



**FAKULTETI I
SHKENCAVE TË
NATYRËS**

**Ndërtimi i një Web API për një
Bibliotekë publike.**

Adela Cibuku

Tik A1

Programim MVC

Të krijohen 2 modele

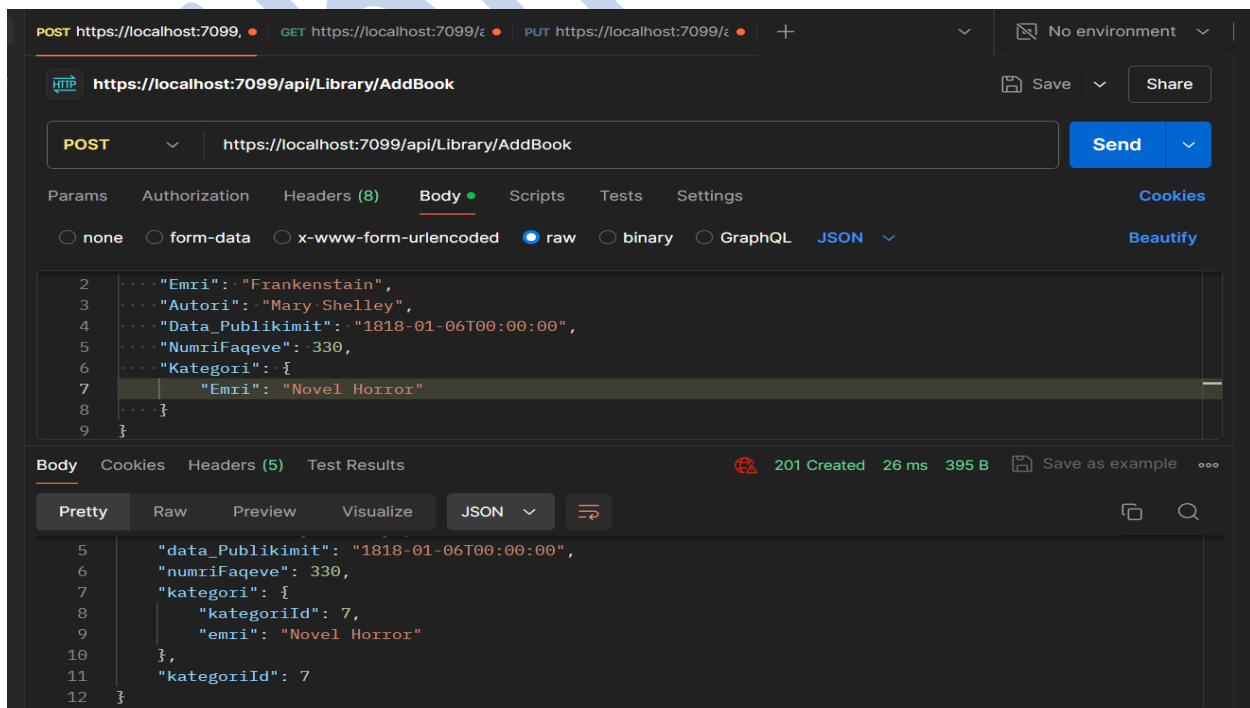
Libri -> LibriId, Emri, Autori, Data_Publikimit, NumriFaqeve, Kategorid

Kategoria -> Kategorid, Emri =>

```
namespace Biblioteka.Models
{
    public class Kategori
    {
        public int KategoriId { get; set; }
        public string Emri { get; set; }
    }
}

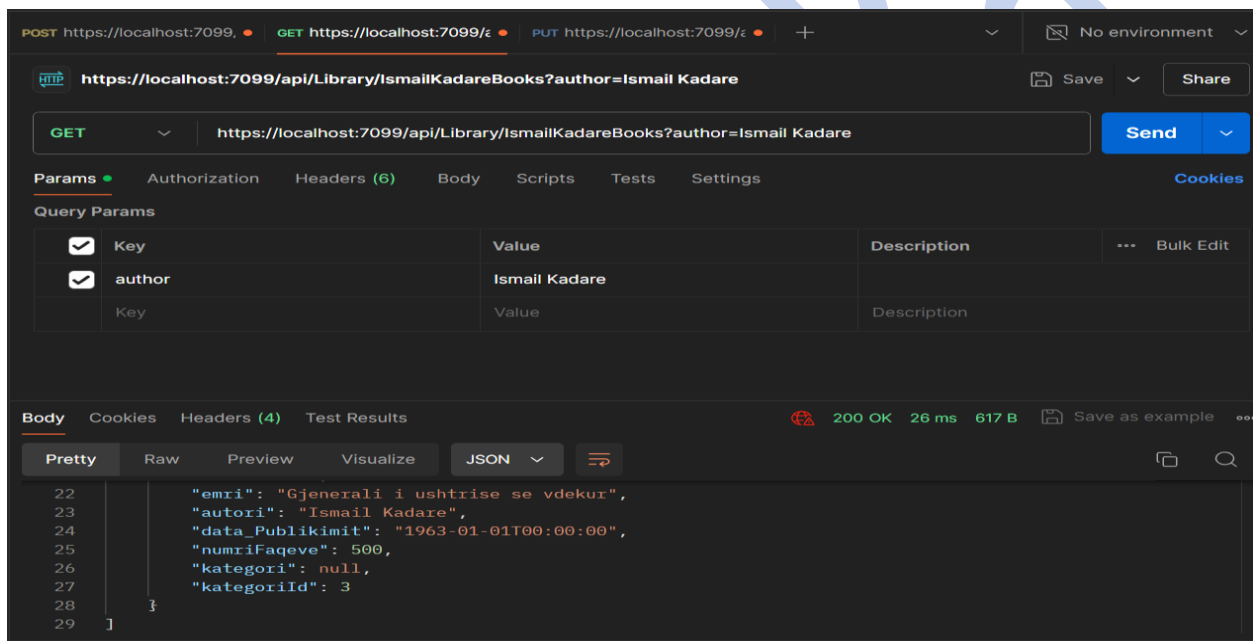
namespace Biblioteka.Models
{
    public class Libri
    {
        public int LibriId { get; set; }
        public string? Emri { get; set; }
        public string? Autori { get; set; }
        public DateTime Data_Publikimit { get; set; }
        public int NumriFaqeve { get; set; }
        public Kategori Kategori { get; set; }
        public int KategoriId { get; set; }
    }
}
```

Pas krijimit të modeleve bëjme lidhjen me db shtojme migrimet perkatese . Me pas ndertojme nje controller Api Brenda nje foler te quajtur API. Ky controller do te krijohe bosh fillimisht dhe do te krijohe actionat e shtimit [post] te librave dhe kategorive dhe marrjes [get] te librave dhe kategorive te bibliotekes.



2. Afishoni librat që janë botuar nga Ismail Kadare para vitit 2010. (GET) (Autori vjen si parameter)

```
[HttpGet("IsmailKadareBooks")]
public ActionResult<IEnumerable<Libri>> GetIsmailKadareBooks(string author)
{
    var books = _context.Libri.Where(b => b.Autori == author &&
b.Data_Publikimit.Year < 2010).ToList();
    if (books == null || books.Count == 0)
    {
        return NotFound("Nuk ka libra të botuar nga Ismail Kadare para vitit
2010.");
    }
    return books;
}
```



3. Beni update gjithë librat te cilat jane botuar pas vitit 2000 dhe vendosni kategorine Millenium (PUT). (Viti vjen si parameter)

```
[HttpPut("UpdateBooks/{year}")]
public IActionResult UpdateBooks(int year)
{
    // Check if the "Millenium" category exists
    var millenniumCategory = _context.Kategori.FirstOrDefault(c => c.Emri ==
"Millenium");

    // If "Millenium" category doesn't exist, create it
    if (millenniumCategory == null)
    {
        millenniumCategory = new Kategori { Emri = "Millenium" };
        _context.Kategori.Add(millenniumCategory);
        _context.SaveChanges();
    }
}
```

```

// Get books published after the specified year
var booksToUpdate = _context.Libri
    .Where(b => b.Data_Publikimit.Year > year)
    .ToList();

// If there are no books to update, return an error
if (booksToUpdate == null || booksToUpdate.Count == 0)
{
    return NotFound("Nuk ka libra për të përditësuar pas vitit " +
year);
}

// Update the categories of books to "Millenium"
foreach (var book in booksToUpdate)
{
    book.Kategori = millenniumCategory; // Set the category
    book.KategoriId = millenniumCategory.KategoriId; // Set the category
ID
}

_context.SaveChanges();
return NoContent();
}

```

POST https://localhost:7099/api/Library/UpdateBooks/2000

PUT https://localhost:7099/api/Library/UpdateBooks/2000

Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (2) Test Results

204 No Content 2.21s 81B Save as example

Pretty Raw Preview Visualize Text

```

1

```

GET https://localhost:7099/api/Library/GetBookById/6

Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (4) Test Results

200 OK 101ms 329B Save as example

Pretty Raw Preview Visualize JSON

```

5  "data_Publikimit": "2019-08-06T08:00:00",
6  "numriFaqeve": 330,
7  "kategori": {
8    "kategoriId": 8,
9    "emri": "Millenium"
10 },
11 "kategoriId": 8
12 }

```

4. Krijoni API që mundëson fshirjen e Librave. (DELETE)

```
// DELETE: api/Library/DeleteBook/5
[HttpDelete("DeleteBook/{id}")]
public IActionResult DeleteBook(int id)
{
    var bookToDelete = _context.Libri.Find(id);
    if (bookToDelete == null)
    {
        return NotFound("Libri nuk u gjet.");
    }

    _context.Libri.Remove(bookToDelete);
    _context.SaveChanges();
    return NoContent();
}
```

Ky action është delete dhe merr një id të caktuar libri dhe e fshin të dhënat e tij nga db

The screenshot shows a Postman interface for a DELETE request to `https://localhost:7099/api/Library/DeleteBook/4`. The request is successful, returning a `204 No Content` status. The response body is empty.

Key	Value	Description
Key	Value	Description

The screenshot shows a Postman interface for a GET request to `https://localhost:7099/api/Library/GetBookById/4`. The request fails, returning a `404 Not Found` status. The response body contains the text `Libri nuk u gjet.`.

Key	Value	Description
Key	Value	Description