

REST API

G04Code

Marko Cvijanović
Mihailo Bubnjević

GUID

- Global Unique Identifier
- 128-bit integer (16 bytes)
- 5,316,911,983,139,663,491,615,228,241,121,400,000 mogućih kombinacija
- Koristimo kao globalni identifikator svih redova u bazi
- Zamenjuje ID kao primary key
- Replikacija baze zahteva GUID ako će podaci biti pisani u sve baze
- Primer GUID-a:

C5024087-C75B-4ED4-B594-71AB6D91AB6B

GUID

- Prednosti:
 - Jedinствен za sve tabele, sve baze i sve severe
 - Olakšava spajanje tabela iz različitih baza
 - Može da se generiše bilo gde bez proveravanja baze
- Mane:
 - 4 puta veći od tradicionalnih primarnih ključeva
 - Udarac na performanse i memoriju

RESTful API

- REST (REpresentational state transfer)
- Razdvaja Klijenta i Server - ne znaju i ne zanima ih stanje ovog drugog
- Stateless - svaki zahtev je nezavistan
- Principi kojima se treba voditi:
 - Imenice, ne glagoli
 - Konzistentnost
 - Šalji korisniku samo ono što mu treba
 - Verzionisanje
 - KISS (Keep it Simple, Stupid)
 - Validirari sve
 - Ažurirana dokumentacija...

Slojevita Arhitektura (Layer Architecture)

- **Data Access Layer (DAL) - Repository**
 - Odgovoran za komunikaciju sa bazom i rad sa podacima
- **Business Logic Layer (BLL) - Service**
 - Obalja svu domensku logiku
- **Presentation Layer / UI - Controller**
 - Odgovoran za komunikaciju i interakciju sa korisnikom

Class Library

- Class library je koncept koji se koristi u .NET-u kako bismo razdvojili komponente u module koje ispunjavaju konkretne funkcionalnosti.
- Postoje 3 vrste:
 - Platform-specific - vezane su za jednu .NET platformu
 - Portable - dostupne na više .NET platformi
 - .NET Standard - spaja prethodna dva koncepta u jedan model koji pruža funkcionalnost obe vrste

Presentation Layer - Kontroleri

- Jedina dostupna tačka korisniku koji želi da koristi sistem
- Prevodi korisničke zahteve u podatke sa kojima sistem može da radi putem
- Obavlja serijalizaciju i deserijalizaciju
- Daje “feedback” korisniku koji on može da razume
- Validira resurse zahteva i mapira ih na parametre ako su adekvatni
- Kontroleri sadrže endpoint metode koje takodje nazivamo **Akcije**

Kontroleri

- Da bi imali funkcionalnosti obrade HTTP zahteva i generisanja odgovora, treba da naslede ControllerBase baznu klasu
- ControllerBase je lightweight verzija Controller bazne klase koja je nasleđuje i dodaje funkcionalnost za MVC .NET aplikacije
- ControllerBase nam pruža korisne funkcije za generisanje odgovora kao što su:
 - Ok()
 - BadRequest()
 - NotFound()
 - Created()...

Kontroleri

- Atributi ili Anotacije nam pomažu da konfigurišemo ponašanje kontrolera i njihovi metoda
- Primeri atributa:
 - **[ApiController]** - pruža dodatne funkcionalnosti koje smanjuju količinu boilerplate koda
 - **[Route]** - za specifikaciju URL šablona na nivou kontrolera ili metode
 - **[Consumes]** i **[Produces]** - za konkretisanje tipova podata koje će controller ili metoda da šalje i prima
 - **[HttpGet]**, **[HttpPost]**, **[HttpPut]**... - za definisanje http metode
 - **[FromBody]**, **[FromRoute]**... - za mapiranje podataka iz zahteva

[ApiController]

- Obavlja automatsku validaciju podataka iz zahteva i vraća status kod 400 bez potrebe za proverom ModelState.IsValid
- Dodaje **ValidationProblem** format za HTTP 400 odgovore
- Zahteva od kontrolera da definiše rutu putem **[Route]** atributa i čini konvencionalne načine rutiranja(*app.MapControllerRoute()*) nedostupnim
- Pretpostavlja gde u zahtevu se nalazi vrednost za parametar bez eksplicitnog navođenja putem atributa kao što su FromBody, FromRoute...
- To nazivamo “Binding source parameter inference”

[ApiController]

- Pravila **Binding source parameter inference**-a:

Binding Attribute	Inference Rule
[FromBody]	Inferred for any complex type
[FromForm]	Inferred for any parameters of type <code>IFormFile</code> or <code>IFormFileCollection</code>
[FromRoute]	Inferred for any action parameter whose name matches a parameter in the route template
[FromQuery]	Inferred for all the other action parameters
[FromHeader]	No rule for header binding

[ApiController]

- Možemo da onemogućimo ova pravila kroz sledeću konfiguraciju:

```
builder.Services.AddControllers()  
    .ConfigureApiBehaviorOptions(options =>  
    {  
        options.SuppressInferBindingSourcesForParameters = true;  
    });
```

- To će zahtevati da ručno definišemo izvor podataka

Atributi za definisanje izvora podataka

Attribute	Binding source
<u>[FromBody]</u>	Request body
<u>[FromForm]</u>	Form data in the request body
<u>[FromHeader]</u>	Request header
<u>[FromQuery]</u>	Request query string parameter
<u>[FromRoute]</u>	Route data from the current request
<u>[FromServices]</u>	The request service injected as an action parameter
<u>[AsParameters]</u>	<u>Method parameters</u>

Consumes i Produces

- Default-no podešavanje za [Consumes] je da prima sve tipove sadržaja.
- Koristimo [Consumes] kada želimo da ogrničimo podržane tipove.
- Na primer: **[Consumes("application/xml")]**
- Sa atributom [Produces] možemo da definišemo konkretan format u kom će se slati odgovori na zahteve

Povratni tipovi Akcija (Action Return types)

- **Specifični tip**
 - vraćamo primitivni ili kompleksni data tip, kao što je string ili naš objekat
- **ActionResult**
 - pogodan kada imamo više različitih povratnih tipova unutar jedne akcije
 - Ima podršku za većinu HTTP status kodova
 - Npr: `OkObjectResult(200)`, `NotFoundResult(404)`...
- **ActionResult<T>**
 - nasledjuje `ActionResult`, `T` definiše tip podataka koje vraćamo
 - skraćuje drugi Atribute kao što je `[ProducesResponseType]` gde više ne moramo da navodimo tip, samo status kod.

Povratni tipovi Akcija (Action Return types)

- **IResult**
 - Novina u .NET 6.0
 - Ponaša se slično kao **ActionResult** gde da bismo vratili određeni status kod moramo da iskoristimo statičnu klasu **Results** i nad njom pozovemo adekvatnu metodu (**Results.Ok()**, **Results.NotFound()**..)
- **Result<T>**
 - Statična klasa **TypedResult** se koristi da vraća konkretnu **IResult** implementaciju
 - **Results<TResult1, TResultN>** je unija povratnih kodova koje definišemo u potpisu metode
 - Npr: **public Results<NotFound, Ok<Object>> GetById(int id)**