

SVEUČILIŠTE U MOSTARU

**FAKULTET STROJARSTVA, RAČUNARSTVA I ELEKTROTEHNIKE**

**TEHNIČKA DOKUMENTACIJA**

**ELEKTRONIČKO POSLOVANJE**

**Konferencije i predavači – mini web aplikacija povezana sa Genesis bazom podataka**

Studenti:

Vlado Boškić

Ivan Sesar

Profesor: izv.prof.dr.sc. Željko Stojkić

Mostar, veljača 2020.

Sadržaj:

[1.Uvod 3](#_Toc32518449)

[2. Spajanje na bazu i inicijalizacija web aplikacije 4](#_Toc32518450)

[2.1 Pokretanje aplikacije na lokalnom računalu 6](#_Toc32518451)

[3. Definiranje funkcija za rad u Genesisu 7](#_Toc32518452)

[3.1 Definiranje REST servisa za konferencije 8](#_Toc32518453)

[3.2 Definiranje REST servisa za predavače 10](#_Toc32518454)

[3.3 Kreiranje i brisanje registracije 11](#_Toc32518455)

[4. Pokretanje i rad s aplikacijom 13](#_Toc32518456)

# 1.Uvod

Zadatak naše grupe bio je napravit web aplikaciju koja bi nekoj firmi olakšala organiziranje konferencija, te evidentiranje predavača koji trebaju nastupiti na konferencijama. Svi podaci vezani za konferencije i predavače trebaju biti spremljeni u Genesis bazu podataka. Svaku kreiranu konferenciju treba moći urediti ili izbrisati. Iste funkcionalnosti su implementirane za predavače. Također, svaki predavač treba imati mogućnost da se prijavi na bilo koju konferenciju. Program smo napravili u Visual studio programskom okviru koristeći C# programski jezik.

# 2. Spajanje na bazu i inicijalizacija web aplikacije

Da bismo pristupili Genesis bazi podataka, prvo se na nju moramo povezati. Na bazu se spajamo preko definiranog linka i korisničkog imena i lozinke te produkt ključa.

|  |
| --- |
| "HttpSettings": {  "Username": "Student1",  "Password": "student1",  "Url": "http://212.39.115.5:8585/genesisrest.svc",  "ProductKey":  "Test\_REST+SOAP\_20200401\_B1B3\_zqwdd4UfCGoxW8EUQ7EaV0Y4P39yTTBwfXGEVaRP/PmMLcq7633ZQ1FsZQxqAwBEchgs3Wrg7kQ/uWtOj3EPQg==" |

Podatke za spajanje na bazu dohvaćamo kroz klasu HttpSettings.

|  |
| --- |
| namespace EP.Conference.Settings  {  public class HttpSettings  {  public string Url { get; set; }  public string Username { get; set; }  public string Password { get; set; }  public string ProductKey { get; set; }  }  } |

Nakon definiranih podataka za spajanje na bazu i definiranja klase koja dohvaća pristupne podatke, možemo se spojiti na bazu.

Datoteka *Startup.cs* sadrži klasu pokretanja koja se prvo pokreće kada se aplikacija pokrene i u njoj definiramo spajanje na bazu podataka. U *Startup.cs* također su definirani servisi koji su potrebni za rad sa Genesis bazom podataka.

|  |
| --- |
| public void ConfigureServices(IServiceCollection services)  {  services.Configure<HttpSettings>(\_httpSettingsSection);  services.AddTransient<IGenesisService, GenesisService>();  services.AddSingleton<IGenesisFactory, GenesisFactory>();  services.AddDbContext<ConferenceDbContext>(options => options.UseInMemoryDatabase("ConferenceDb"));  services.Configure<CookiePolicyOptions>(options =>  {  // This lambda determines whether user consent for non-essential cookies is needed for a given request.  options.CheckConsentNeeded = context => true;  options.MinimumSameSitePolicy = SameSiteMode.None;  });  services.AddAutoMapper(AppDomain.CurrentDomain.GetAssemblies());  services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version\_2\_2);  } |

## 2.1 Pokretanje aplikacije na lokalnom računalu

Datoteka *launchSettings.json* koristi se samo na lokalnom razvojnom stroju. To znači da ova datoteka nije potrebna kada na proizvodnom poslužitelju objavljujemo osnovnu aplikaciju.

U njoj se definirani portovi na kojima se aplikacija pokreće.

|  |
| --- |
| {  "iisSettings": {  "windowsAuthentication": false,  "anonymousAuthentication": true,  "iisExpress": {  "applicationUrl": "http://localhost:51215",  "sslPort": 44325  }  },  "profiles": {  "IIS Express": {  "commandName": "IISExpress",  "launchBrowser": true,  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  }  },  "EP.Conference": {  "commandName": "Project",  "launchBrowser": true,  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  },  "applicationUrl": "https://localhost:5001;http://localhost:5000"  } |

# 3. Definiranje funkcija za rad u Genesisu

U klasama *IGenesisService* i *GenesisService* definirane su naredbe za pristup i uređivanje podataka u bazi podataka.

U aplikaciji postoje polja konferencije, predavači te registracija. U konferencijma možemo kreirati novu konferenciju , urediti istu ili je izbrisati. Defeinirana su i dva pogleda, jedan za prikaz svih konferencija, a drugi za pregled detelja određene konferencije slanjem ID-a konfrencije.

Kod predavača također možemo vršiti slična uređivanja i definirana su dva pogleda kao kod predavača. Za registraciju predavača potrebna su funkcije samo za registraciju predavača i brisanje predavača sa određene konferencije.

|  |
| --- |
| namespace EP.Conference.Contracts  {  public interface IGenesisService  {  #region Registration  Task DeleteRegistrationAsync(string id);  Task CreateRegistrationAsync(RegistrationFields registerFields);  Task<List<RegistrationViewModel>> GetRegistrationsAsync();  Task<RegistrationViewModel> GetRegistrationAsync(string id);  #endregion  #region Speakers  Task<List<SpeakerViewModel>> GetSpeakersAsync()  Task<ExtendedSpeakersField> GetSpeakerAsync(string id);  Task UpdateSpeakerAsync(ExtendedSpeakersField model);  Task CreateSpeakerAsync(SpeakerFields model);  Task DeleteSpeakerAsync(string id);  #endregion  #region Conference  Task DeleteConferenceAsync(string id);  Task UpdateConferenceAsync(ConferenceFields model);  Task CreateConferenceAsync(ConferenceFields model);  Task<List<ConferenceViewModel>> GetConferencesAsync();  Task<ConferenceViewModel> GetConferenceAsync(string id);  #endregion  }  } |

## 3.1 Definiranje REST servisa za konferencije

U ovom dijelu definirani su servisi potrebni za rad sa konferencijama. Korištene su metode: DELETE, GET , PUT i POST.

Funkcija za kreiranje konferencije koristi POST metodu i pohranjuje se u EVENT objekt u Genesisu.

|  |
| --- |
| public async Task CreateConferenceAsync(ConferenceFields model)  {  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/EVENT", Method.POST, DataFormat.Json);  var body = \_mapper.Map<ConferenceFields, ConferenceRequest>(model);  var data = new {fields = body};  var serialized = JsonConvert.SerializeObject(data);  request.AddJsonBody(serialized);  await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

U funkciji iznad se također poziva i funckija *ConferenceFields* koja prikazuje predefinirana polja za konferencije.

|  |
| --- |
| [Display(Name = "Kraj")]  public DateTime EV\_ENDDT { get; set; }  public bool EV\_FURTHERPARTICIPANTS { get; set; }  [Display(Name = "Početak")]  public DateTime EV\_STARTDT { get; set; }  [Display(Name = "Naziv konferencije")]  public string EV\_TITLE { get; set; }  [Display(Name = "Lokacija")]  public string EV\_VENUE { get; set; } |

Funkcija za ažuriranje podataka unutar konferencije naziva se *UpdateConferenceAsync* i koristi PUT metodu.

|  |
| --- |
| public async Task UpdateConferenceAsync(ConferenceFields model)  {  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/EVENT/{model.GGUID}", Method.PUT, DataFormat.Json);  request.AddHeader("If-Match", model.INSERTUSER);  var body = \_mapper.Map<ConferenceFields, ConferenceRequest>(model);  var data = new { fields = body };  var serialized = JsonConvert.SerializeObject(data);  request.AddJsonBody(serialized);  await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

Funkcija za brisanje konferencije naziva *DeleteConferenceAsync* se i koristi DELETE metodu.

|  |
| --- |
| public async Task DeleteConferenceAsync(string id)  {  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/EVENT/{id}", Method.DELETE, DataFormat.Json);  await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

## 3.2 Definiranje REST servisa za predavače

Da bi kreirali novog predavača, koristimo POST metodu i spremamo ga u objekt ADDRESS u Genesis bazi podataka.

|  |
| --- |
| public async Task CreateSpeakerAsync(SpeakerFields model)  {  model.GWBRANCH = "Education and Teaching";  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/ADDRESS", Method.POST, DataFormat.Json);  var data = new { fields = model };  var serialized = JsonConvert.SerializeObject(data);  request.AddJsonBody(serialized);  var restResponse = await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

Za uređivanje informacija pojedinog predavača koristimo funkciju *UpdateSpeakerAsync* i pozivamo metodu PUT.

|  |
| --- |
| public async Task UpdateSpeakerAsync(ExtendedSpeakersField model)  {  model.GWBRANCH = "Education and Teaching";  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/ADDRESS/{model.GGUID}", Method.PUT, DataFormat.Json);  request.AddHeader("If-Match", model.Etag);  var body = \_mapper.Map<ExtendedSpeakersField, SpeakerFields>(model);  var data = new { fields = body };  var serialized = JsonConvert.SerializeObject(data);  request.AddJsonBody(serialized);  var restResponse = await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

Za brisanje određenog predavača poziva se funkcija DeleteSpeakerAsync u kojoj se koristi DELETE metoda.

|  |
| --- |
| public async Task DeleteSpeakerAsync(string id)  {  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/ADDRESS/{id}", Method.DELETE, DataFormat.Json);  await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

I predavači imaju svoju funkciju za pogled, a naziva se *SpeakerFields*.

|  |
| --- |
| public class SpeakerFields  {  [DisplayName("Ime")]  public string CHRISTIANNAME { get; set; }  [DisplayName("Prezime")]  public string NAME { get; set; }  [DisplayName("Id")]  [Key] public string GGUID { get; set; }  [DisplayName("Kompanija")]  public string COMPNAME { get; set; }  public string GWBRANCH { get; set; }  } |

## 3.3 Kreiranje i brisanje registracije

U aplikaciji je omogućeno da se na svaku konferenciju prijavi jedan ili više predavača. Prijave se također mogu i ukloniti.

Funkcija za prijavu na konferenciju zove se *CreateRegistrationAsync* i koristi REST POST metodu i sprema se u objekt REGISTRATION na Genesisu.

|  |
| --- |
| public async Task CreateRegistrationAsync(RegistrationFields registerFields)  {  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/REGISTRATION", Method.POST,  DataFormat.Json);  var body = \_mapper.Map<RegistrationFields, RegisterRequest>(registerFields);  var data = new { fields = body };  var serialized = JsonConvert.SerializeObject(data);  request.AddJsonBody(serialized);  await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

Za brisanje registracije koristimo DeleteRegistrationAsync funkciju kojoj prosljeđujemo ID registracije koju želimo izbrisati. Ova funkcija koristi REST DELETE metodu.

|  |
| --- |
| public async Task DeleteRegistrationAsync(string id)  {  var client = \_factory.GetGenesisClient();  var cancellationTokenSource = new CancellationTokenSource();  var request = new RestRequest($"/v3.0/type/REGISTRATION/{id}", Method.DELETE, DataFormat.Json);  await client.ExecuteAsync(request, cancellationTokenSource.Token);  } |

# 

# 4. Pokretanje i rad s aplikacijom

Da bi pokrenuli aplikaciju moramo imati instaliran Visual Studio i ASP.Net 2.2.

Linkovi za instalaciju:

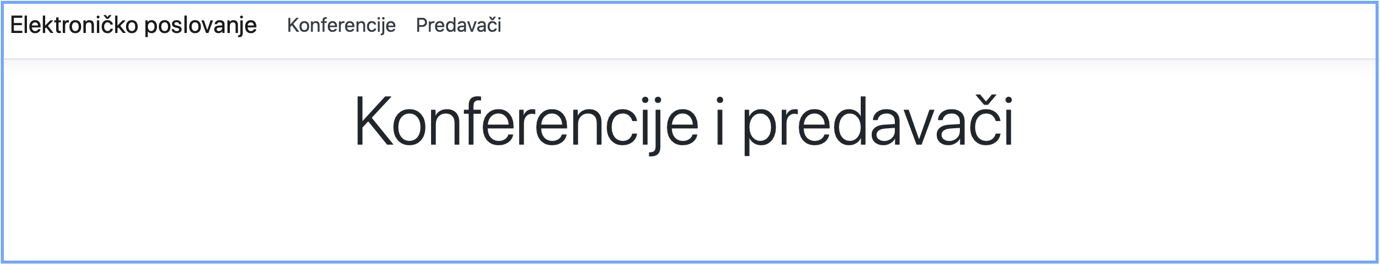
<https://visualstudio.microsoft.com/downloads/> ; <https://dotnet.microsoft.com/download/dotnet-core/2.2>

Nakon pokretanja projekta u Visual Studiju, u web browseru otvara se početna stranica web aplikacije.



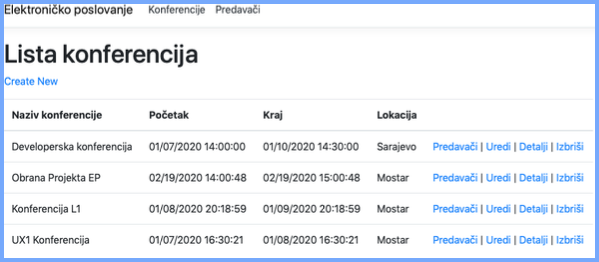
**Slika 1.** Uspijesno kompajliran i pokrenut kod aplikacije

Početna stranica otvara se u novom prozoru defalutnog web browsera i prikazana je na slici 2.



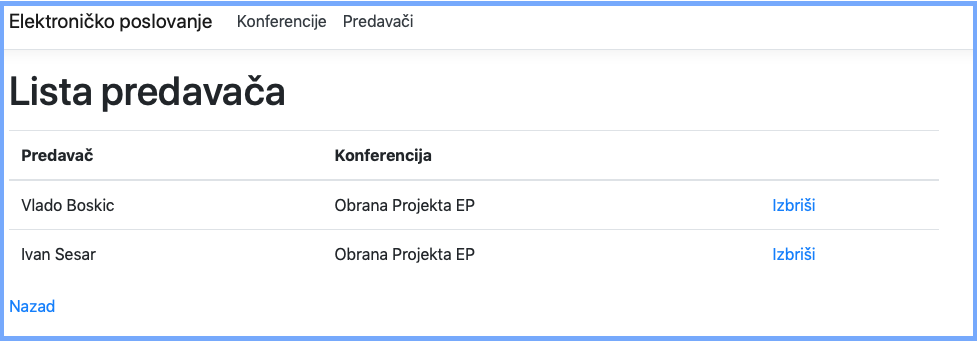
**Slika 2.** Početna stranica web aplikacije

Klikom na tab Konferencije u web aplikaciji učitavaju nam se sve konferencije koje su pohranjene u bazi podataka. Pored naziva svake konferencije nalazi se vrijeme i mjesto održavanja. Svaku konferenciju možemo urediti, izbrisati ili pogledati koji predavači nastupaju na istoj konferenciji.

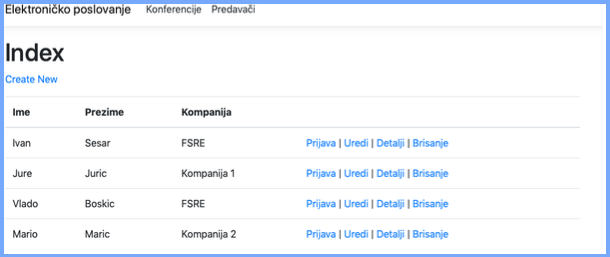


**Slika 3.** Prikaz konferencija

Ako u listi konferencija kliknemo na predavači, pokazuju nam se svi predavači koji su prijavljeni na tu konferenciju.



**Slika 4.** Prikaz predavača za određenu konferenciju

**Slika 5.** Prikaz svih predavača