

# Spike: SPIKE: Data Import Component: Investigate an approach to move backend file upload functionality into a separate java library

<https://issues.folio.org/browse/STCOM-447>

## Spike goals:

- check if it is possible to define a RAML file without any endpoints just to declare traits and imports, the goal is to generate DTOs and reusable traits.
- check if it is possible to extend or inherit JSON schemas. if not demonstrate how to include or wrap JSON schemas.

- check if it is possible to define a RAML file without any endpoints just to declare traits and imports, the goal is to generate DTOs and reusable traits

Unfortunately, it is not possible to declare RAML file without any endpoint, one endpoint with at least one method should exist.

- check if it is possible to extend or inherit JSON schemas. if not demonstrate how to include or wrap JSON schemas.

It is possible to extend JSON schemas. The detailed information on how to do it described below.

Let's assume we have following **raml** file

### library.raml

```
##RAML 1.0

title: Book Library API
version: v0.1

documentation:
  - title: Introduction
    content: Automated access to books
  - title: Licensing
    content: Please respect copyrights on our books.

types:
  book: !include bookParent.json
  bookCollection: !include bookCollection.json
  errors: !include raml-util/schemas/errors.schema

traits:
  language: !include raml-util/traits/language.raml

resourceTypes:
  collection: !include raml-util/rtypes/collection.raml
  collection-item: !include raml-util/rtypes/item-collection.raml

/books:
  description: The collection of library books
  type:
    collection:
      schemaCollection: bookCollection
      schemaItem: book
      exampleCollection: !include examples/bookCollection.sample
      exampleItem: !include examples/book.sample
```

The definition of the book collection looks similar to other collections used in project

### bookCollection.json

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "description": "Collection of books",
  "properties": {
    "books": {
      "type": "array",
      "items": {
        "type": "object",
        "$ref": "bookChild.json"
      }
    },
    "totalRecords": {
      "type": "integer"
    }
  },
  "required": [
    "books",
    "totalRecords"
  ]
}
```

Here, for the reference field I have used "bookChild.json" file, which is a child of other json file

### bookParent.json

```
{
  "$schema": "http://json-schema.org/draft-04
/schema#",
  "type": "object",
  "description": "A book parent",
  "additionalProperties": false,
  "properties": {
    "id": {
      "type": "string"
    },
    "name": {
      "description": "The name of the book",
      "type": "string"
    },
    "author": {
      "description": "The author of the book",
      "type": "string"
    },
    "metadata": {
      "description": "Metadata about creation and
changes, provided by the server (client should not
provide)",
      "type": "object",
      "$ref": "raml-util/schemas/metadata.schema",
      "readonly": true
    }
  },
  "required": [
    "name",
    "author"
  ]
}
```

### bookChild.json

```
{
  "$schema": "http://json-schema.org/draft-04
/schema#",
  "type": "object",
  "description": "A book child",
  "additionalProperties": false,
  "extends": {
    "$ref": "bookParent.json"
  },
  "properties": {
    "publisher": {
      "description": "The publisher of the book",
      "type": "string"
    }
  }
}
```

The main point is to use

```
"extends" : {
  "$ref" : "<json_file>"
}
```

After project build we will have following models:

### BookParent

```
package org.folio.rest.jaxrs.model;

import javax.validation.Valid;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Null;
import com.fasterxml.jackson.annotation.
JsonInclude;
import com.fasterxml.jackson.annotation.
JsonProperty;
import com.fasterxml.jackson.annotation.
JsonPropertyDescription;
import com.fasterxml.jackson.annotation.
JsonPropertyOrder;

/**
 * A book parent
 *
 */
@JsonInclude(JsonInclude.Include.NON_NULL)
@JsonPropertyOrder({
  "id",
  "name",
  "author",
  "metadata"
})
public class BookParent {

    @JsonProperty("id")
    private String id;
    /**
     * The name of the book
     * (Required)
     *
     */
    @JsonProperty("name")
    @JsonPropertyDescription("The name of the
book")
    @NotNull
    private String name;
    /**
     * The author of the book
     * (Required)
     *
     */
    @JsonProperty("author")
    @JsonPropertyDescription("The author of the
book")
    @NotNull
    private String author;
    /**
     * Metadata Schema
     * <p>
     * Metadata about creation and changes to
records, provided by the server (client should not
provide)
     *
     */
}
```

### BookChild

```
package org.folio.rest.jaxrs.model;

import com.fasterxml.jackson.annotation.
JsonInclude;
import com.fasterxml.jackson.annotation.
JsonProperty;
import com.fasterxml.jackson.annotation.
JsonPropertyDescription;
import com.fasterxml.jackson.annotation.
JsonPropertyOrder;

/**
 * A book child
 *
 */
@JsonInclude(JsonInclude.Include.NON_NULL)
@JsonPropertyOrder({
  "publisher"
})
public class BookChild
    extends BookParent
{

    /**
     * The publisher of the book
     *
     */
    @JsonProperty("publisher")
    @JsonPropertyDescription("The publisher of the
book")
    private String publisher;

    /**
     * The publisher of the book
     *
     */
    @JsonProperty("publisher")
    public String getPublisher() {
        return publisher;
    }

    /**
     * The publisher of the book
     *
     */
    @JsonProperty("publisher")
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }

    public BookChild withPublisher(String
publisher) {
        this.publisher = publisher;
        return this;
    }

    @Override
```

```

    @JsonProperty("metadata")
    @JsonPropertyDescription("Metadata about
creation and changes to records, provided by the
server (client should not provide)")
    @Null
    @Valid
    private Metadata metadata;

    @JsonProperty("id")
    public String getId() {
        return id;
    }

    @JsonProperty("id")
    public void setId(String id) {
        this.id = id;
    }

    public BookParent withId(String id) {
        this.id = id;
        return this;
    }

    /**
     * The name of the book
     * (Required)
     */
    @JsonProperty("name")
    public String getName() {
        return name;
    }

    /**
     * The name of the book
     * (Required)
     */
    @JsonProperty("name")
    public void setName(String name) {
        this.name = name;
    }

    public BookParent withName(String name) {
        this.name = name;
        return this;
    }

    /**
     * The author of the book
     * (Required)
     */
    @JsonProperty("author")
    public String getAuthor() {
        return author;
    }

    /**
     * The author of the book
     * (Required)
     */
    @JsonProperty("author")
    public void setAuthor(String author) {
        this.author = author;
    }

    public BookParent withAuthor(String author) {
        this.author = author;
        return this;
    }

```

```

    public BookChild withId(String id) {
        super.withId(id);
        return this;
    }

    @Override
    public BookChild withName(String name) {
        super.withName(name);
        return this;
    }

    @Override
    public BookChild withAuthor(String author) {
        super.withAuthor(author);
        return this;
    }

    @Override
    public BookChild withMetadata(Metadata
metadata) {
        super.withMetadata(metadata);
        return this;
    }
}

```

```
    }

    /**
     * Metadata Schema
     * <p>
     * Metadata about creation and changes to
records, provided by the server (client should not
provide)
     *
     */
    @JsonProperty("metadata")
    public Metadata getMetadata() {
        return metadata;
    }

    /**
     * Metadata Schema
     * <p>
     * Metadata about creation and changes to
records, provided by the server (client should not
provide)
     *
     */
    @JsonProperty("metadata")
    public void setMetadata(Metadata metadata) {
        this.metadata = metadata;
    }

    public BookParent withMetadata(Metadata
metadata) {
        this.metadata = metadata;
        return this;
    }
}
```