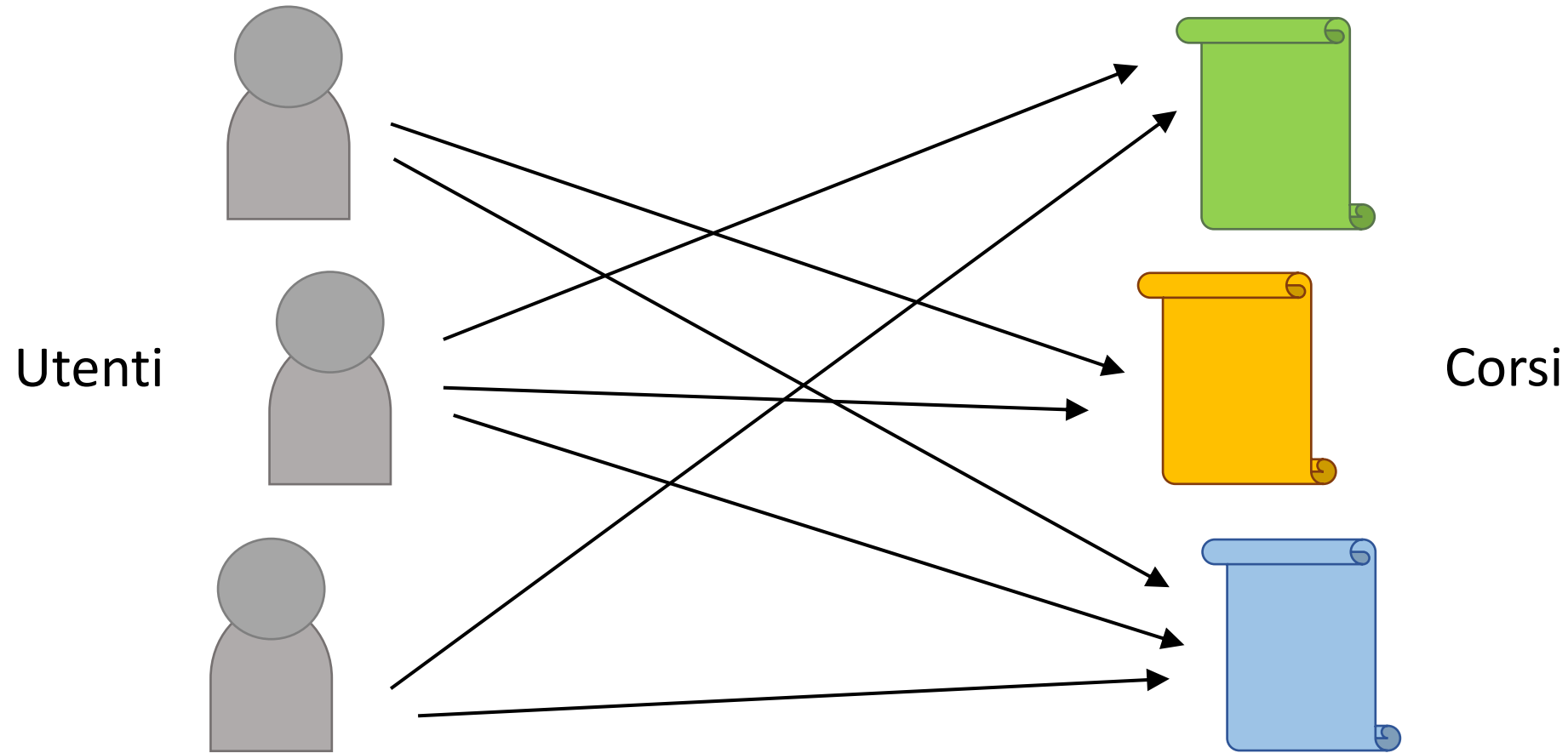


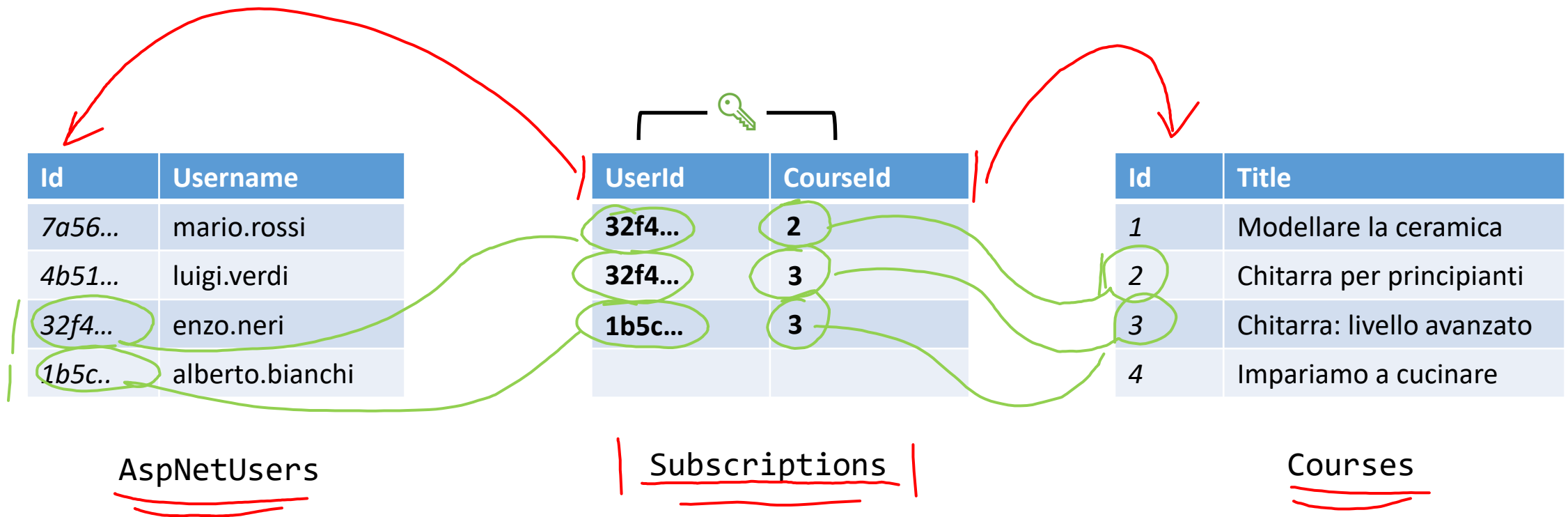
Sezione 21

Funzionalità e-commerce

Relazione tra utenti e corsi



Relazione tra utenti e corsi (database)



Altre colonne della tabella di raccordo

<u>UserId</u>	<u>CourseId</u>

Subscriptions

- Data del pagamento?
- Quanto ha pagato?
- Con che modalità?
- L'id della transazione?
- La valutazione?

Tabella Subscriptions

[illegible]

Relazione tra utenti e corsi (EFCore 5)

```
public class ApplicationUser
{
    public string FullName { get; set; }
    public ICollection<Course> SubscribedCourses { get; set; }
}

public class Course
{
    public string Title { get; set; }
    public ICollection<ApplicationUser> SubscribedUsers { get; set; }
}
```

EFCore 5: mappare relazioni multi-a-molti

Nel metodo **OnModelCreating** del file **MyCourseDbContext.cs**

```
modelBuilder.Entity<Course>(entity =>
{
    entity.HasMany(course => course.SubscribedUsers)
           .WithMany(user => user.SubscribedCourses)
           .UsingEntity(join => join.ToTable("CourseStudents"));
})
```

EFCore 5: relazioni multi-a-molti con join entity

Nel metodo **OnModelCreating** del file **MyCourseDbContext.cs**

```
entity.HasMany(course => course.SubscribedUsers)
    .WithMany(user => user.SubscribedCourses)
    .UsingEntity<Subscription>(
        entity => // Mapping da Subscription a User
        entity => // Mapping da Subscription a Course
        entity => {
            entity.ToTable("Subscriptions");
            // Altro mapping
        }
    );
```

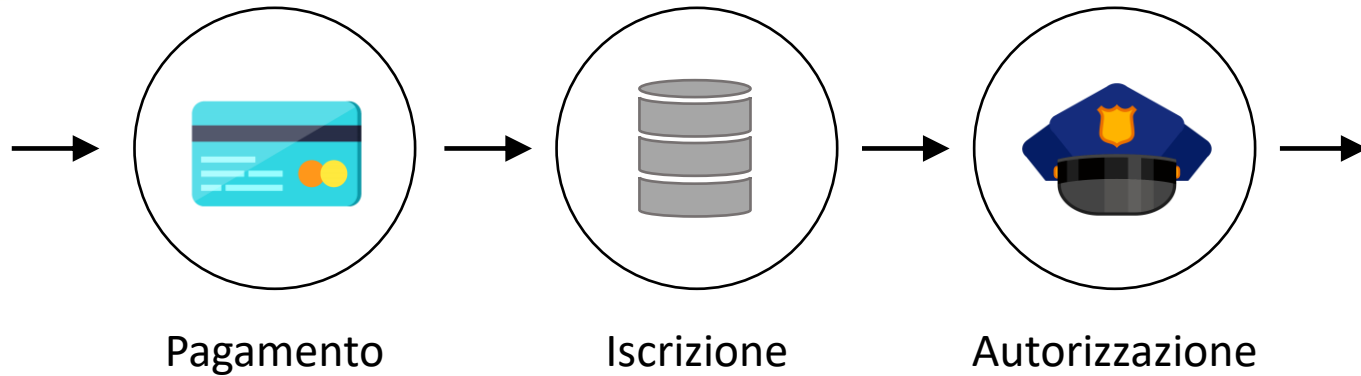

Accesso alle lezioni di un corso

Iscriviti

di Miranda Milanesi
★★★★★
EUR 10,99



Fai una domanda



Lezione 1

Durata stimata: 00:04:04

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia id est laborum et dolorum fuga. Et harum quidem rerum facilis expedita distinctio.

Happy little clouds <3



[← torna al corso](#)

DateTime.UtcNow

Tempo coordinato universale

Mi dà l'ora corrente secondo il fuso orario UTC, scelto come riferimento globale e da cui dipendono tutti gli altri fusi orari.

DateTime.Now	2021-07-14 21:00:00
DateTime.UtcNow	2021-07-14 19:00:00
DateTimeOffset.Now	2021-07-14 21:00:00+02:00

Realizzare un policy provider personalizzato

AuthorizationPolicy

- Decide quale policy restituire a fronte di un nome;
- La policy in questione può essere creata "al volo" anziché essere pre-registrata dalla classe Startup.

```
public class MultiAuthorizationPolicyProvider : DefaultAuthorizationPolicyProvider {  
    public override async Task<AuthorizationPolicy> GetPolicyAsync(string policyName) {  
        // Qui creo al volo la policy "combinata"  
    }  
}
```

- Lo registro dalla classe Startup

```
services.AddSingleton<IAuthorizationPolicyProvider, MultiAuthorizationPolicyProvider>();
```

Fai in modo che gli utenti ci diano
il loro numero di carta di credito

Raccogliamo noi i dati della carta di credito?

Numero di carta di credito

0000-0000-0000-00000

Scadenza

gg/mm

Codice di sicurezza

CVV/CVV2

Nominativo

NOME COGNOME

INVIA

Payment Card Industry Data Security Standard

Gli esercenti che trattano carte di credito devono garantire:

1. Sviluppo e gestione di reti e sistemi sicuri;
2. Protezione dei dati dei titolari di carta;
3. Attuare un programma per la gestione delle vulnerabilità;
4. Implementazione di misure rigide di controllo accessi;
5. Monitoraggio e test regolari delle reti;
6. Gestire una politica di sicurezza delle informazioni.

...un impegno non da poco!

Servizi di pagamento esterni (gateway)

Garantiscono al posto nostro che i dati della carta di credito dell'utente verranno trattati nel rispetto delle normative.



3,4% + 0,35 €

www.paypal.com



1,4% + 0,25 €

www.stripe.com



Accidenti che commissioni!

Sì, ma non dobbiamo conformarci
al PCI DSS, che sarebbe più gravoso

POS virtuali

Danno il vantaggio che i soldi vengono direttamente accreditati sul conto corrente. Chiedere alla banca quali sono le tariffe.



Ok, sento che mi dice la banca.

Checkout

Da MyCourse



[←](#) [MyCourse](#) TEST MODE

L'arte dell'Ikebana

14,00 €

Powered by **stripe**

[Termini](#) [Privacy](#)

G Pay

O paga con carta

Email

Informazioni sulla carta

1234 1234 1234 1234

VISA

MM / AA

CVC

11

Titolare della carta

Paese o regione

Italia

☐

Salva le informazioni per pagare più velocemente la prossima volta

i

Paga 14,00 €

Torna a MyCourse



Implementazione di PayPal Checkout

- Numero ordine?
- Cosa sta comprando?
- Quanto deve pagare?
- Dove lo reindirizzo?

Iscriviti

Pay

Informazioni sulla carta

1234 1234 1234 1234

MM / AA CVC

Titolare della carta

Paga 14,00 €

sandbox.paypal.com

AUTORIZZA
il pagamento

token

CATTURA
il pagamento

Subscribe

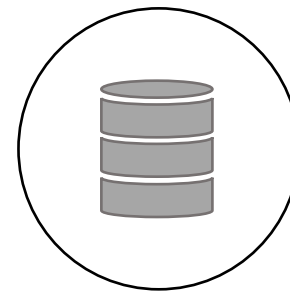
di Miranda Milanese



EUR 10,99



Pagamento



Iscrizione



Autorizzazione

Paypal Checkout

Installare la SDK

```
dotnet add package PayPalCheckoutSdk  
dotnet add package PayPalHttp
```

Creare una sandbox dal portale sviluppatori

<https://developer.paypal.com>

Seguire la documentazione

<https://developer.paypal.com/docs/api/overview/>

Sandbox di Paypal

- E' l'ambiente di test accessibile da www.sandbox.paypal.com
- È uguale in tutto e per tutto al "normale" paypal.com ma tutte le operazioni che possiamo fare dalla sandbox sono finte.
- Ci permette di sperimentare in libertà.



Stripe Checkout

Installare la SDK

```
dotnet add package Stripe.net
```

Accedere alla dashboard per ottenere la chiave segreta

<https://dashboard.stripe.com/>

Seguire la documentazione

<https://stripe.com/docs/api>

Generare URL in maniera imperativa

Usiamo il servizio `LinkGenerator`

```
string url = linkGenerator.GetUriByAction(httpContextAccessor.HttpContext,  
"https://localhost:5001/Courses/Subscribe/1"  
    action: "Subscribe",  
    controller: "Courses",  
    values: new { id = courseId });
```


C# Startup.cs

25

26

You, seconds ago | 2 authors (You and others)

27 namespace MyCourse

28 {

1 reference | You, seconds ago | 2 authors (You and others)

29 public class Startup

30 {

0 references

31 public void ConfigureServices(IServiceCollection services)

32 {

33 // Servizi di pagamento: abilito Stripe

34 // services.AddTransient<IPaymentGateway, PaypalPaymentGateway>();

35 services.AddTransient<IPaymentGateway, StripePaymentGateway>();

36

37

38

39

40

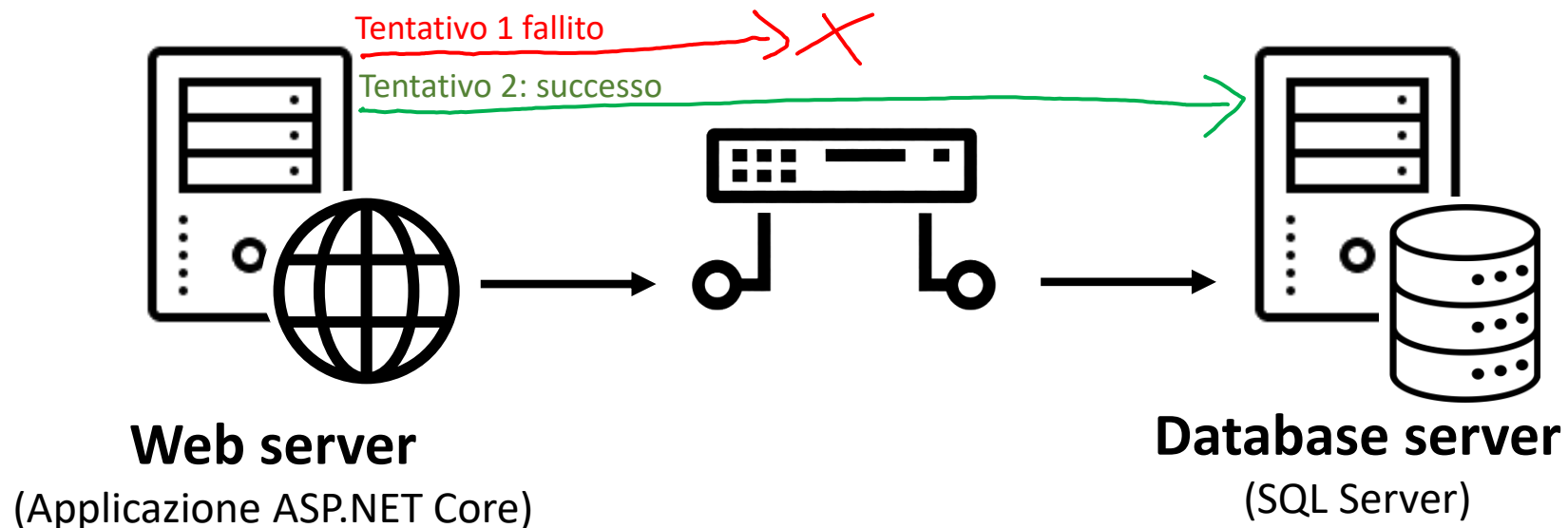
41

42

43

Errori transienti

- Possono impedire la corretta connessione al database;
- Sono momentanei e si «risolvono da soli» semplicemente riprovando.



Mitigare gli errori transienti

- Implementare la «connection resiliency», ovvero: se fallisce, ci riprovo.
- Implementare una «fallback strategy», ovvero: scrivo da un'altra parte (ad esempio su un file di testo a cui accederemo in caso di mancata iscrizione – meglio che perdere i dati della transazione definitivamente).

Connection Resiliency con ADO.NET

Se fallisce, ci riprovo (per un numero massimo di tentativi).

```
private async Task<SqliteConnection> GetOpenedConnection(Cancellation token)
{
    int maxRetries = 3;
    while (true)
    {
        try
        {
            maxRetries--; // Consumo un tentativo
            SqliteConnection conn = new(connectionStringOptions.CurrentValue.Default);
            await conn.OpenAsync(token);
            return conn;
        }
        catch (Exception) when (maxRetries > 0)
        {
            await Task.Delay(1000); // Se ho ancora tentativi, aspetto 1 secondo e ci riprovo
        }
    }
}
```

Connection Resiliency con EFCore

Supporta già questo comportamento, basta abilitarlo dalla classe Startup
Usiamo il metodo EnableRetryOnFailure.

```
services.AddDbContextPool<MyCourseDbContext>(optionsBuilder => {  
    string connectionString = Configuration.GetConnectionString("Default");  
    optionsBuilder.UseSqlServer(connectionString, options =>  
    {  
        options.EnableRetryOnFailure(3); // numero di tentativi  
    });  
});
```

2021-07-30	14:58:02	Paypal	AABBCCDDEEFF	EUR	15,40	CourseId	1	UserId	aabbccdd	CAPTURED
2021-07-30	14:59:15	Paypal	GGHHIIJJKKLL	EUR	12,30	CourseId	6	UserId	eeffgghh	CAPTURED
2021-07-30	15:05:21	Paypal	AABBCCDDEEFF	EUR	11,20	CourseId	4	UserId	iijjkkll	CAPTURED
2021-07-30	15:15:54	Paypal	AABBCCDDEEFF	EUR	22,60	CourseId	9	UserId	mmnnoopp	CAPTURED
2021-07-30	15:18:21	Paypal	AABBCCDDEEFF	EUR	21,10	CourseId	2	UserId	qqrrsstt	CAPTURED
2021-07-30	15:20:46	Paypal	AABBCCDDEEFF	EUR	10,90	CourseId	5	UserId	uuvvwvwx	CAPTURED
2021-07-30	15:22:56	Paypal	AABBCCDDEEFF	EUR	14,20	CourseId	6	UserId	yyzzaabb	CAPTURED
2021-07-30	15:23:42	Paypal	AABBCCDDEEFF	EUR	11,50	CourseId	4	UserId	ccddeeff	CAPTURED

Punto C

Tracciare su file di log le operazioni più importanti, come l'esito delle transazioni monetarie

Il tuo voto



L'arte dell'Ike

Lorem ipsum dolor sit amet con
eum incidunt repudiandae bea
inventore dolores molestias? P

itaque non, quasi cum voluptates quidem repudiandae doloribus? Autem mollitia esse odio nihil
atque non ea quisquam consequuntur exercitationem? Amet! Non ut itaque qui tempore illum!
Amet, accusamus minima. Ut rerum praesentium obcaecati sint, accusantium maxime odio
voluptatibus quaerat repudiandae corrupti magnam, non perferendis. Officia recusandae delectus
dolor quidem reprehenderit! Dolores eos eveniet quod molestiae praesentium earum fugit
similique fugiat? Molestias veniam eos enim! Ad, id. Rem similique explicabo deleniti possimus
facilis rerum deserunt minus aperiam suscipit! Ipsa, id laudantium. Rem distinctio ex magni unde
doloremque a, quae nesciunt, obcaecati animi perspiciatis earum, vel consectetur pariatur
tempora dicta. Quos architecto delectus, quis nostrum repudiandae molestiae quas distinctio
atque cupiditate temporibus? Deserunt optio molestias alias aspernatur. Ducimus veniam

Sei iscritto al corso

di Anastasia Trentini



EUR 12,99

~~EUR 15,99~~



Punto 10

Lo studente può esprimere un voto da 1 a 5 sul corso ma solo se è iscritto a quel corso

Generare una ricevuta PDF con Rotativa (1/2)

- Installare il pacchetto `Rotativa.AspNetCore`
- Scaricare i file binari di `wkhtmltopdf`
<https://wkhtmltopdf.org/downloads.html>
- Salvare i file binari nella directory `Rotativa` dell'applicazione
- Configurare `Rotativa`

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    RotativaConfiguration.Setup(env.ContentRootPath);
    // ...
}
```


Generare una ricevuta PDF con Rotativa (2/2)

Creare un'action MVC da cui restituire un oggetto ViewAsPdf

```
[Authorize(Policy = nameof(Policy.CourseSubscriber))]  
public async Task<IActionResult> Receipt(int id) {  
    CourseSubscriptionViewModel viewModel = await courseService.GetCourseSubscriptionAsync(id);  
  
    return pdf = new ViewAsPdf {  
        Model = viewModel,  
        ViewName = nameof(Receipt),  
        PageMargins = new Margins { Top = 10, Left = 10, Right = 10, Bottom = 20 },  
        PageSize = Size.A4,  
        PageOrientation = Orientation.Portrait,  
        FileName = $"{viewModel.Title} - ricevuta iscrizione.pdf"  
    };  
}
```

Progresso nella specifica

- Punto 8: iscrizione al corso;
- Punto 9: pagamento online;
- Punto 10: valutazione del corso;
- Punto c) log delle transazioni.

Requisiti funzionali

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23		

Requisiti non funzionali

a	b	c	d
---	---	---	---