



# .NET Core Azure Sprint **BOOK PARTNER PORTAL**



Document Revision History

Date	Revision No.	Author	Summary of Changes
20-06-2023	1.0	Ashwini Thorat	Case Study Created



## Table of Contents

<b>Introduction 4</b>	
Setup Checklist	4
Instructions	4
<b>Problem Statement</b>	<b>5</b>
Objective	5
Project structure	7
Implementation	9
Summary of the functionality to be built:	9

## INTRODUCTION

---

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Book Partner Portal. This document contains the requirements, workflow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the .NET LOT.

## SETUP CHECKLIST

### Minimum System Requirements

- Intel Core and Windows 10 and above
- Memory 8 GB
- Edge / Chrome browser
- SQL Server 2019 and SQL Server Management Studio
- Visual Studio 2022
- Visual Studio Code
- Git

## INSTRUCTIONS

- The code modules in the Sprint should follow all the coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory Sprint. Store your Project here.
- High Level Design document should be created (HLD).
- You can refer to your course material.
- You may also look up the help provided in the MSDN
- Since this project work will span over week, you will need to take care of maintaining the code.

## PROBLEM STATEMENT

---

### OBJECTIVE

Development of Book Partner Portal -To offer a great user experience for authors and publisher to submit their books/titles to portal. A common interface for both roles to publish the titles/books.

### Abstract of the project:

#### 1) Publisher

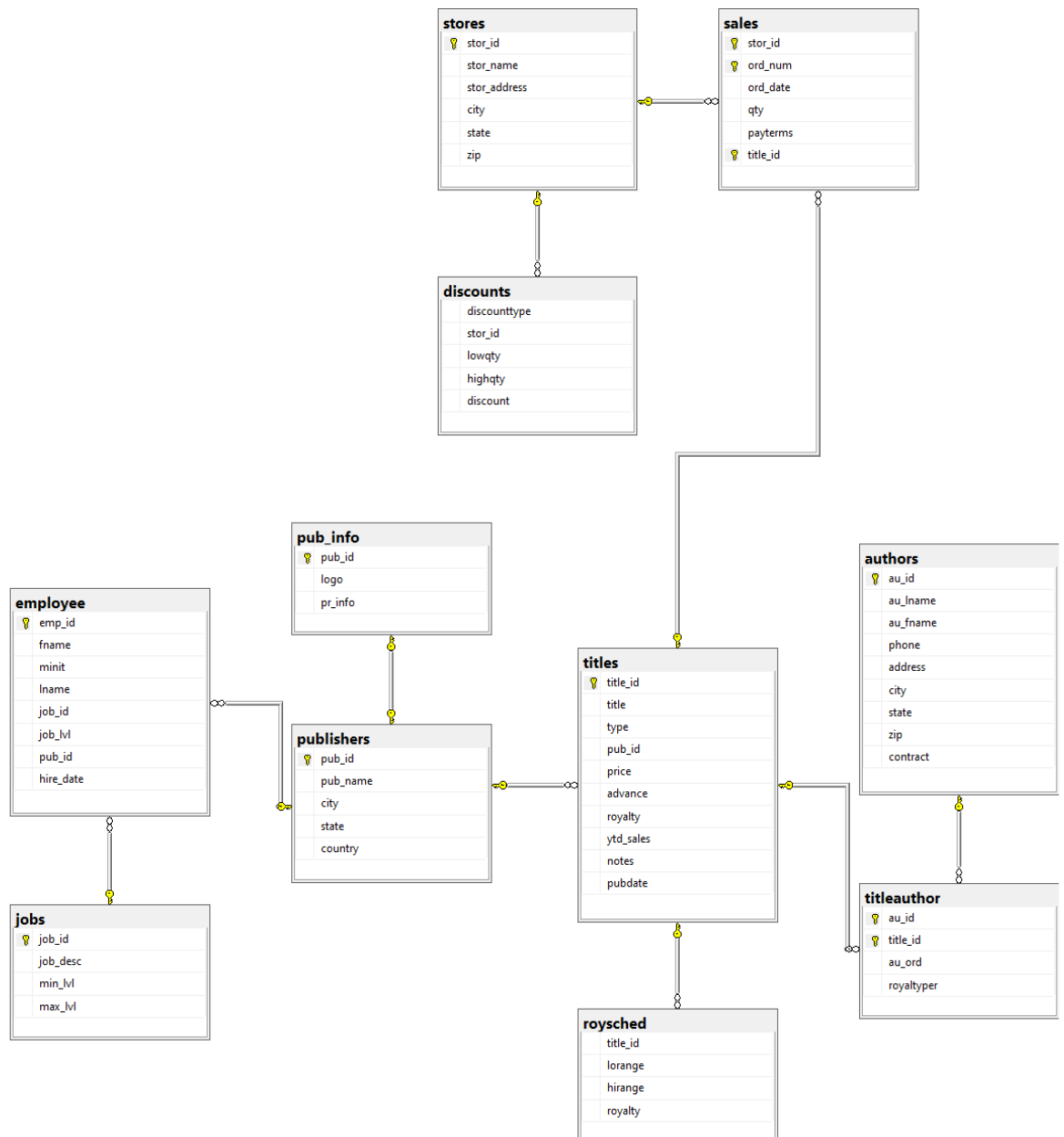
- a) Add /Update Publisher Information
- b) Approve the publish of title/book requested by author
- c) View all books
- d) View all authors
- e) View all employees

#### 2) Author

- a) Add/Update author details
- b) Add/Update title details
- c) Request the publisher to publish the title
- d) View all own titles
- e) View all publishers
- f) View all stores selling the title

### Database Schema:

<https://github.com/microsoft/sql-server-samples/blob/master/samples/databases/northwind-pubs/instpubs.sql>

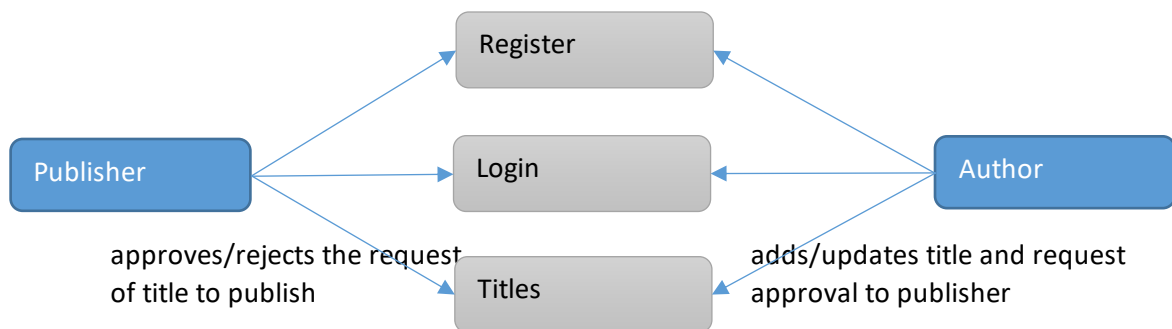


Create Client Application in Angular and Services using ASP.NET Core Web API

## MODULE LIST AND MODULE DETAILS

### Modules

- Publisher
- Author
- Login



## PROJECT STRUCTURE

The project should have minimum target framework – .NET 6.0 for Core WEB API Services and Angular 10 for UI Design



## API EXPECTATION

Create API using

- Visual Studio 2022 IDE
- .NET 6.0 / C# 5.
- Angular
- SQL Server 2019
- Entity Framework Core

## REQUIREMENT

In this sprint, you must use:

- Angular
- Use Entity Framework Core
- Create Web API and host separately



- Web App
- Azure BLOB Storage
- Azure Function
- Azure Logic App

Create Client Application in Angular and Services using Web API Core and host them on Azure Cloud.

## Design guidelines

- All the exceptions/errors to be captured and user-friendly message to be displayed on the Common Error page.
  - .NET Web API must be created using Database First Approach.
1. All Rest API request and response format must be in JSON.
  2. 200, 201, 400, 404, 401, 500, 505 – use appropriate http status code in your response object.
  3. Handle the Rest exceptions globally using Rest exception handler for all endpoints.
  4. Sample Error Responses are shown in below table, participants to identify error paths for all the Endpoints.
  5. Participants should not alter database schema under any circumstances except wherever changes are mentioned explicitly.
  6. All GET and POST requests for customer are secured
  7. Alter logo column of pub\_info to imageURL

Endpoints	HTTP Verb/Method	Description	Success Response	Error Response
/api/stores/	POST	Add new Store	String: Record Created Successfully	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/stores/	GET	Get all stores	Collections of stores	
/api/stores/name/{name}	GET	Search stores by name	Collections of stores	
/api/stores/city /{city}	GET	Search stores by city name	Collection of stores	
/api/stores/zip /{zip}	GET	Search cities by zip	Collection of stores	



## .NET CORE AZURE SPRINT BOOK PARTNER PORTAL

/api/stores/Id/{id}	GET	Search city by id	Store informaiton	
/api/stores/update /{id}	PATCH	Update the any specific details	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/stores/update/{id}	PUT	Update the whole info	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }

Endpoints	HTTP Verb/Method	Description	Success Response	Error Response
/api/authors/add	POST	Add new author	String: Record Created Successfully	
/api/authors/	GET	Get All authors	Collection of authors	
/api/authors/lname/{ln}	GET	Search Films by lastname	Collection of authors	
/api/authors/fname/{fn}	GET	Search Films by firstname	Collection of authors	
/api/authors/phone/{phno}	GET	Search Films by phone number	Author data	
/api/authors/zip/{zip}	GET	Search Films by zip	Collection of authors	
/api/authors/state/{state}	GET	Search Films by state	Collection of authors	
/api/authors/city/{city}	GET	Search Films by city	Collection of authors	
/api/authors/update/{id}	PATCH	Update the author details	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/authors/update/{id}	PUT	Update the whole info	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }

Endpoints	HTTP Verb/Method	Description	Success Response	Error Response
/api/titles/post	POST	Add new title Object in DB	String: Record Created Successfully	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }



## .NET CORE AZURE SPRINT BOOK PARTNER PORTAL

/api/authortitles/post	POST	Add new Authortitle Object in DB	String: Record Created Successfully	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/titles/	GET	Get all titles	Collection of titles	
/api/ titles/title/{title}	GET	Search titles by title	Title info	
/api/ titles/type/{type}	GET	Search titles by type	Collection of titles	
/api/ titles/pubid/{pubid}	GET	Search title by pub-id	Collection of titles	
/api/ titles/pubdate/{pubdate}	GET	Search title by pub-date	Collection of titles	
/api/ titles/authurname/{name}	GET	Search title by authurname	Collection of titles	
/api/titles/top5titles	GET	Search top 5 title by ytd	Collection of titles	
/api/titles/{id}	PUT	Update all details	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/titles/{id}	PATCH	Update specific details	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }

Endpoints	HTTP Verb/Method	Description	Success Response	Error Response
/api/publishers/post	POST	Add new publisher object to DB	String: Record Created Successfully	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/publishers/	GET	Get all publishers	Collection of publishers	
/api/publishers/pubname/{name}	GET	Search publisher by pub name	Collection of publishers	
/api/publishers/city/{city}	GET	Search publisher by pub city	Collection of publishers	
/api/publishers/state/{stat}	GET	Search publisher by pub state	Collection of publishers	



## .NET CORE AZURE SPRINT BOOK PARTNER PORTAL

/api/publishers/country/{country}	GET	Search publisher by pub country	Collection of publishers	
/api/staff/country/{country}	GET	Search Customers by Country	Collection of Staff with Address	
/api/staff/phone/{phone}	GET	Search Staff by phone number	Staff with Address	
/api/publishers/id	PUT	Update publishers info	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/publishers/id	PATCH	Update publishers info	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/pubinfo/id	PUT	Update pub info object	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/pubinfo/id	PATCH	Update pub info object	No Content 204	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }

Endpoints	HTTP Verb/Method	Description	Success Response	Error Response
/api/employees/	POST	Add new employee object to DB	String: Record Created Successfully	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/employees/	GET	Get all employees	Collections of Employees	
/api/employees/{id}	GET	Search all employees by id	Collections of Employees	
/api/employees/pubid/{id}	GET	Search all employees by pubid	Collections of Employees	
/api/employees/fname/{fname}	GET	Search all employees by fname	Collections of Employees	
/api/employees/lname/{lname}	GET	Search all employees by lname	Collections of Employees	
/api/employees/hiredate/{hiredate}	GET	Search all employees by hiredate	Collections of Employees	



Endpoints	HTTP Verb/Method	Description	Success Response	Error Response
/api/jobs/	POST	Add new job	String: Record Created Successfully	{ "timeStamp": "2023-06-08", "message": "Validation failed ..." }
/api/jobs/	GET	Get all job details	Collection of jobs	
/api/jobs/{id}	GET	Get job details by id	Job object	
/api/jobs/minlvl/{minlvl}	GET	Get jobs by minlvl	Collection of jobs	
/api/jobs/maxlvl/{maxlvl}	GET	Get jobs by maxlvl	Collection of jobs	

The information contained in this document is proprietary and confidential. For Capgemini only. | 13 / 9

**SUMMARY OF THE FUNCTIONALITY TO BE BUILT:**

The participants need to develop the Book Partner Portal by building the functionality incrementally in each of the course modules of .NET LOT.

Sr. No	Course	Duration	Functionality to be built
		(in PDs)	
1	Angular ASP.NET Core Web API Entity Framework Core SQL Server Azure services	6	Developing Angular App and Web API and hosting them on Web App (Azure App Service) Using Azure BLOB Storage, Using Logic App to validate data, Using Function app to write data to SQL Server