

# Generating OpenAPI Specifications for Advanced Input and Output Scenarios

---



**Kevin Dockx**

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



# Coming Up



**Describing media types**

**Varying schema variation by media type**  
`IOperationFilter`



“application/json” tells us something about the *format* of the representation, but nothing about the actual *type*.

The case for vendor-specific media types



# Content Negotiation with Vendor-specific Media Types

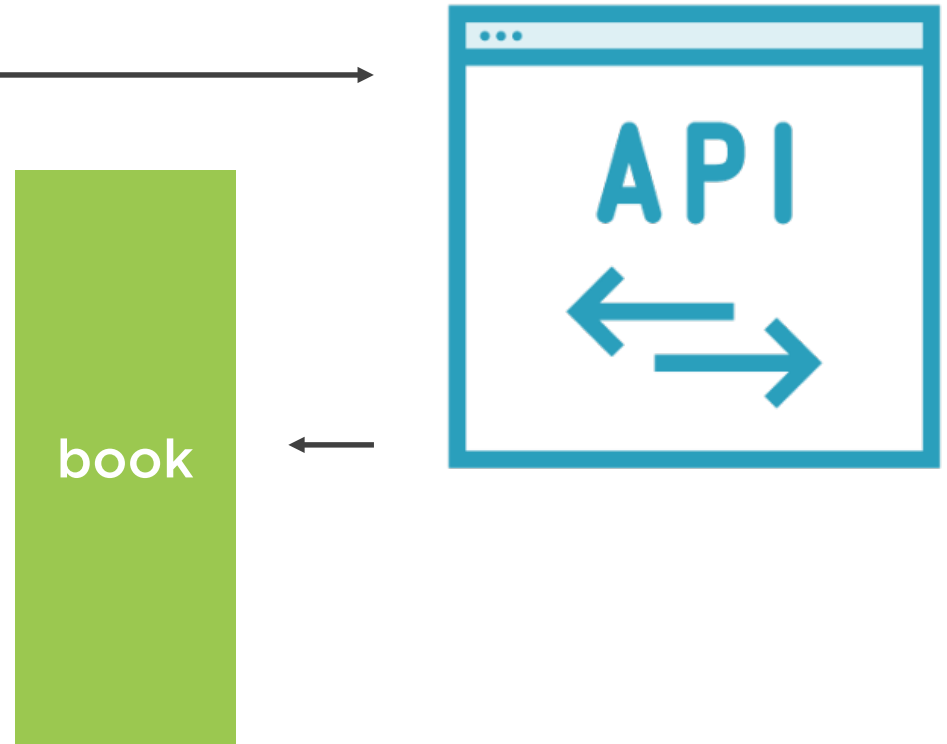
GET api/authors/{authorId}/books/{bookId}

← application/json?

```
{  
  "authorFirstName": "Stephen",  
  "authorLastName": "Fry",  
  ...  
}
```

← application/json?

```
{  
  "authorName": "Stephen Fry",  
  ...  
}
```



# Content Negotiation with Vendor-specific Media Types

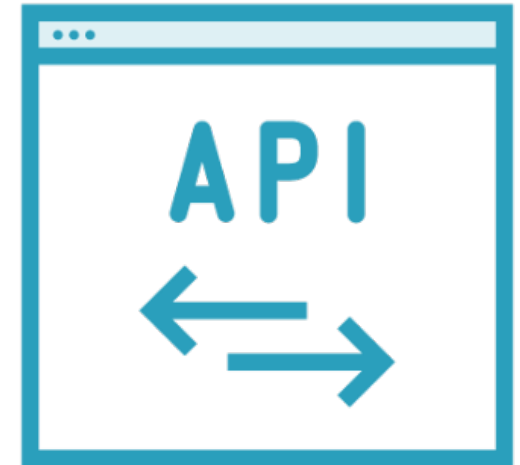
GET api/authors/{authorId}/books/{bookId}

**application/json**  
**application/vnd.marvin.book+json**

{  
 "authorFirstName": "Stephen",  
 "authorLastName": "Fry",  
 ...  
}

**application/vnd.marvin.  
bookwithconcatenatedauthorname+json**

{  
 "authorName": "Stephen Fry",  
 ...  
}



# Demo



## Supporting vendor-specific media types



```
"responses": {
  "200": {"description": "Returns the requested book",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      },
      "application/xml": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      }
    }
  }
}, ...
```

---

## OpenAPI Support for Schema Variation by Media Type (Output)



```
"responses": {
  "200": {"description": "Returns the requested book",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/Book"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      },
      "application/xml": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      }
    }
  },
  ...
}
```

---

## OpenAPI Support for Schema Variation by Media Type (Output)





```
"responses": {
  "200": {"description": "Returns the requested book",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/Book"
        }
      },
      "application/vnd.marvin.book+json": {
        "schema": {
          "$ref": "#/components/schemas/Book"
        }
      },
      "application/xml": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      }
    }
  }
}, ...
```

---

## OpenAPI Support for Schema Variation by Media Type (Output)



```
"responses": {
  "200": {"description": "Returns the requested book",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/Book"
        }
      },
      "application/vnd.marvin.book+json": {
        "schema": {
          "$ref": "#/components/schemas/Book"
        }
      },
      "application/vnd.marvin.bookwithconcatenatedauthorname+json": {
        "schema": {
          "$ref": "#/components/schemas/BookWithConcatenatedAuthorName"
        }
      }
    }
  }, ...
}
```

---

## OpenAPI Support for Schema Variation by Media Type (Output)

**Supported since OpenAPI 3**

**Swashbuckle doesn't support this out of the box at the moment of recording**



# Demo



Supporting schema variation by media type (output, ResolveConflictingActions)



# Demo



Supporting schema variation by media type (output, IOperationFilter)



# OpenAPI Support for Schema Variation by Media Type (Input)

**POST** api/authors/{authorId}/books

---

```
{  
  "title": "Mythos",  
  "description": "A collection of myths and ...",  
  ...  
}
```



# OpenAPI Support for Schema Variation by Media Type (Input)

**POST api/authors/{authorId}/books**

---

```
{  
  "title": "Mythos",  
  "description": "A collection of myths and ...",  
  ...  
}
```

application/json?

```
{  
  "title": "Mythos",  
  "description": "A collection of myths and ...",  
  "amountOfPages": 412,  
  ...  
}
```

application/json?



# OpenAPI Support for Schema Variation by Media Type (Input)

POST api/authors/{authorId}/books

---

```
{  
  "title": "Mythos",  
  "description": "A collection of myths and ...",  
  ...  
}
```

application/json

application/vnd.marvin.bookforcreation+json

```
{  
  "title": "Mythos",  
  "description": "A collection of myths and ...",  
  "amountOfPages": 412,  
  ...  
}
```

application/vnd.marvin

.bookforcreationwithamountofpages+json



```
"requestBody": {  
  "content": {  
    "application/json": {  
      "schema": {  
        "$ref": "#/components/schemas/BookForCreation" }},  
    "application/vnd.marvin.bookforcreation+json": {  
      "schema": {  
        "$ref": "#/components/schemas/BookForCreation" }},  
    "application/vnd.marvin.bookwithamountofpages+json": {  
      "schema": {  
        "$ref": "#/components/schemas/BookForCreationWithAmountOfPages" }}}}, ...
```

---

## OpenAPI Support for Schema Variation by Media Type (Input)

**Media types as children of the content tag of a request body**





# Demo



Supporting schema variation by media type (input)



```
POST api/authors/{authorId}/books  
Content-Type: "application/bookforcreation+json"  
Accept: "application/bookwithconcatenatedauthorname+json"
```

---

## Advanced Scenarios

**Combine Accept & Content-type headers to match your scenario**



```
GET api/authors/{authorId}/books/{bookId}  
Accept: "application/bookwithamountofpages+json"
```

---

## Advanced Scenarios

**Combine Accept & Content-type headers to match your scenario**



```
GET api/authors/{authorId}/books/{bookId}  
Accept: "application/bookwithamountofpages+xml"
```

---

## Advanced Scenarios

**Combine Accept & Content-type headers to match your scenario**



# Summary



**Use vendor-specific media types to specify the actual type of the data**

**OpenAPI 3 allows schema variation by media type**

- Use `IOperationFilter` to implement support in Swashbuckle