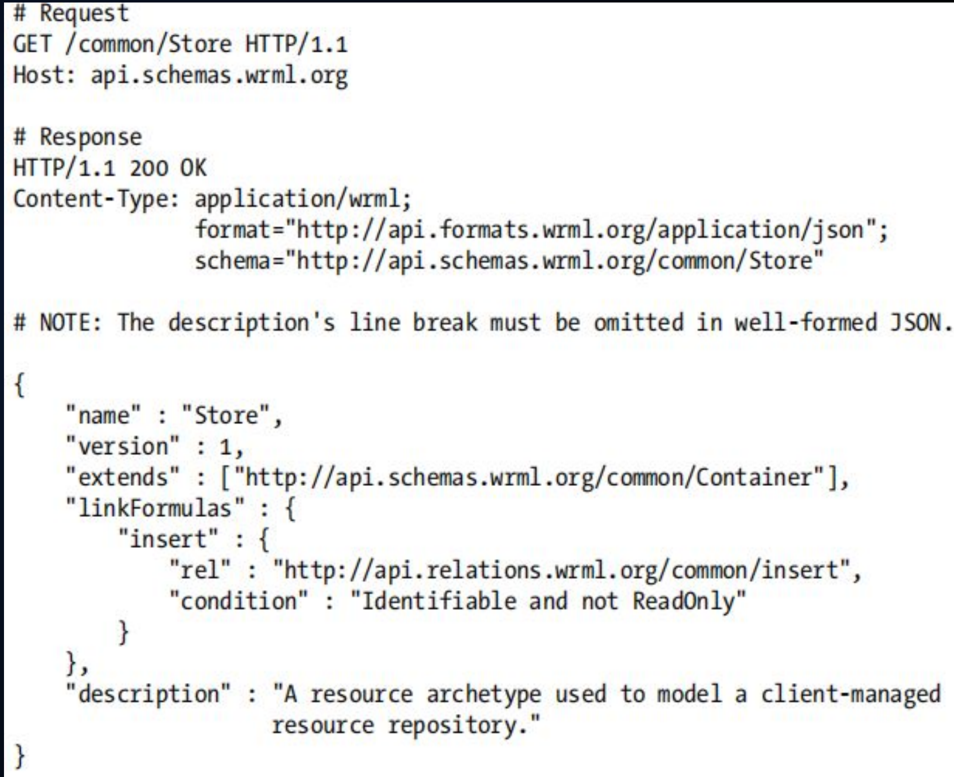
```
# Request
GET /common/Container HTTP/1.1
Host: api.schemas.wrml.org

# Response
HTTP/1.1 200 OK
Content-Type: application/wrml;
    format="http://api.formats.wrml.org/application/json";
    schema="http://api.schemas.wrml.org/common/Container"
{
    "name" : "Container",
    "version" : 1,
    "extends" : ["http://api.schemas.wrml.org/common/Document"],
    "fields" : {···
    },
    "stateFacts" : [
        "Empty",
        "FirstPage",
        "LastPage",
        "Paginated"
    ],
    "linkFormulas" : {
        "delete" : {
        "rel" : "http://api.relations.wrml.org/common/delete",
        "condition" : "Identifiable and not Docroot and Empty"
    },
    "next" : {
        "rel" : "http://api.relations.wrml.org/common/next",
        "condition" : "(Identifiable and not Empty) and (Paginated and not LastPage)"
    },
    "previous" : {
        "rel" : "http://api.relations.wrml.org/common/previous",
        "condition" : "(Identifiable and not Empty) and (Paginated and not FirstPage)"
    }
    },
    "description" : "A base container of elements."
}
```

Collection Schema Representation



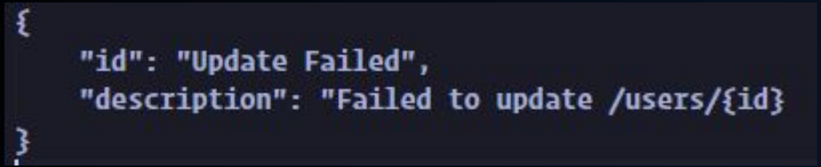
```
# Request
GET /common/Collection HTTP/1.1
Host: api.schemas.wrml.org

# Response
HTTP/1.1 200 OK
Content-Type: application/wrml;
              format="http://api.formats.wrml.org/application/json";
              schema="http://api.schemas.wrml.org/common/Collection"

# NOTE: The description's line break must be omitted in well-formed JSON.

{
    "name" : "Collection",
    "version" : 1,
    "extends" : ["http://api.schemas.wrml.org/common/Container"],
    "linkFormulas" : {
        "create" : {
            "rel" : "http://api.relations.wrml.org/common/create",
            "condition" : "Identifiable and not ReadOnly"
        }
    },
    "description" : "A resource archetype used to model a server-managed
                    directory of resources."
}
```

Store Schema Representation

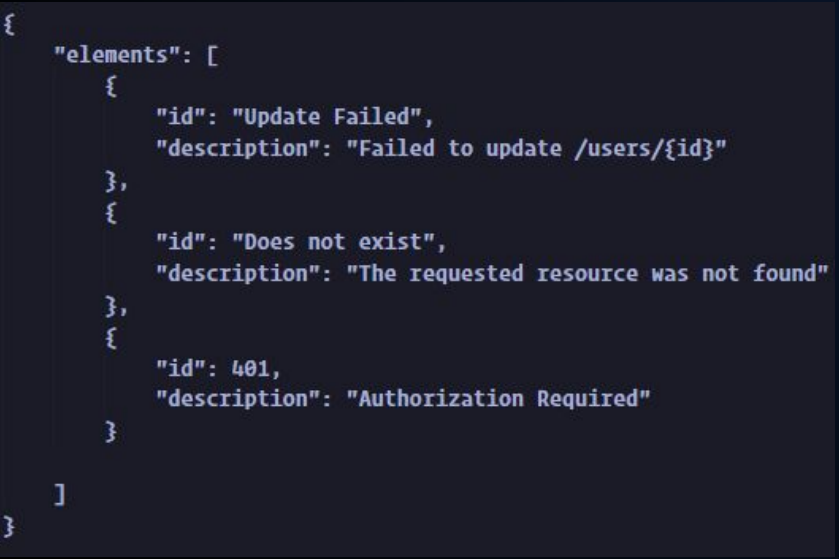


```
# Request
GET /common/Store HTTP/1.1
Host: api.schemas.wrml.org

# Response
HTTP/1.1 200 OK
Content-Type: application/wrml;
              format="http://api.formats.wrml.org/application/json";
              schema="http://api.schemas.wrml.org/common/Store"

# NOTE: The description's line break must be omitted in well-formed JSON.

{
    "name" : "Store",
    "version" : 1,
    "extends" : ["http://api.schemas.wrml.org/common/Container"],
    "linkFormulas" : {
        "insert" : {
            "rel" : "http://api.relations.wrml.org/common/insert",
            "condition" : "Identifiable and not ReadOnly"
        }
    },
    "description" : "A resource archetype used to model a client-managed
                    resource repository."
}
```

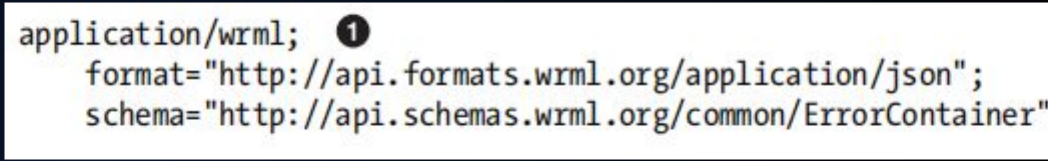
Error Representation



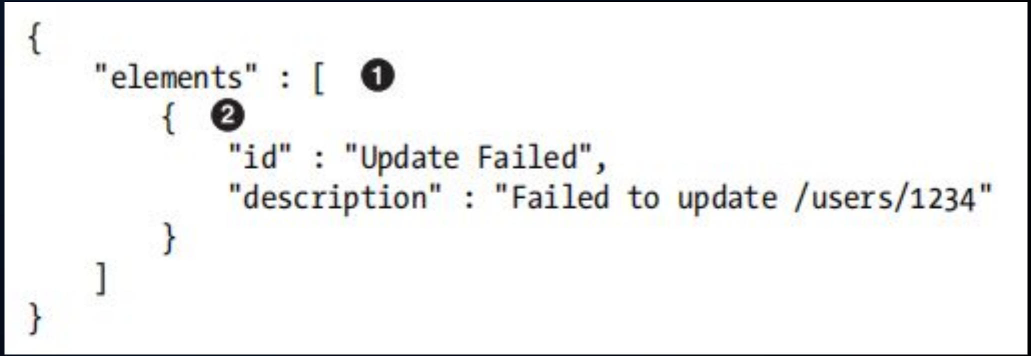
```
{
    "id": "Update Failed",
    "description": "Failed to update /users/{id}
}
```



```
{
    "elements": [
        {
            "id": "Update Failed",
            "description": "Failed to update /users/{id}"
        },
        {
            "id": "Does not exist",
            "description": "The requested resource was not found"
        },
        {
            "id": 401,
            "description": "Authorization Required"
        }

    ]
}
```



```
application/wrml;  ❶
    format="http://api.formats.wrml.org/application/json";
    schema="http://api.schemas.wrml.org/common/ErrorContainer"
```



```
{
    "elements" : [  ❶
        {  ❷
            "id" : "Update Failed",
            "description" : "Failed to update /users/1234"
        }
    ]
}
```

Example self created