https://www.halvorsen.blog

ASP.NET Core CRUD Web API

Hans-Petter Halvorsen

Contents

- We will create a simple “Web API” using ASP.NET.
  - ASP.NET is a server-side framework for creating web pages and web contents.
  - We will use SQL Server as the Database system.
  - We will implement a simple CRUD Web API that Create, Read, Update and Delete data in the Database.
  - We will use Visual Studio as the Code editor.
- Finally, we will use Python and Thonny editor to test the Web API.

Introduction

Hans-Petter Halvorsen

API

- Application Programming Interface (API).
- An API is a way for two or more computer programs or components to communicate with each other.
- It is a type of software interface that offers a service to other software.
- APIs come in many shapes, some examples are SOAP API, REST API, Web API, GraphQL API, etc.
- Most programming languages today have components/libraries that can be used both to create APIs and to consume APIs (using existing APIs).

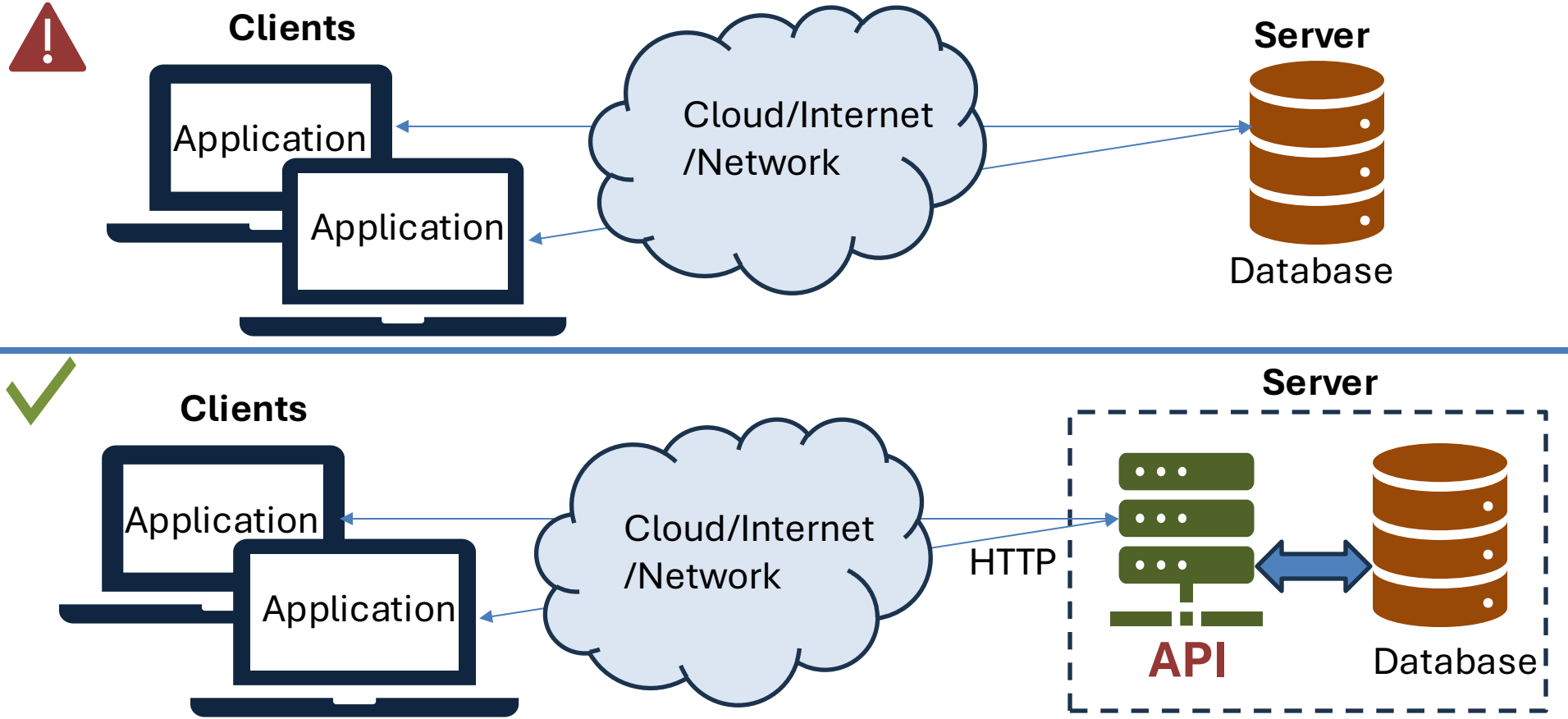
Web API

- We can create/use APIs for internal use inside an Application or between 2 or more Applications.
- Basically, an API can be just a Class with Methods that you use several places inside an Application or that you share between multiple Applications.
- A set of Stored Procedures in a Database can also be an API.
- When the Application that consume/use the API is on a local PC and the API itself is located on a Server, we can talk about so-called “Web APIs”.
- Such Web APIs also very often perform CRUD operations against a Database located on the Web.
- Note! Normally it is not allowed to connect directly to a Database located in the Cloud from a local computer unless you configure and give access to the IP addresses for those clients.

CRUD: **C**reate, **R**ead, **U**pdate, **D**elete Data

Web API

Normally it is not allowed to connect directly to a Database located in the Cloud from a local computer unless you configure and give access to the IP addresses for those clients.



JSON

- When it comes to Web APIs and REST APIs JSON is the standard for the data format.
- Example:

```
{
  "Name": "John Wayne",
  "Work": "Actor",
  "Age": 52
  "Children": [
    "Lisa",
    "Thomas",
    "Knut"
  ]
}
```

https://en.wikipedia.org/wiki/JSON

Why use Web API?

- Normally it is not allowed to connect directly to a Database located in the Cloud from a local computer
  - unless you configure and give access to the IP addresses for those clients.
  - Typically, your IT Department don't allow that
- You can use the same API for multiple Applications, let say you have a Desktop App, an iPhone App and an Android App
  - All can use the same API
  - You save time and money by developing only once instead of specific code for each application.
- You want to expose data to externals services or persons, e.g., a Weather API that can be used by persons, external apps or services. Other examples: Hotel/plane reservations and ticket systems

API Summary

- “Web APIs”, “REST APIs” or “HTTP APIs” are basically the same.
- It is more or less just different names for the same.
- They all communicate via Internet and use HTTP as communication protocol.
- And they use JSON (or sometimes XML) as Data Format.
- Very often they implement CRUD functionality.

ASP.NET Web API Example

Hans-Petter Halvorsen

Example

We will create a Web API with CRUD functionality, meaning we will insert, read, update and delete data in a Database.

- We will start by creating a Database and Table using SQL Server and SQL Server Management Studio.
- Then we will create the ASP.NET code for the Web API.
- We will test the API in the URL in the Web Browser.
- Finaly we will test the API creating some basic Python examples.

Tools

The following tool will be used in this example:

- SQL Server
  - SQL Server Management Studio
- Visual Studio
- ASP.NET
- Python
  - Thonny Python Editor

Database

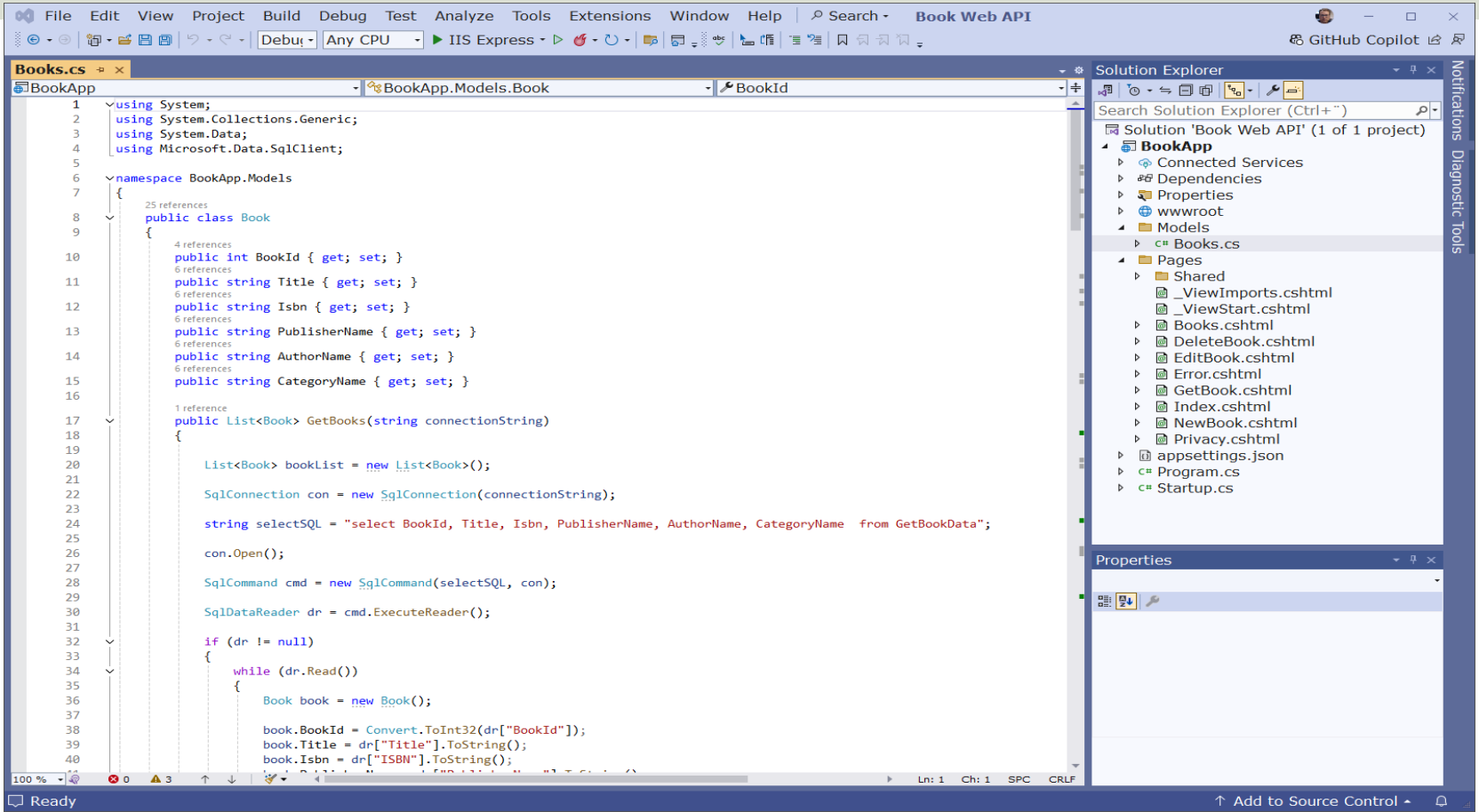
```
CREATE TABLE [AUTHOR]
(
    [AuthorId] [int] IDENTITY(1, 1) NOT
    [AuthorName] [varchar](50) NOT NULL
    [Address] [varchar](50) NULL,
    [Phone] [varchar](50) NULL,
    [PostCode] [varchar](50) NULL,
    [PostAddress] [varchar](50) NULL,
)
```

```
CREATE TABLE [PUBLISHER]
(
    [PublisherId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    [PublisherName] [varchar](50) NOT NULL UNIQUE,
    [Description] [varchar](1000) NULL,
    [Address] [varchar](50) NULL,
    [Phone] [varchar](50) NULL,
    [PostCode] [varchar](50) NULL,
    [PostAddress] [varchar](50) NULL,
    [EMail] [varchar](50) NULL,
)
```

```
CREATE TABLE [BOOK]
(
    [BookId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    [Title] [varchar](50) NOT NULL UNIQUE,
    [ISBN] [varchar](20) NOT NULL,
    [PublisherId] [int] NOT NULL F
    [AuthorId] [int] NOT NULL FORE
    [CategoryId] [int] NOT NULL FO
    [Description] [varchar](1000)
    [Year] [date] NULL,
    [Edition] [int] NULL,
    [AverageRating] [float] NULL,
)
```

```
CREATE TABLE [CATEGORY]
(
    [CategoryId] [int] IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    [CategoryName] [varchar](50) NOT NULL UNIQUE,
    [Description] [varchar](1000) NULL,
)
```

Visual Studio



Book Class

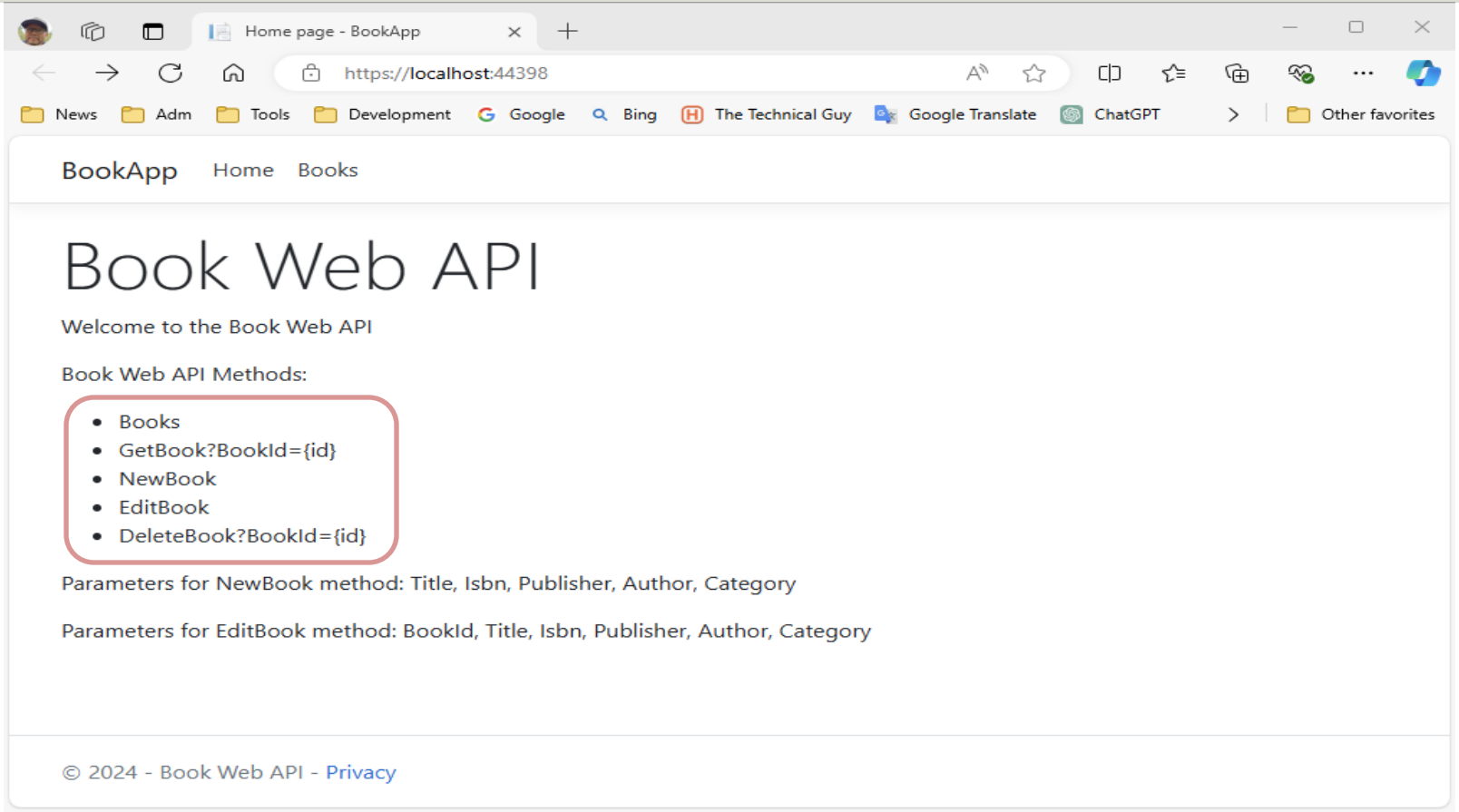
```
using System;
using System.Collections.Generic;
using System.Data;
using Microsoft.Data.SqlClient;

namespace BookApp.Models
{
    public class Book
    {
        public int BookId { get; set; }
        public string Title { get; set; }
        public string Isbn { get; set; }
        public string PublisherName { get; set; }
        public string AuthorName { get; set; }
        public string CategoryName { get; set; }

        public List<Book> GetBooks(string connectionString) {
          ..
        }

        public Book GetBookData(string connectionString, int bookId) {
          ..
        }

        public void CreateBook(string connectionString, Book book) {
          ..
        }

        public void EditBook(string connectionString, Book book)  {
          ..
        }

        public void DeleteBook(string connectionString, int bookId) {
          ..
        }

    }

}
```

API “Methods”

We create the following API CRUD methods

- Books
- GetBook?BookId={id}
- NewBook
- EditBook
- DeleteBook?BookId={id}

Running API in Visual Studio



Books

This method is used to retrieve All Books from the Database

Hans-Petter Halvorsen

GetBooks - C# Method

Book.cs

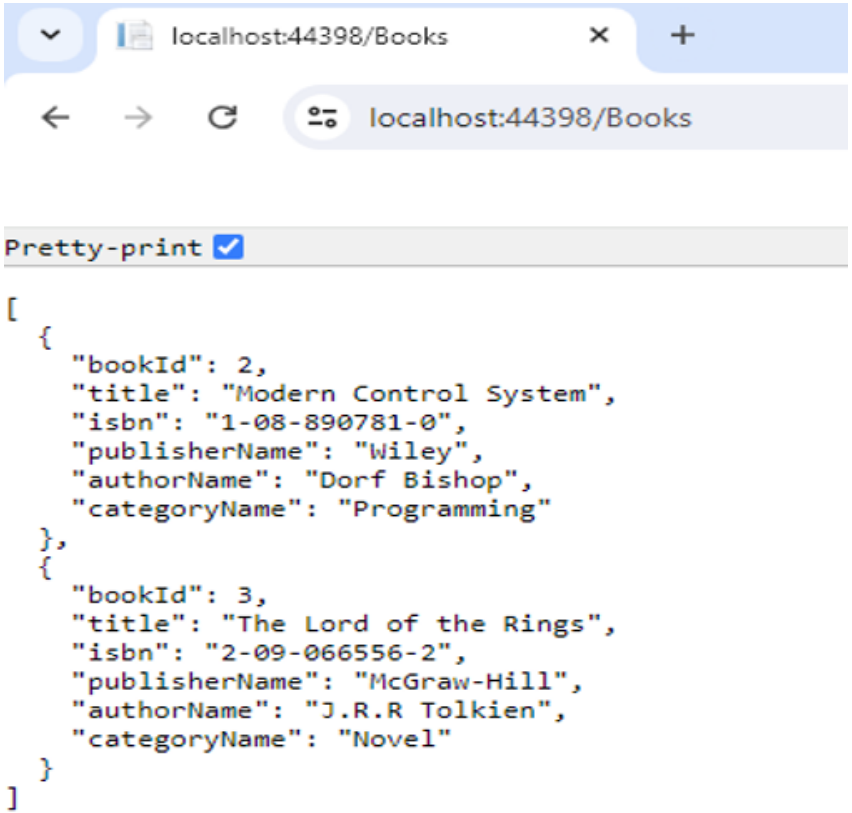
```
public List<Book> GetBooks(string connectionString)
{

    List<Book> bookList = new List<Book>();

    SqlConnection con = new SqlConnection(connectionString);
    string selectSQL = "select BookId, Title, Isbn, PublisherName, AuthorName, CategoryName  from GetBookData";
    con.Open();

    SqlCommand cmd = new SqlCommand(selectSQL, con);
    SqlDataReader dr = cmd.ExecuteReader();

    if (dr != null)
    {
        while (dr.Read())
        {
            Book book = new Book();

            book.BookId = Convert.ToInt32(dr["BookId"]);
            book.Title = dr["Title"].ToString();
            book.Isbn = dr["ISBN"].ToString();
            book.PublisherName = dr["PublisherName"].ToString();
            book.AuthorName = dr["AuthorName"].ToString();
            book.CategoryName = dr["CategoryName"].ToString();

            bookList.Add(book);
        }
    }
    return bookList;
}
```

Books API Code

Books.cshtml

```
public JsonResult OnGet()
{
    List<Book> books = new List<Book>();

    books = GetBookList();
    return new JsonResult(books);
}

private List<Book> GetBookList()
{
    connectionString = _configuration.GetConnectionString("ConnectionString");

    List<Book> bookList = new List<Book>();

    Book book = new Book();

    bookList = book.GetBooks(connectionString);

    return bookList;
}
```

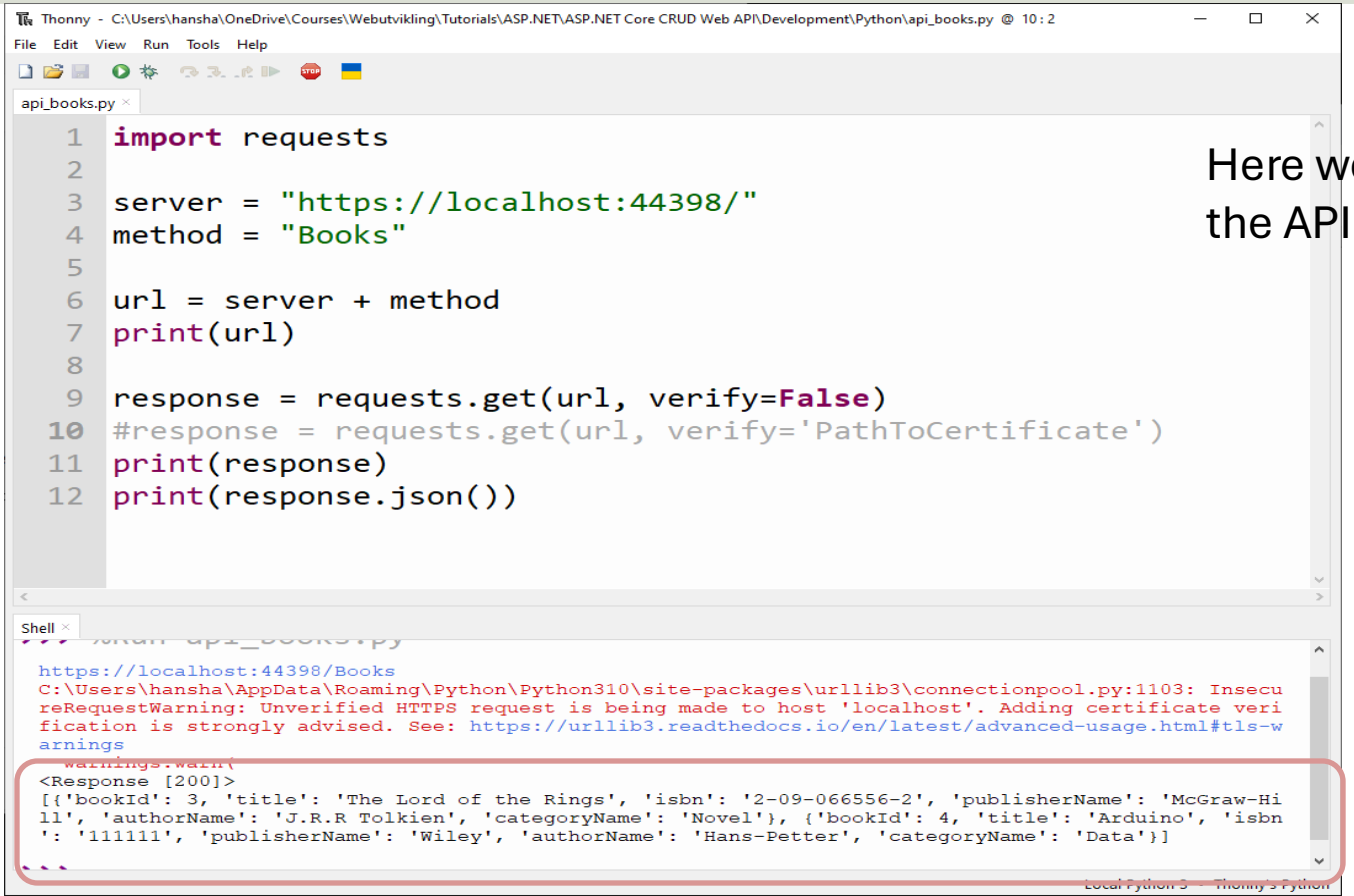
Books - Web Browser



There are 2 books in the database at the moment.

https://localhost:44398/Books

Books - Thonny



Here we use (consume) the API from Python

Books - Python

```
import requests

server = "https://localhost:44398/"
method = "Books"

url = server + method
print(url)

response = requests.get(url, verify=False)
print(response)
print(response.json())
```

GetBook

This method is used to retrieve a specific Book from the Database

Hans-Petter Halvorsen

GetBookData - C# Method

Book.cs

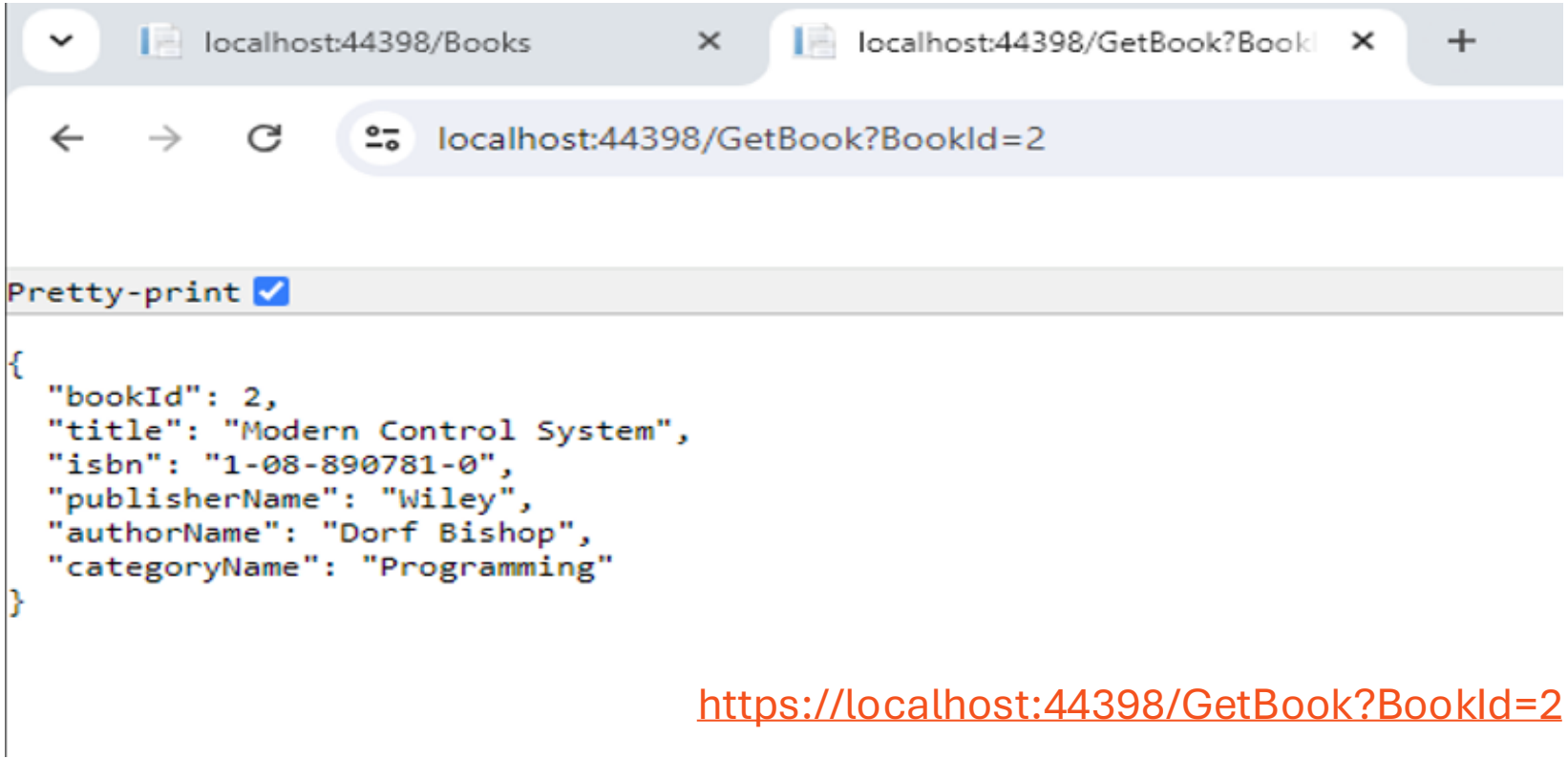
```
public Book GetBookData(string connectionString, int bookId)
{

    SqlConnection con = new SqlConnection(connectionString);
    string selectSQL = "select BookId, Title, Isbn, PublisherName, AuthorName, CategoryName  from GetBookData where BookId
                     = " + bookId;
    con.Open();
    SqlCommand cmd = new SqlCommand(selectSQL, con);
    SqlDataReader dr = cmd.ExecuteReader();

    Book book = new Book();

    if (dr != null)
    {
        while (dr.Read())
        {
            book.BookId = Convert.ToInt32(dr["BookId"]);
            book.Title = dr["Title"].ToString();
            book.Isbn = dr["ISBN"].ToString();
            book.PublisherName = dr["PublisherName"].ToString();
            book.AuthorName = dr["AuthorName"].ToString();
            book.CategoryName = dr["CategoryName"].ToString();
        }
    }
    return book;
}
```

GetBook API Code

GetBook.cshtml

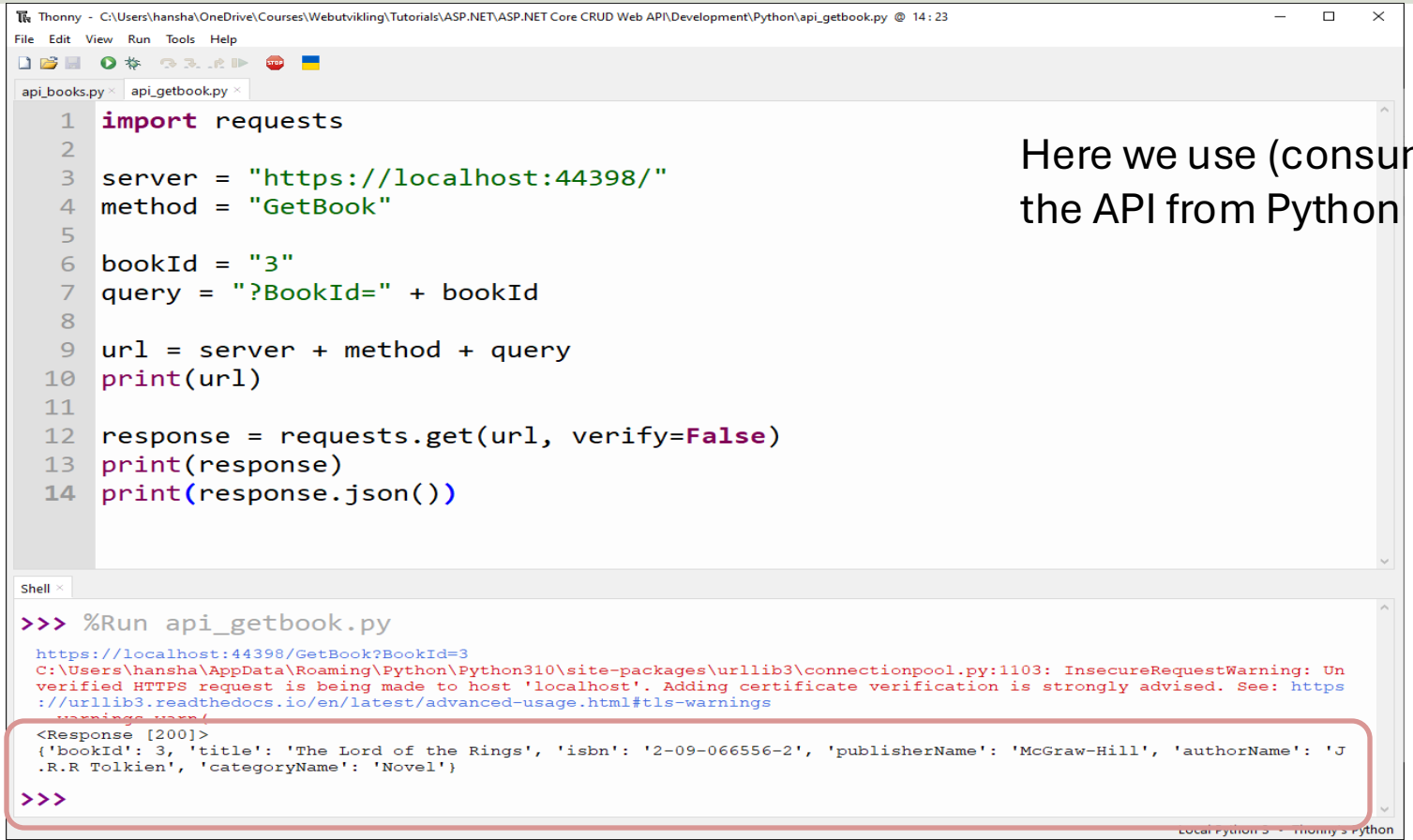
```
public JsonResult OnGet()
{
    int bookId;

    bookId = Convert.ToInt16(Request.Query["bookid"]);

    Book book = new Book();

    connectionString =
        _configuration.GetConnectionString("ConnectionString");

    book = book.GetBookData(connectionString, bookId);

    return new JsonResult(book);
}
```

GetBook - Web Browser



https://localhost:44398/GetBook?BookId=2

GetBook - Thonny



Here we use (consume) the API from Python

GetBook - Python

```
import requests

server = "https://localhost:44398/"
method = "GetBook"

bookId = "3"
query = "?BookId=" + bookId

url = server + method + query
print(url)

response = requests.get(url, verify=False)
print(response)
print(response.json())
```

NewBook

This method is used to retrieve All Books from the Database

Hans-Petter Halvorsen

CreateBook – C# Method

Book.cs

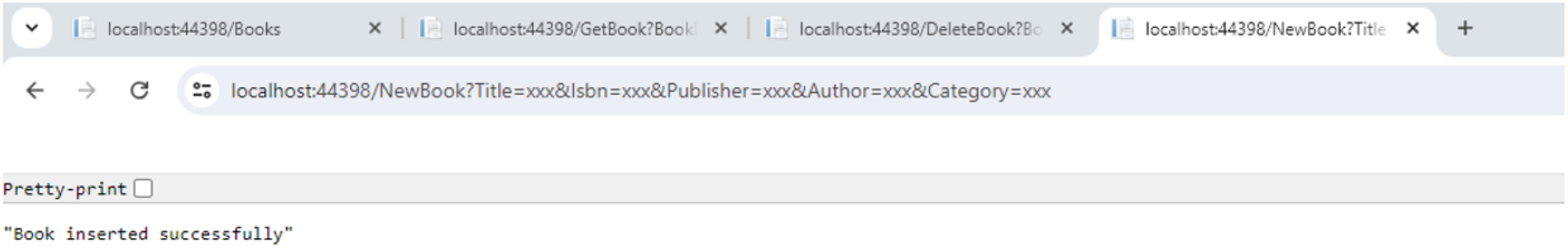
```
public void CreateBook(string connectionString, Book book)
{
    try
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            SqlCommand cmd = new SqlCommand("CreateBook", con);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.Add(new SqlParameter("@Title", book.Title));
            cmd.Parameters.Add(new SqlParameter("@Isbn", book.Isbn));
            cmd.Parameters.Add(new SqlParameter("@PublisherName", book.PublisherName));
            cmd.Parameters.Add(new SqlParameter("@AuthorName", book.AuthorName));
            cmd.Parameters.Add(new SqlParameter("@CategoryName", book.CategoryName));

            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

NewBook API Code

NewBook.cshtml

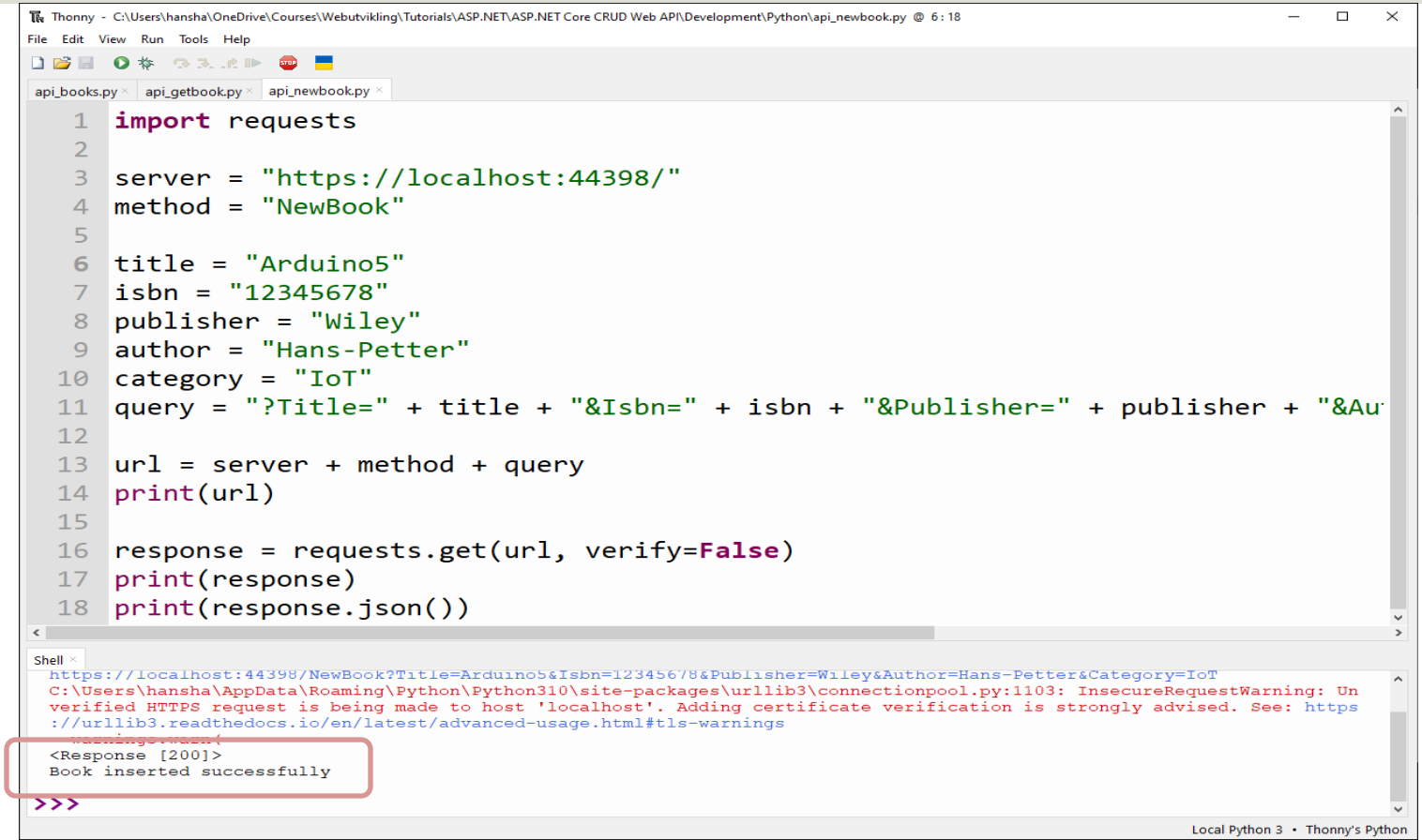
```
public JsonResult OnGet()
{
    Book book = new Book();

    book.Title = Request.Query["Title"];
    book.Isbn = Request.Query["Isbn"];
    book.PublisherName = Request.Query["Publisher"];
    book.AuthorName = Request.Query["Author"];
    book.CategoryName = Request.Query["Category"];

    connectionString =
              _configuration.GetConnectionString("ConnectionString");

    book.CreateBook(connectionString, book);

    return new JsonResult("Book inserted successfully");
}
```

NewBook – Web Browser



https://localhost:44398/NewBook?Title=xxx&Isbn=xxx&Publisher=xxx&Author=xxx&Category=xxx

NewBook - Thonny



NewBook - Python

```
import requests

server = "https://localhost:44398/"
method = "NewBook"

title = "Arduino2"
isbn = "12345678"
publisher = "Wiley"
author = "Hans-Petter"
category = "IoT"
query = "?Title=" + title + "&Isbn=" + isbn + "&Publisher=" + publisher + "&Author="
+ author + "&Category=" + category

url = server + method + query
print(url)

response = requests.get(url, verify=False)
print(response)
print(response.json())
```

EditBook

This method is used to retrieve All Books from the Database

Hans-Petter Halvorsen

EditBook – C# Method

Book.cs

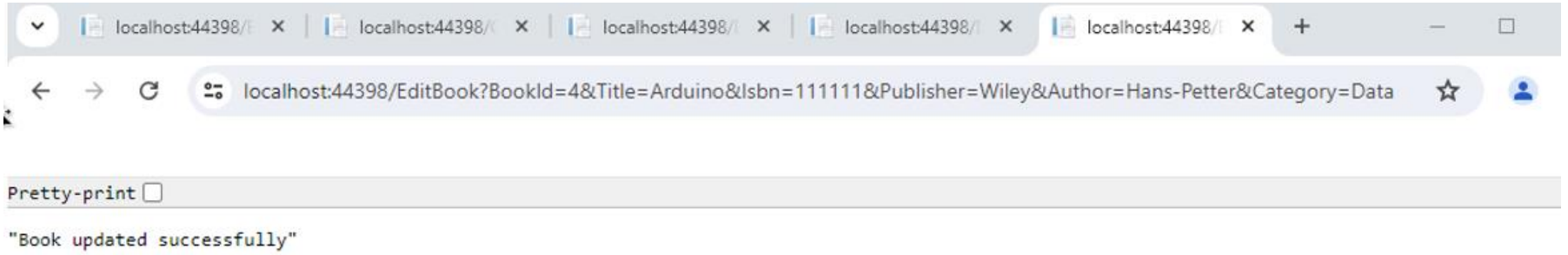
```
public void EditBook(string connectionString, Book book)
{
    try
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            SqlCommand cmd = new SqlCommand("UpdateBook", con);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.Add(new SqlParameter("@BookId", book.BookId));
            cmd.Parameters.Add(new SqlParameter("@Title", book.Title));
            cmd.Parameters.Add(new SqlParameter("@Isbn", book.Isbn));
            cmd.Parameters.Add(new SqlParameter("@PublisherName", book.PublisherName));
            cmd.Parameters.Add(new SqlParameter("@AuthorName", book.AuthorName));
            cmd.Parameters.Add(new SqlParameter("@CategoryName", book.CategoryName));

            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }

    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

EditBook API Code

EditBook.cshtml

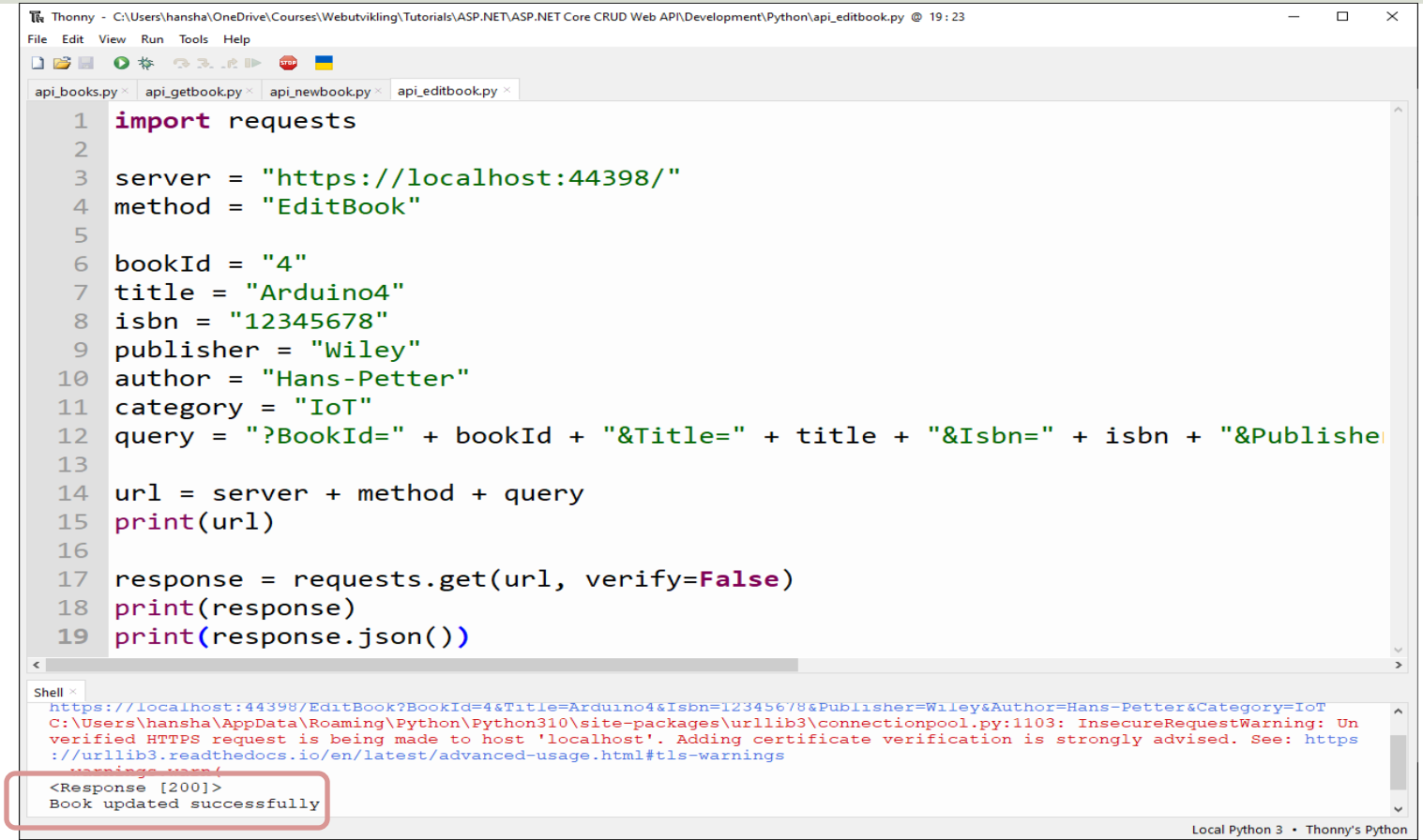
```
public JsonResult OnGet()
{
    Book book = new Book();
    book.BookId = Convert.ToInt16(Request.Query["BookId"]);
    book.Title = Request.Query["Title"];
    book.Isbn = Request.Query["Isbn"];
    book.PublisherName = Request.Query["Publisher"];
    book.AuthorName = Request.Query["Author"];
    book.CategoryName = Request.Query["Category"];

    connectionString =
           _configuration.GetConnectionString("ConnectionString");

    book.EditBook(connectionString, book);

    return new JsonResult("Book updated successfully");
}
```

EditBook – Web Browser



https://localhost:44398/EditBook?BookId=4&Title=Arduino&Isbn=111111&Publisher=Wiley&Author=Hans-Petter&Category=Data

EditBook - Thonny



EditBook - Python

```
import requests

server = "https://localhost:44398/"
method = "EditBook"

bookId = "4"
title = "Arduino"
isbn = "12345678"
publisher = "Wiley"
author = "Hans-Petter"
category = "IoT"
query = "?BookId=" + bookId + "&Title=" + title + "&Isbn=" + isbn + "&Publisher=" +
publisher + "&Author=" + author + "&Category=" + category

url = server + method + query
print(url)

response = requests.get(url, verify=False)
print(response)
print(response.json())
```

DeleteBook

This method is used to retrieve All Books from the Database

Hans-Petter Halvorsen

DeleteBook – C# Method

Book.cs

```
public void DeleteBook(string connectionString, int bookId)
{
    try
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            SqlCommand cmd = new SqlCommand("DeleteBook", con);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.Add(new SqlParameter("@BookId", bookId));

            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }

    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

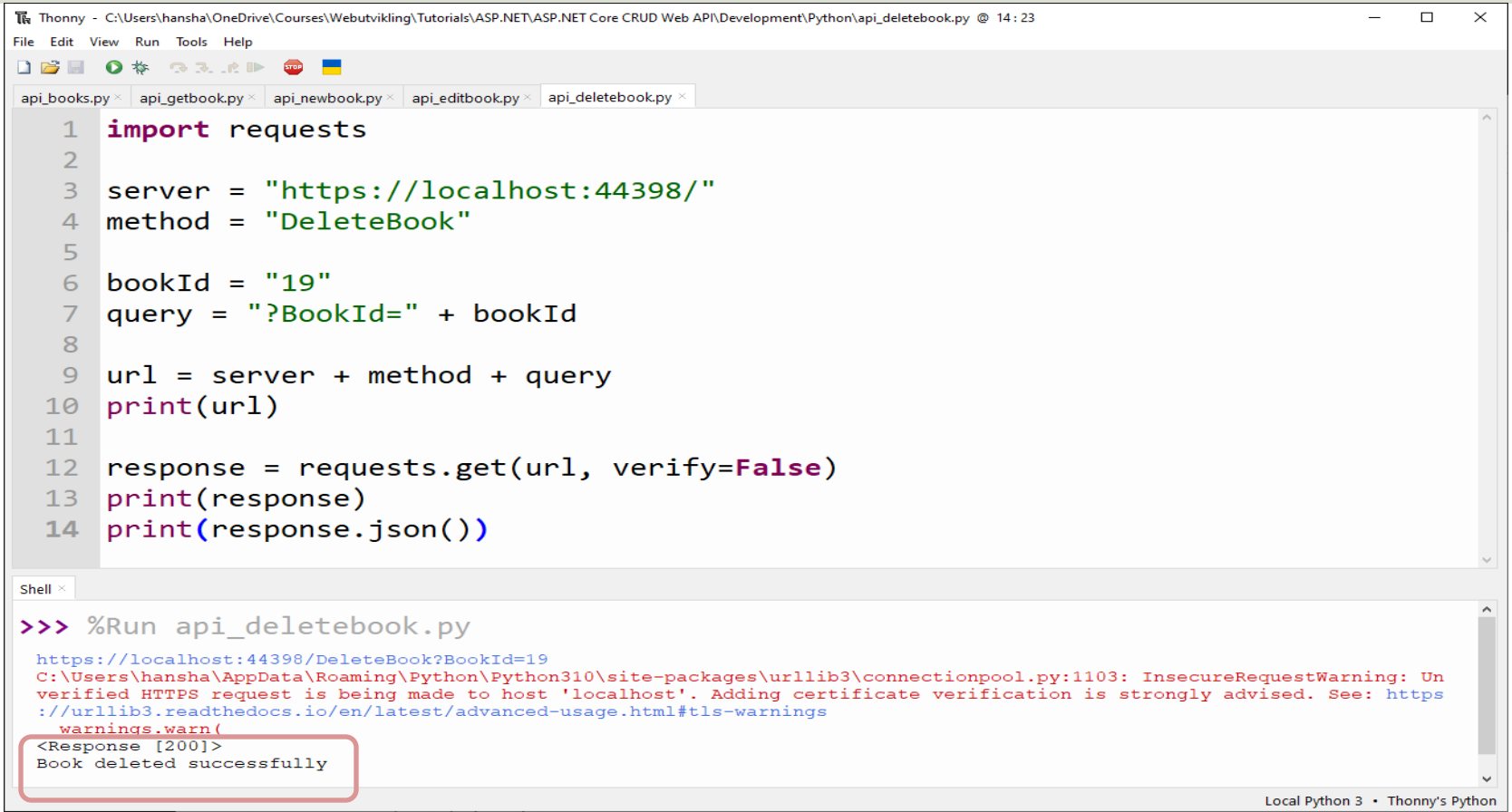
DeleteBook API Code

```
public JsonResult OnGet()
{
    int bookId;
    bookId = Convert.ToInt16(Request.Query["BookId"]);

    connectionString =
        _configuration.GetConnectionString("ConnectionString");

    Book book = new Book();
    book.DeleteBook(connectionString, bookId);

    return new JsonResult("Book deleted successfully");
}
```

DeleteBook - Web Browser

https://localhost:44398/DeleteBook?BookId=2

DeleteBook - Thonny



DeleteBook - Python

```
import requests

server = "https://localhost:44398/"
method = "DeleteBook"

bookId = "19"
query = "?BookId=" + bookId

url = server + method + query
print(url)

response = requests.get(url, verify=False)
print(response)
print(response.json())
```

Summary

Hans-Petter Halvorsen

Web API Structure

Make sure to keep a good folder structure. Sooner or later, you will have multiple APIs, and you also probably need to maintain multiple versions of the same API, e.g., like this:

- API
  - Book
    - 1.0
      - » PHP API Files
    - 2.0
      - » PHP API Files
  - Customer
    - 1.0
      - » PHP API Files
    - ..

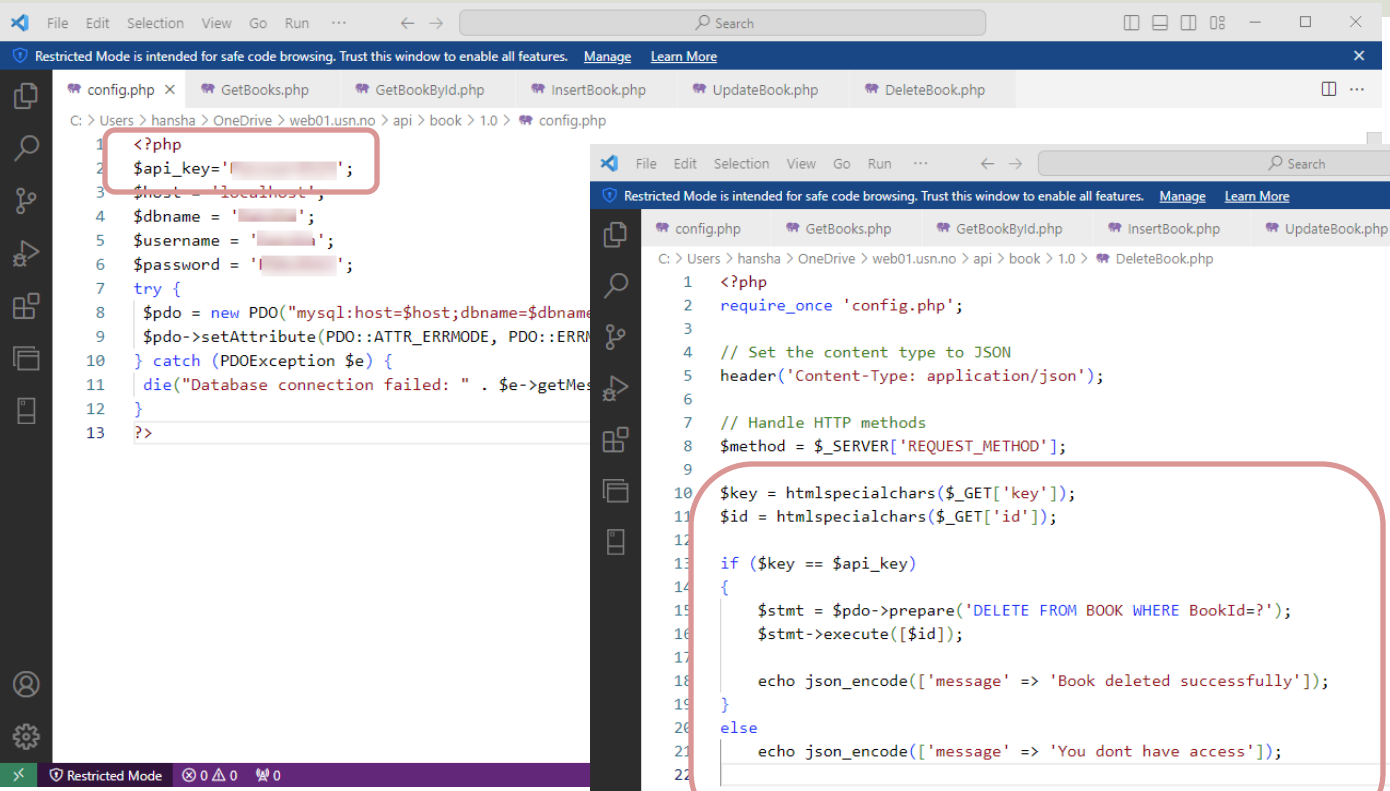
Basic Security and Access Control

- It is a good idea to add some security and access control.
- In that way only users that has access can get, insert, update or delete data.
- You then need to add a basic check in your PHP files to check if password is correct.

Example:

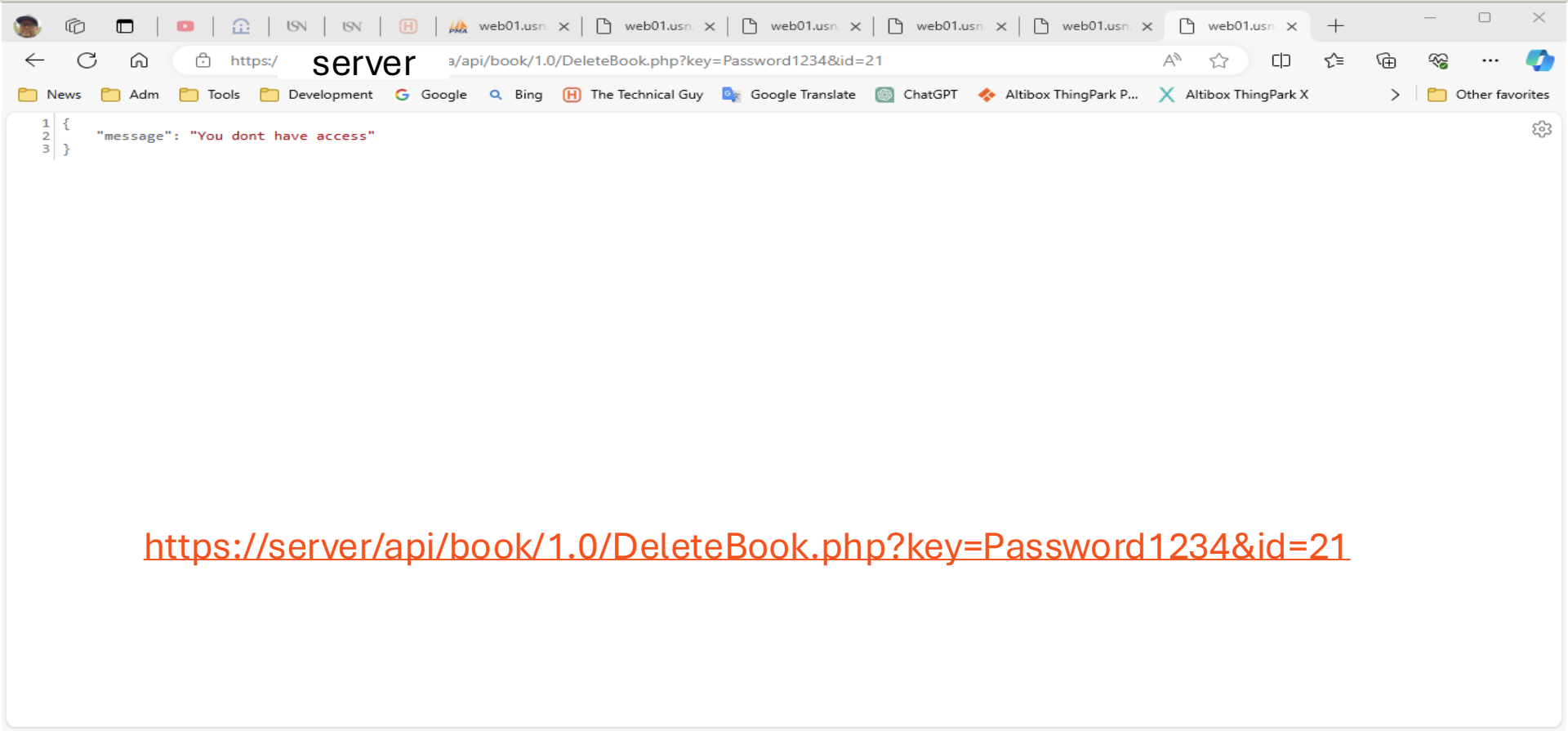
https://server/api/book/1.0/DeleteBook.php**?api_key=xxx**&id=20

PHP Example


File Edit Selection View Go Run
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
config.php
GetBooks.php
GetBookById.php
InsertBook.php
UpdateBook.php
DeleteBook.php
C: > Users > hansha > OneDrive > web01.usn.no > api > book > 1.0 > config.php
<?php
$api_key='[redacted]';
$host = 'localhost';
$dbname = '[redacted]';
$username = '[redacted]';
$password = '[redacted]';
try {
$pdo = new PDO("mysql:host=$host;dbname=$dbname
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERR
} catch (PDOException $e) {
die("Database connection failed: " . $e->getMes
}
?>
Restricted Mode 0 0 0

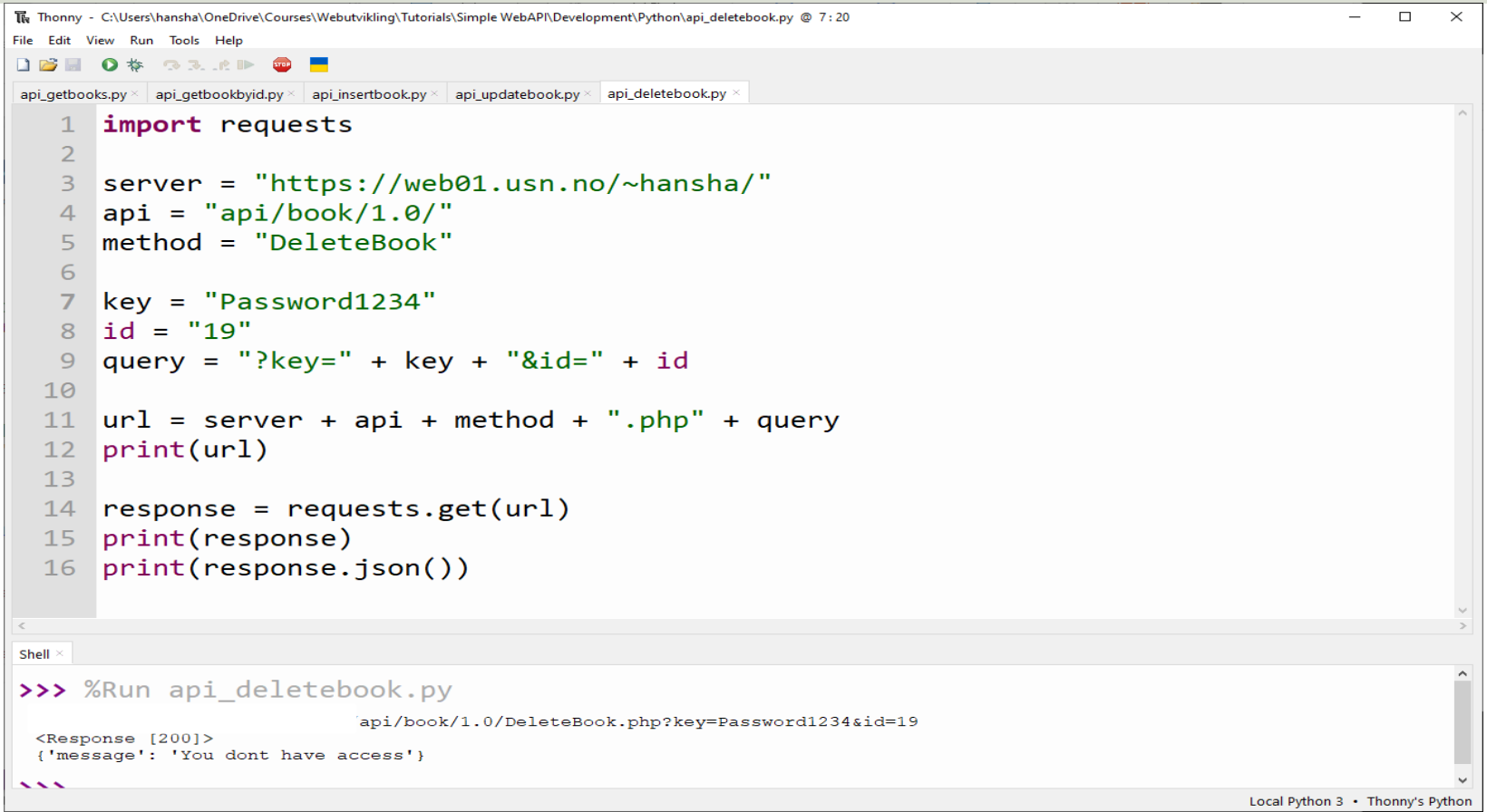
File Edit Selection View Go Run
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
config.php
GetBooks.php
GetBookById.php
InsertBook.php
UpdateBook.php
DeleteBook.php
C: > Users > hansha > OneDrive > web01.usn.no > api > book > 1.0 > DeleteBook.php
<?php
require_once 'config.php';
// Set the content type to JSON
header('Content-Type: application/json');
// Handle HTTP methods
$method = $_SERVER['REQUEST_METHOD'];
$key = htmlspecialchars($_GET['key']);
$id = htmlspecialchars($_GET['id']);
if ($key == $api_key)
{
$stmt = $pdo->prepare('DELETE FROM BOOK WHERE BookId=?');
$stmt->execute([$id]);
echo json_encode(['message' => 'Book deleted successfully']);
}
else
echo json_encode(['message' => 'You dont have access']);
?>
Restricted Mode 0 0 0
Ln 22, Col 1 Spaces: 4 UTF-8 CRLF PHP

Test in Web Browser



https://server/api/book/1.0/DeleteBook.php?key=Password1234&id=21

Python


Thonny - C:\Users\hansha\OneDrive\Courses\Webutvikling\Tutorials\Simple WebAPI\Development\Python\api_deletebook.py @ 7 : 20
File Edit View Run Tools Help
api_getbooks.py
api_getbookbyid.py
api_insertbook.py
api_updatebook.py
api_deletebook.py
1 import requests
2
3 server = "https://web01.usn.no/~hansha/"
4 api = "api/book/1.0/"
5 method = "DeleteBook"
6
7 key = "Password1234"
8 id = "19"
9 query = "?key=" + key + "&id=" + id
10
11 url = server + api + method + ".php" + query
12 print(url)
13
14 response = requests.get(url)
15 print(response)
16 print(response.json())
Shell
>>> %Run api_deletebook.py
api/book/1.0/DeleteBook.php?key=Password1234&id=19
<Response [200]>
{'message': 'You dont have access'}
>>>
Local Python 3 • Thonny's Python

Summary

- We have created a simple Web API using ASP.NET Core.
- We tested the Web API using Python.
- In general, we can use any kind of programming language to interact with this API.
  - E.g., we an create a Windows Forms Application in Visual Studio and C#.
- In that way we can insert, read, update or delete data in the remote database from a local application.
- Normally you cannot directly interact with a remote SQL Database from your local computer due to security reasons.
- There are lots of improvements to be made to make a better code structure (create classes, etc.), make it more robust with error handling, improved security, etc. But I leave that to you to improve.
- The code is made simple to illustrate the basic principles using Web APIs.

References

- PHP Tutorial: https://www.w3schools.com/php
- MySQL Tutorial: https://www.w3schools.com/mysql
- https://medium.com/@miladev95/how-to-make-crud-rest-api-in-php-with-mysql-5063ae4cc89
- Python & APIs: https://realpython.com/python-api/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog