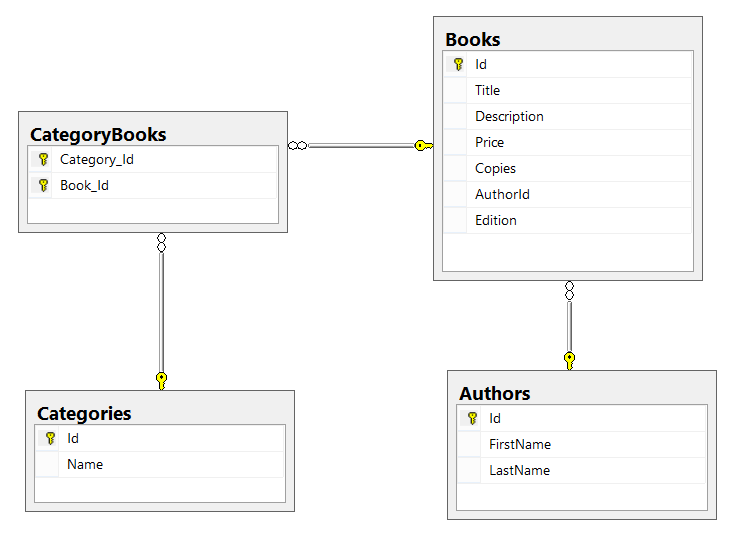
# Exercises: Web API Introduction

Problems for exercises and homework for the [“C# MVC Frameworks - ASP.NET” course @ SoftUni](https://softuni.bg/trainings/1537/csharp-mvc-frameworks-asp-net-march-2017).

## Book Shop Service

Create a REST Service with Web API for accessing a **book shop**. The shop should hold **books**, **authors** and **categories**.



Use the [**Book Shop Code First lab**](https://softuni.bg/downloads/svn/DB-Fundamentals/DB-Advanced-EntityFramework/Oct-2016/05.%20DB-Advanced-EntityFramework-EntityFramework-Relations/BookShopSystem-Solution.zip) from the **Database Applications** course as a basis for your service. Check the **Individual Accounts** authenticationwhen creating your project.

### Endpoints

Define the following endpoints in your service project. Do **NOT** use scaffolding (auto-generating endpoints)!

|  |  |  |
| --- | --- | --- |
| **Http  Method** | **Endpoint** | **Description** |
| GET | /api/authors/{id} | Gets author with **id**, **first** **name**, **last** **name** and a list of all his/her **book** **titles**. |
| POST | /api/authors | Creates a new author with **first name** and **last name** (mandatory). |
| GET | /api/authors/{id}/books | Gets books from author by id. Returns **all data about the book** + **category names**. |
| GET | /api/books/{id} | Gets data about a book by id. Returns **all data about the book** + **category names** + **author name** and **id**. |
| GET | /api/books?search={word} | Gets **top 10 books** which contain the given substring, sorted by **title** (ascending). Returns only the **title** and **id** of the books. |
| PUT | /api/books/{id} | Edits the book. Receives book **title**, **description**, **price**, **copies**, **edition**, **age restriction**, **release date** and **author id**. |
| DELETE | /api/books/{id} | Deletes the book. |
| POST | /api/books | Adds a new book with **title**, **description**, **price**, **copies**, **edition**, **age restriction**, **release date** and a string with space-separated **category names**. |
| GET | /api/categories | Gets all categories (**id** and **name**). |
| GET | /api/categories/{id} | Gets a category (**id** and **name**). |
| PUT | /api/categories/{id} | Edits a category **name**. Make sure no duplicates are created. |
| DELETE | /api/categories/{id} | Deletes a category. |
| POST | /api/categories | Adds a new category by **name**. Make sure no duplicates are created. |

Test the **endpoints** through [**Postman**](https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop).

### Requirements

* Separate your **models**, **data layer** and **services layer** into **separate** **projects**.
* Define separate controllers for **books**, **authors** and **categories**.
* Return proper **status codes** with the data.
* Use **binding models** (validated with attributes) for receiving data and **view models** for returning data.
* \* Enable **OData** for all **GET actions** where there's a collection returned.