**Assignment 3 – Describing an API with an API description format**

**Setup**

Composer API Requirements

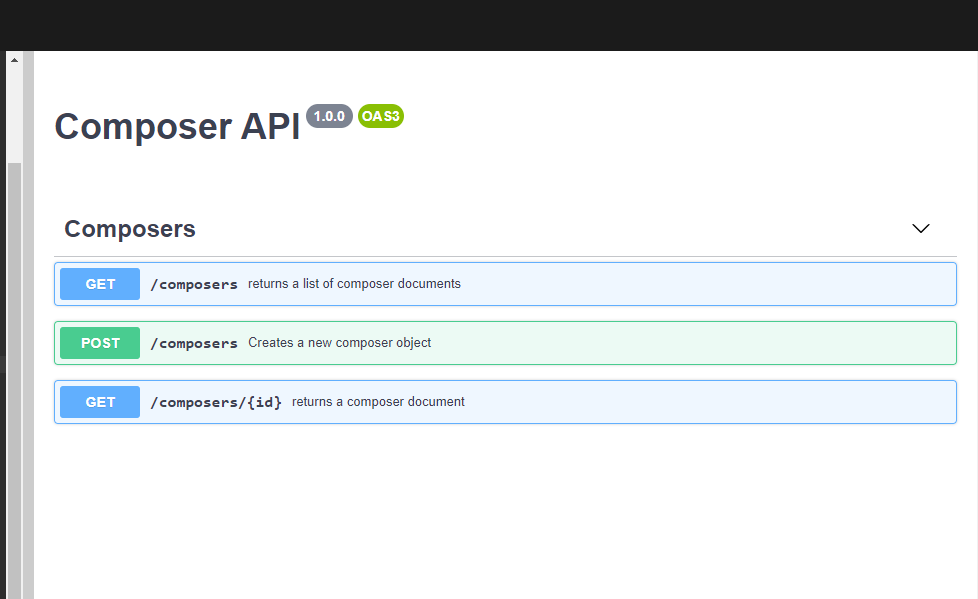
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **HTTP Protocol** | **Parameters/Request Body** | **Data Type** | **Path** |
| findAllComposers | GET | n/a | n/a | /composers |
| findComposerById | GET | id | string | /composers/{id} |
| createComposer | POST | firstName/lastName | string/string | /composers |

* ~~Group the operations by the tag “Composers.” Refer to the Reading and Videos section for assistance.~~
* Validate the file for syntax and indentation errors using the Swagger editor and YAML validator.
* ~~Include responses with descriptions for 200, 500, and 501.  Responses with 500 represent “Server Exceptions” and responses with 501 represent “MongoDB Exceptions.”~~
* ~~You must use YAML and set the openapi version to 3.0.0.  This week’s reading assignment provides several examples on how to create YAML files with GET, PUT, POST, and DELETE operations.~~
* The YAML file must be added to your GitHub repository under a directory named docs

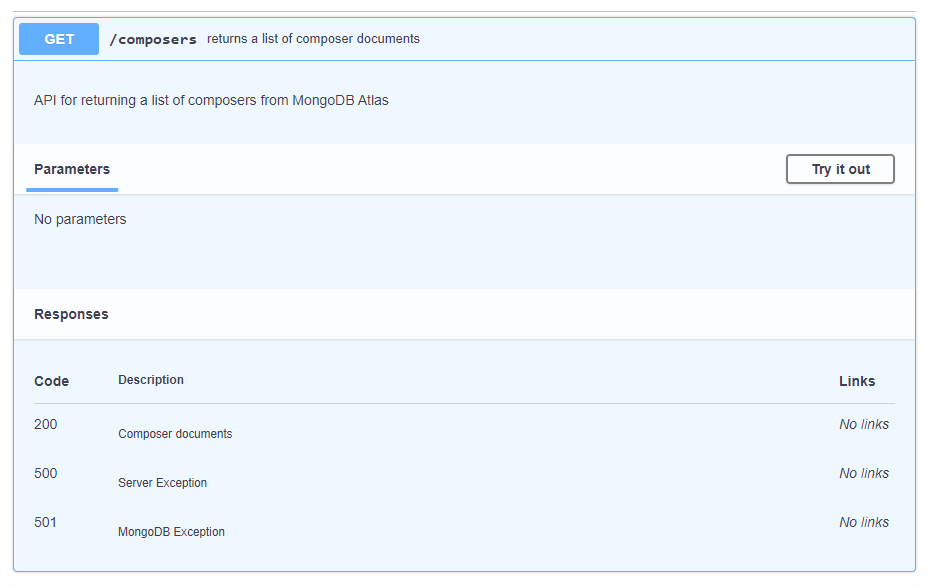
**Instructions**

* Add a new directory to GitHub repository named docs.
* Add a new file to the docs directory and name it **<yourLastName>-composers.yaml.**
* Using what you learned from this week’s reading assignment and what you explored during week’s one and two, describe the operations for the Composer API in the newly created YAML file.  For help, use the Swagger [editor](https://editor.swagger.io/#/) and YAML [validator](https://codebeautify.org/yaml-validator).

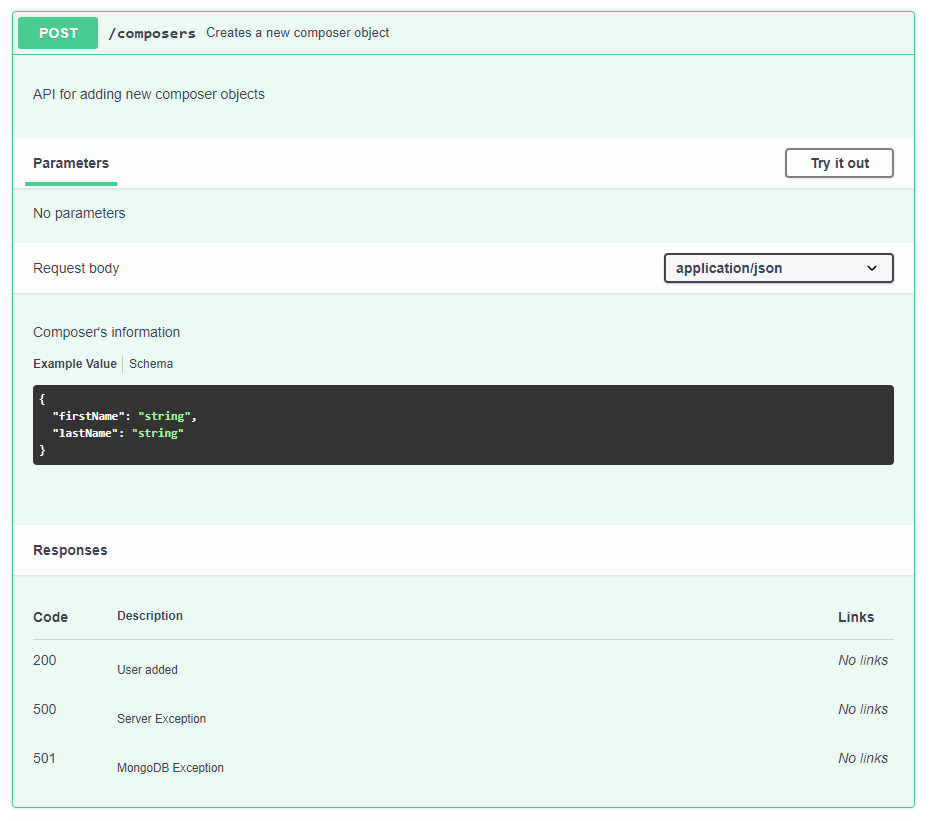
**Exhibit A. Display of the completed view from the Swagger Editor**



**Exhibit B. Display of the findAllComposers operation**



**Exhibit C. Display of the createComposer operation**



**Exhibit D. Display of the findComposerById operation**

