

Publishing user updated events

Script start

Let's now update our Identity microservice to publish a new event any time a user is updated.

In Identity repo

1. Update Contracts:

```
namespace Play.Identity.Contracts
{
    ...
    public record GilDebited(Guid CorrelationId);
    public record UserUpdated(Guid UserId, string Email, decimal NewTotalGil);
}
```

2. Update DebitGilConsumer:

```
public class DebitGilConsumer : IConsumer<DebitGil>
{
    ...
    public async Task Consume(ConsumeContext<DebitGil> context)
    {
        ...

        await userManager.UpdateAsync(user);

        var gitDebitedTask = context.Publish(new GilDebited(message.CorrelationId));
        var userUpdatedTask = context.Publish(new UserUpdated(user.Id, user.Email, user.Gil));

        await Task.WhenAll(userUpdatedTask, gitDebitedTask);
    }
}
```

3. Update Register.cshtml.cs:

```
public class RegisterModel : PageModel
{
    ...
    private readonly IdentitySettings identitySettings;
    private readonly IPublishEndpoint publishEndpoint;

    public RegisterModel(
```

```

...
IOptions<IdentitySettings> identityOptions,
IPublishEndpoint publishEndpoint)
{
    ...
    this.identitySettings = identityOptions.Value;
    this.publishEndpoint = publishEndpoint;
}

...

public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    ...
    if (ModelState.IsValid)
    {
        ...
        if (result.Succeeded)
        {
            ...
            await _userManager.AddToRoleAsync(user, Roles.Player);

            await publishEndpoint.Publish(new UserUpdated(user.Id, user.Email, user.Gil));

            var code = await _userManager.GenerateEmailConfirmationTokenAsync(user);
            ...
        }
        ...
    }
    ...
}
}

```

4. Update UsersController:

```

public class UsersController : ControllerBase
{
    private readonly UserManager<ApplicationUser> userManager;
    private readonly IPublishEndpoint publishEndpoint;

    public UsersController(
        UserManager<ApplicationUser> userManager,
        IPublishEndpoint publishEndpoint)
    {

```

```

    this.userManager = userManager;
    this.publishEndpoint = publishEndpoint;
}

...

// /users/{123}
[HttpPut("{id}")]
public async Task<IActionResult> PutAsync(Guid id, UpdateUserDto userDto)
{
    ...

    await userManager.UpdateAsync(user);

    await publishEndpoint.Publish(new UserUpdated(user.Id, user.Email, user.Gil));

    return NoContent();
}

// /users/{123}
[HttpDelete("{id}")]
public async Task<IActionResult> DeleteAsync(Guid id)
{
    ...

    await userManager.DeleteAsync(user);

    await publishEndpoint.Publish(new UserUpdated(user.Id, user.Email, 0));

    return NoContent();
}
}

```

5. Publish contracts NuGet package:

```
dotnet pack -p:PackageVersion=1.0.1 -o ..\..\packages
```

In the next lesson we will update the Trading microservice so it can consume the new events from Inventory and Identity.