Technical Disclosure Commons

Defensive Publications Series

March 17, 2017

A method of providing meaningful, self-descripting metadata to REST return JSON data

Lee A. Preimesberger
Hewlett Packard Enterprise

Sean Pope
Hewlett Packard Enterprise

Jasmit Tarang
Hewlett Packard Enterprise

A method of providing meaningful, self-descripting metadata to REST return JSON data

This document describes a method where REST data can be wrapped in a well-defined metadata format which enables convenient self-documenting return values and also enables programmatically discovering and utilizing new REST service entry points.

In the REST world – the currently popular solution is OData – which provides a structured data definition through the return data itself. The information can be used to help parse and interpret the return data – but has the weakness in that it is mostly a schema format that is intended to aid in parsing the data for static languages, such as using Jackson under Java. The data returns allows automated parsing for these languages but does not assist coders and smart clients of the data. This document describes a format, called 'crested JSON' which describes a format friendlier to developers and automated discovery routines.

- Crested JSON is a simple structured JSON spec intended for use as REST responses.
- All Crested JSON must contain exactly 2 fields at the outer level; data, metadata

Crested JSON example 0:

```
{
"data": {},
"metadata": {
"resource": "/example",
"description": "An example resource for Crested JSON",
"parent": null,
"children": {
"1": {
"resource": "/example/1"
}
}
}
}
```

Crested JSON example 1:

```
{
"data": {
"item": {
"text": "example 1"
}
},
"metadata": {
"resource": "/example/1",
"description": "Example resource 1",
"parent": {
"resource": "/example",
"description": "An example resource for Crested JSON",
"parent": null,
"children": {
"1": {
"resource": "/example/1"
}
}
},
"children": {}
}
}
```

The object at the "data" field (referred to as the data object) stores the primary data presented to the client. The value of the "data" field must not be null. The data object may be empty.

The data object may include the following optional fields; item, inventory, error.

data object example 0:

```
{
"inventory": [
{
"text": "example",
"type": "example"
},
{
"text": "example2",
"type": "example"
}
],
}

data object example 1:
{
"item": {
"text": "example 1"
},
"errors": [
{
"message": "example error"
}
]
}
```

The optional "item" field, if present, must hold a JSON object (referred to as the item object) that must not be empty (must contain at least one field).

The intended use of the item object is to present information about a single resource or query result.

The optional "inventory" field, if present, must hold an array (referred to as the inventory array) that must contain one or more JSON objects that must not be empty (must contain at least one field).

The intended use of the inventory array is to present a logical collection of resources or query results.

The optional "error" field, if present, must hold an array (referred to as the error array) that must contain one or more JSON objects that must not be empty (must contain at least one field).

The intended use of the error array is to present information about errors that occurred while performing the requested operation.

The object at the "metadata" field (referred to as the metadata object) stores navigational and contextual metadata presented to the client.

The metadata object must contain exactly 4 fields; resource, parent, children, description.

metadata object example 0:

```
{
"resource": "/example",
"description": "An example resource for Crested JSON",
"parent": null,
"children": {
"1": {
"resource": "/example/1"
}
}
}
```

metadata object example 1:

```
{
"resource": "/example/1",
"description": "Example resource 1",
"parent": {
"resource": "/example",
"description": "An example resource for Crested JSON",
"parent": null,
"children": {
"1": {
"resource": "/example/1"
}
}
},
"children": {}
}
```

The mandatory "resource" field must hold a string that is the relative path of the request given for the current resource.

The mandatory "description" field must hold a string that is the description of the current resource.

The mandatory "parent" field must hold the metadata object for the parent resource or null if the current resource is the top level of the service.

The mandatory "children" field must hold a children object for resources that can be navigated to from the current resource.

children object example 0:

```
{
"1": {
"resource": "/example/1"
}
}
```

children object example 1:

```
{
"foo": {
"resource": "/rest/foo"
},
"bar" {
"description": "The resource for performing a bar"
"resource": "/rest/bar"
}
}
```

The children object must contain zero or more fields for resources. Each field must contain exactly one resource object.

The fields are intended to be the names of the resources they contain.

A resource object must contain a "resource" field that holds a string that is the relative path of that resource.

A resource object may contain a "description" field that holds a string that is the description of that resource.

<u>More examples:</u>

// Resource at someexamplesite.com/v1/rest

<u>Crested JSON response:</u>

```
{
"data": {},
"metadata": {
"resource": "/v1/rest",
"description": "Base resource for the rest service",
"parent": null,
"children": {
"foo": {
"description": "The resource to perform a foo operation",
"resource": "/v1/rest/foo"
},
"bar": {
"description": "The resource to view different bar operations",
"resource": "/v1/rest/bar"
}
}
}
}
```

// Resource at someexamplesite.com/v1/rest/foo

<u>Crested JSON response:</u>

```
{
"data": {
"item": {
"operation": "foo",
"result": 9001
}
},
"metadata": {
"resource": "/v1/rest/foo",
"description": "The resource to perform a foo operation",
"parent": {
"resource": "/v1/rest",
"description": "Base resource for the rest service",
"parent": null,
"children": {
"foo": {
"description": "The resource to perform a foo operation",
"resource": "/v1/rest/foo"
},
"bar": {
"description": "The resource to view different bar operations",
"resource": "/v1/rest/bar"
}
}
},
"children": {}
}
}
```

// Resource at someexamplesite.com/v1/rest/bar

Crested JSON response:

```
{
"data": {
"inventory": [
{
"operation": "shrubbery",
"description": "Performs the shrubbery operation",
"href": "/v1/rest/bar?operation=shrubbery"
},
{
"operation": "spam",
"description": "Performs the spam operation",
"href": "/v1/rest/bar?operation=spam"
},
]
},
"metadata": {
"resource": "/v1/rest/bar",
"description": "The resource to view different bar operations",
"parent": {
"resource": "/v1/rest",
"description": "Base resource for the rest service",
"parent": null,
"children": {
```

```
"foo": {
"description": "The resource to perform a foo operation",
"resource": "/v1/rest/foo"
},
"bar": {
"description": "The resource to view different bar operations",
"resource": "/v1/rest/bar"
}
}
},
"children": {}
}
}
```

// Result from GET request to someexamplesite.com/v1/rest/bar?operation=shrubbery

Crested JSON response:

```
{
"data": {
"item": {
"result": {
"quote": "Ni!",
"orator": "Knights who say 'Ni'"
}
},
"inventory": [
{
"operation": "shrubbery",
"description": "Performs the shrubbery operation",
"href": "/v1/rest/bar?operation=shrubbery"
},
{
"operation": "spam",
"description": "Performs the spam operation",
"href": "/v1/rest/bar?operation=spam"
},
]
},
"metadata": {
"resource": "/v1/rest/bar",
"description": "The resource to view different bar operations",
"parent": {
"resource": "/v1/rest",
"description": "Base resource for the rest service",
"parent": null,
"children": {
```

```
"foo": {
"description": "The resource to perform a foo operation",
"resource": "/v1/rest/foo"
},
"bar": {
"description": "The resource to view different bar operations",
"resource": "/v1/rest/bar"
}
}
},
"children": {}
}
}
```

Disclosed by Lee A Preimesberger, Sean Pope and Jasmit Tarang, Hewlett Packard Enterprise