

GIK2F7 Projektarbete

WebApi + WPF App

WebApi

WebApi - DatabaseService.cs

- Hanterar förfrågningar till databasen
- Tar emot data och skickar tillbaka data

DatabaseService.Get(...)

- Två överlagrade metoder
- SqlConnection
- QueryAsync

```
1 reference
154 public async Task<GameInfo> Get(int Id)
155 {
156     using (var Connection = new SqlConnection(databaseConfig.Name))
157     {
158         var res = await Connection.QueryAsync<GameInfo>("SELECT Id, Name, Description, Image, Grade FROM Games WHERE Id=@Id", new { Id });
159         return res.FirstOrDefault();
160     }
161 }
1 reference
138 public async Task<IEnumerable<GameInfo>> Get()
139 {
140     using (var Connection = new SqlConnection(databaseConfig.Name))
141     {
142         var res = await Connection.QueryAsync<GameInfo>("SELECT Id, Name, Description, Image, Grade FROM Games ORDER BY Id DESC");
143         return res;
144     }
145 }
```

DatabaseService.AddGame(...)

- ExecuteAsync
- FirstOrDefault

1 reference

```
128 public async Task<GameInfo> AddGame(GameInfo game)
129 {
130     using (var Connection = new SqliteConnection(databaseConfig.Name))
131     {
132         var res = await Connection.ExecuteAsync("INSERT INTO Games (Name, Description, Image, Grade) VALUES (@Name, @Description, @Image, @Grade)");
133         var lastInsert = await Connection.QueryAsync<GameInfo>("SELECT Id, Name, Description, Image, Grade FROM Games ORDER BY Id DESC");
134         game.Id = lastInsert.FirstOrDefault<GameInfo>().Id;
135         return game;
136     }
137 }
```

GameRepository

- Tar emot och skickar data till DatabaseService
- Skickar tillbaka det som kommer ifrån DatabaseService

```
0 references
19 public GameRepository(DatabaseConfig DBConfig)
20 {
21     DBService = new DatabaseService(DBConfig);
22 }

1 reference
23 public Task<GameInfo> Add(GameInfo game)
24 {
25     return DBService.AddGame(game);
26 }

2 references
27 public Task<GameInfo> Update(GameInfo game)
28 {
29     return DBService.UpdateGame(game);
30 }
```

GameInfoController

- Innehåller routes
- CRUD operationer
- GetAll, GetGame, AddGame, UpdateGame, DeleteGame
- HttpPost/Get/Put/Delete

```
45 [HttpGet("GetGame/{Id}")]
    0 references
46 public async Task<GameInfo> GetGame(int Id)
47 {
48     return await gameRepository.Get(Id);
49 }
50 [HttpPost("AddGame")]
    0 references
51 public async Task<GameInfo> AddGame(GameInfo NewGame)
52 {
53     return await gameRepository.Add(NewGame);
54 }
```

DatabaseService.UpdateGame(...)

- Använder sig av UpdateWhat(...)
- int 1000, 100, 10, 1 -> 1100, 1010, 1111...
- Kollar vad som ska uppdateras
- Endast det som ska uppdateras uppdateras

DatabaseService.UpdateWhat(...)

1 reference

```
public async Task<GameInfo> UpdateGame(GameInfo game) ...
```

1 reference

```
private int UpdateWhat(GameInfo game)
```

```
{
```

```
    //Wizardry Time! By doing this I basically create an int that can be interpreted as binary where 1 is true and 0 is false.
```

```
    //I did this to be able to use a switch statement instead of a horrendous 2^4 if statements since you can't switch with an array of bools or i
```

```
    //1000 is update name, 0100 is update description, 0010 is update image, 0001 is update grade. Combine for full extent of changes.
```

```
    int UpdateThis = 0;
```

```
    if (game.Name != null && game.Name != "")
```

```
    {    //Set name
```

```
        UpdateThis += 1000;
```

```
    }
```

```
    if (game.Description != null && game.Description != "")
```

```
    {    //Set Description
```

```
        UpdateThis += 100;
```

```
    }
```

```
    if (game.Image != null && game.Image != "")
```

```
    {    //Set Image
```

```
        UpdateThis += 10;
```

```
    }
```

```
    if (game.Grade > -1)
```

```
    {    //Set Grade
```

```
        UpdateThis += 1;
```

```
    }
```

```
    return UpdateThis;
```

```
}
```

WPF App

WPF App - Visibility

- Visibility 0/2, Visible/Collapsed
- Flera element, samma plats

GameApiHandler.cs

- Hanterar förfrågningar mot WebApi
- Tar emot ifrån WebApi

```
11 3 references
12 class GameApiHandler
13 {
14     private string BaseUrl;
15     1reference
16     public GameApiHandler(string Url)
17     {
18         BaseUrl = Url;
19     }
20     1reference
21     public GameInfo GetGame(int Id)
22     {
23         GameInfo Game = null;
24         using (WebClient webClient = new WebClient())
25         {
26             string GetGameUrl = BaseUrl + "GetGame/" + Id.ToString();
27             var jsonData = webClient.DownloadString(GetGameUrl);
28             Game = JsonSerializer.Deserialize<GameInfo>(jsonData);
29         }
30         return Game;
31     }
32 }
```

GameApiHandler.AddGame/UpdateGame

- HttpClient
- StringContent
- PostAsync/PutAsync

```
1reference
40 public async void AddGame(GameInfo game)
41 {
42     using (HttpClient httpClient = new HttpClient())
43     {
44         string AddGameUrl = BaseUrl + "AddGame";
45         var dataToSend = new StringContent(JsonSerializer.Serialize(game), Encoding.UTF8, "Application/json");
46         var jsonData = await httpClient.PostAsync(AddGameUrl, dataToSend);
47     }
48 }
1reference
49 public async void UpdateGame(GameInfo game)
50 {
51     using (HttpClient httpClient = new HttpClient())
52     {
53         string UpdateGameUrl = BaseUrl + "UpdateGame";
54         var dataToSend = new StringContent(JsonSerializer.Serialize(game), Encoding.UTF8, "Application/json");
55         var jsonData = await httpClient.PutAsync(UpdateGameUrl, dataToSend);
56     }
57 }
```

MainWindow.xaml.cs

- Button_Click(...)
- Game_List_SelectionChanged(...)

```
1 reference
32 private void Game_List_SelectionChanged(object sender, SelectionChangedEventArgs e)
33 {
34     if (Game_List.SelectedItem != null)
35     {
36         GameInfo selectedGame = (GameInfo)Game_List.SelectedItem;
37         Selected_Game_Game_Name.Text = selectedGame.name;
38         Selected_Game_Game_Id.Text = selectedGame.id.ToString();
39         Selected_Game_Game_Score.Text = selectedGame.grade.ToString();
40         Selected_Game_Game_Description.Text = selectedGame.description;
41     }
42 }
```

Demo!