

Sezione 16

Validazione e persistenza dei dati

Modifica corso

Salva



Web marketing facile

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et minima sunt quia nulla voluptate, illum eum incidunt repudiandae beatae, vero accusantium minus hic eveniet omnis laborum architecto inventore dolores molestias? Placeat sequi sapiente hic culpa optio quisquam est fugiat dolorem 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 quibusdam, sit saepe illum officiis obcaecati dolore atque totam consequatur exercitationem facilis similique magnam esse et consectetur non temporibus pariatur quae culpa iure? Asperiores reprehenderit, dolores rerum, impedit perferendis voluptatem vero aspernatur odit ipsa possimus nobis. Corrupti harum velit, totam delectus perspiciatis aut necessitatibus odio quasi quisquam, suscipit culpa laborum numquam, voluptatibus vel. Omnis minima quam explicabo deleniti quos accusamus magni provident soluta ex molestias impedit commodi reiciendis, enim rem assumenda sunt pariatur minus praesentium, exercitationem porro dolor. A nam esse recusandae id?

Email di contatto

tutor@example.com

Prezzo intero

EUR ▼ 19.99

Prezzo corrente

EUR ▼ 17.99


Immagine





[Cambia immagine](#)

Task-based UI


Info personali

**Accedi senza password**
Microsoft Authenticator
[Altre info >](#)

**Modifica la password**
Crea una password più sicura
[Modifica >](#)

**Gestisci indirizzi**
Informazioni su fatturazione e spedizione
[Gestisci >](#)


[Profilo](#)[Informazioni contatto](#)




[Cambia immagine](#)

Moreno Gentili


[Modifica il nome](#)




email@example.com



99/99/9999



Italia



italiano (Italia)

[Gestisci il modo in cui accedi a Microsoft](#)


[Modifica data di nascita](#)


[Modifica paese/area geografica](#)


[Modifica lingua di visualizzazione](#)

Task-based UI


Info personali

**Accedi senza password**
Microsoft Authenticator
[Altre info >](#)

**Modifica la password**
Crea una password più sicura
[Modifica >](#)

**Gestisci indirizzi**
Informazioni su fatturazione e spedizione
[Gestisci >](#)

[Profilo](#) [Informazioni contatto](#)




[Cambia immagine](#)

Nome

Cognome

Nuovo | Audio




Salva


Annulla

Task-based UI


Info personali



Accedi senza password
Microsoft Authenticator
[Altre info >](#)



Modifica la password
Crea una password più sicura
[Modifica >](#)



Gestisci indirizzi
Informazioni su fatturazione e spedizione
[Gestisci >](#)

Profilo

Informazioni contatto

Posta elettronica

Questi indirizzi e-mail sono associati al tuo account Microsoft.

email@example.com

Il tuo indirizzo e-mail di accesso

[Gestisci le autorizzazioni per le comunicazioni](#)

[Gestisci questo indirizzo e-mail](#)

Telefono

Questi numeri di telefono sono associati al tuo account Microsoft.

[Proteggi il tuo account anche senza password](#)

+39 999 999 9999

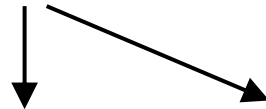
Il numero utilizzato per scaricare app e connettere il telefono e il PC

[Rimuovi questo numero di telefono](#)

Attributo asp-for

- Lo usiamo sui tag helper come input, select, textarea;
- Lo valorizziamo col nome di una proprietà del model;

```
<input type="text" asp-for="Title" placeholder="Digita un titolo">
```



```
<input type="text" name="Title" id="Title" placeholder="Digita un titolo">
```

- Previene gli errori di digitazione.

```
Views\Courses\Create.cshtml(6,135): error CS1061: 'CourseCreateInputModel'  
does not contain a definition for 'TITLE' and no accessible extension method  
'TITLE' accepting a first argument of type 'CourseCreateInputModel'
```

Disambiguazione delle action

```
public IActionResult Create()  
{  
    //Qui mostriamo il form all'utente  
}
```

L'attributo [HttpPost] aiuta il meccanismo di routing a disambiguare la selezione dell'action

```
[HttpPost]  
public IActionResult Create(CreateCourseInputModel inputModel)  
{  
    //Qui chiamiamo il servizio applicativo...  
    //...e reindirizziamo alla pagina di elenco  
}
```

Catalogo corsi

[+ Crea nuovo](#)

Titolo

Valutazione ▼

Prezzo

**Web marketing facile***di Mario Rossi*

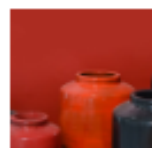
EUR 17.99

~~EUR 19.99~~[Dettagli](#)**L'ABC del fai da te***di Mario Rossi*

EUR 7.99

[Dettagli](#)**Come fare gli origami***di Mario Rossi*

EUR 30.99


~~EUR 34.99~~[Dettagli](#)**Corso di ceramica***di Mario Rossi*

USD 10.99

~~USD 11.99~~[Dettagli](#)

Nuovo corso


Digita un titolo che attiri gli studenti...

 Crea corso!

Inserire una riga e recuperare l'ID con SQL

Comando per l'inserimento di una riga:

```
INSERT INTO Courses (Title, Author) VALUES ('Mio Corso', 'Mario Rossi');
```




Query per il recupero dell'ID:

Tecnologia database	Query
SQLite	SELECT last_insert_rowid();
SQL Server	SELECT SCOPE_IDENTITY();
MySql	SELECT LAST_INSERT_ID();
Oracle	INSERT INTO ... RETURNING Id INTO :param1;
PostgreSQL	INSERT INTO ... RETURNING Id INTO @param1;

Model binding di oggetti complessi

Nuovo corso

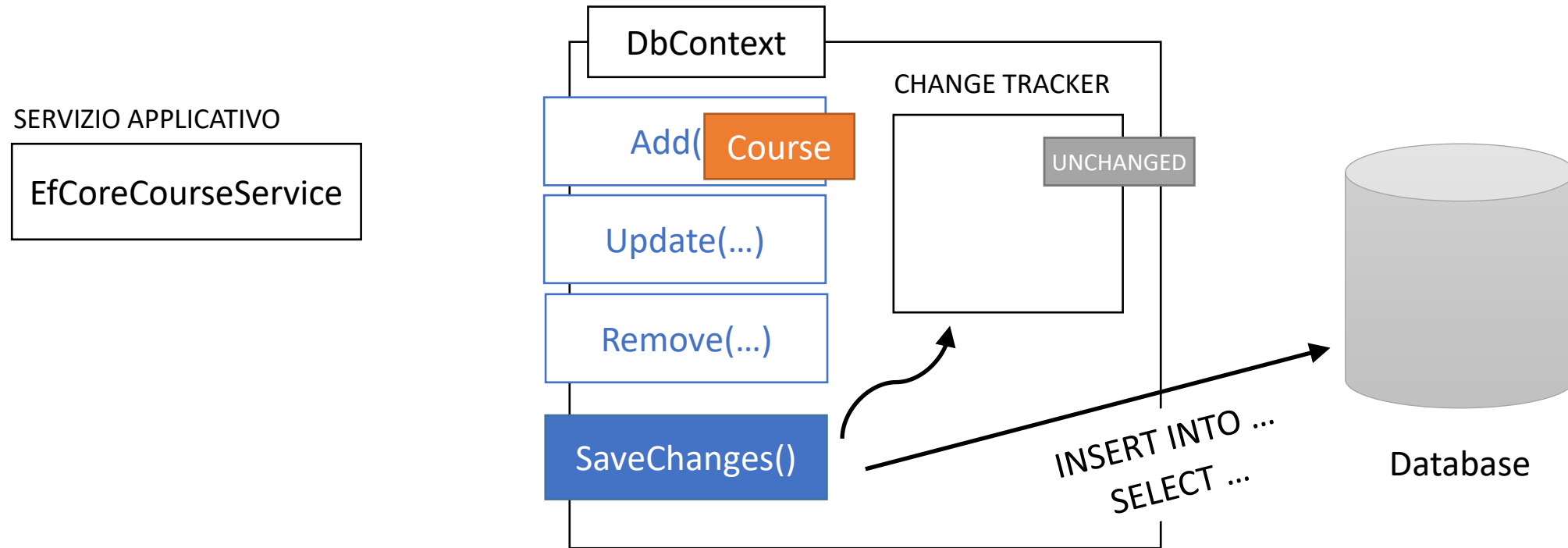
Digita un titolo che attiri gli studenti...

 Crea corso!

```
<input type="text" name="Title" id="Title">
```

```
[HttpPost]
public IActionResult Create
    (CourseCreateInputModel inputModel)
{
    //...
    return RedirectToAction(nameof(Index));
}
```

Persistenza dei dati con EFCore



Il change tracker di EFCore

Ogni entità nel change tracker assume uno di questi 4 stati:



Detached è lo stato assunto dalle entità che abbiamo recuperato con AsNoTracking().

```
dbContext.Entry(course).State = EntityState.
```

- Added
- Deleted
- Detached
- Modified
- Unchanged

Persistere più entità

- Il DbContext è in grado di persistere più di un'entità alla volta.

```
dbContext.Add(course1);  
dbContext.Add(course2);  
dbContext.Remove(course3);  
await dbContext.SaveChangesAsync();
```

- Tutti i comandi SQL vengono eseguiti nel contesto di una transazione.

"Validazione" o "sanitizzazione"?

va • li • da • zió • ne

Controllo della validità e della correttezza di dati realizzato attraverso il confronto con regole e dati già noti e attendibili.

sa • ni • tiz • za • zió • ne

Operazione mediante la quale si rende igienicamente idoneo un impianto o un ambiente.

"Validazione" o "sanitizzazione"?

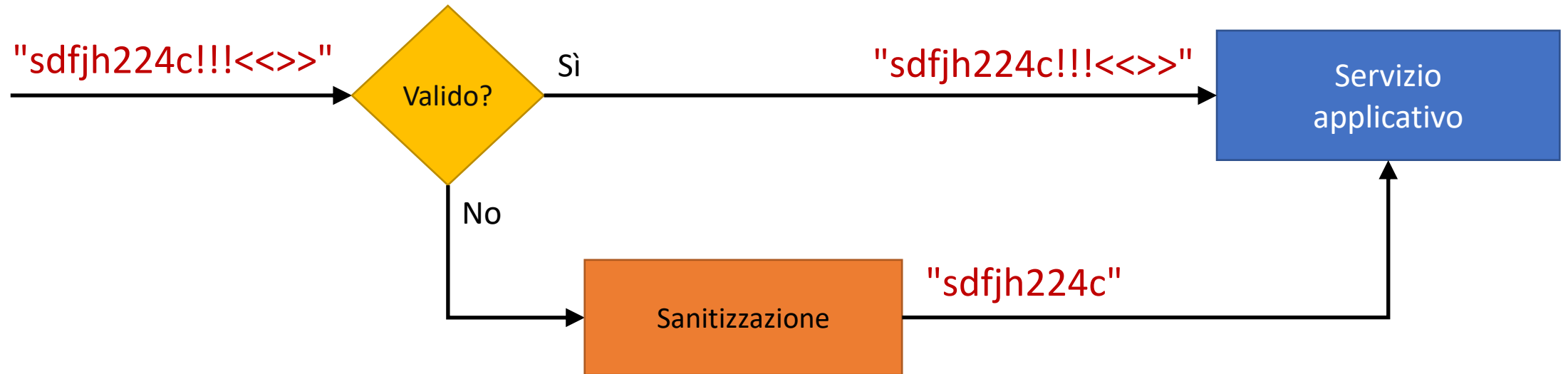
Sono entrambe importanti!

```
if (!orderOptions.Allow.Contains(orderby))  
{  
    orderby = orderOptions.By;  
    ascending = orderOptions.Ascending;  
}
```

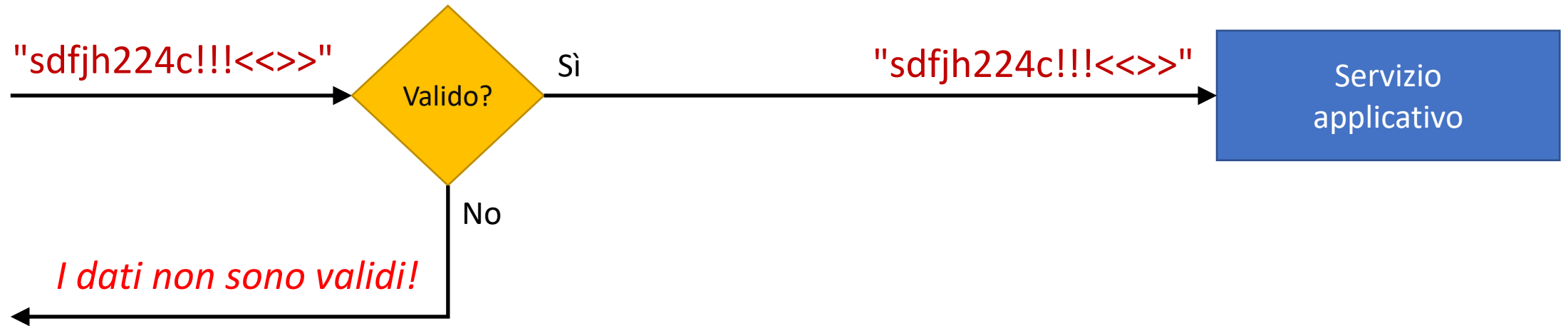
Validazione

Sanitizzazione

Validazione + sanitizzazione...



... o solo validazione?



Il ModelState

- Riporta gli errori di validazione verificatisi nel model binding.
- La sua proprietà **IsValid** è **false** se c'è almeno un errore.

```
[HttpPost]
public async Task<IActionResult> Create(CourseCreateInputModel inputModel)
{
    if (!ModelState.IsValid)
    {
        //...
    }
}
```

- Se vogliamo recuperare tutti gli errori (es. per scriverli nel log)

```
IEnumerable<ModelError> errors = ModelState.Values.SelectMany(value => value.Errors);
```

Regole di validazione: le Data Annotation

- Sono attributi che ci permettono di definire delle regole.
- Il titolo del corso è obbligatorio, quindi poniamo l'attributo **Required** sulla proprietà **Title**.

```
public class CourseCreateInputModel
{
    [Required]
    public string Title { get; set; }
}
```

- Nel ModelState sono trascritti gli errori di validazione, cioè le violazioni alle regole.

Data Annotation

- Si trovano nel namespace `System.ComponentModel.DataAnnotations`

Data Annotation	Scopo
Required	Rende obbligatoria l'immissione del valore
Range (1900, 2019)	Impone che il valore numerico sia entro un certo intervallo
MinLength (10), MaxLength (100)	Regolano la lunghezza minima e massima di una stringa
RegularExpression ("^[a-z]+\$")	Verifica che una stringa sia conforme a un'espressione regolare
Compare (nameof(<i>AltraProprietà</i>))	Verifica se il valore è uguale a quello di un'altra proprietà
CreditCard , EmailAddress , Phone , Url	Impongono un formato noto
Remote ("Action", "Controller")	Utile nella validazione client

Data Annotation

- Possiamo usarne più d'una sulla stessa proprietà

```
public class CourseCreateInputModel
{
    [Required,
    MinLength(10), MaxLength(100),
    RegularExpression(@"^[\\w\\s\\.]+$")]
    public string Title { get; set; }
}
```

Regole di validazione

- Titolo obbligatorio;
- Tra 10 e 100 caratteri;
- Solo lettere e numeri;
- Anche punto e spazio;
- Non deve esistere già.

asp-validation-for

È un tag helper che mostra gli errori di validazione di una proprietà specifica del modello.

```
<span asp-validation-for="Title"></span>
```

Può trovarsi ovunque nella view, ma ha senso metterlo nei pressi dell'elemento `input`.

Personalizzare il testo degli errori

- Si valorizza la proprietà ErrorMessage della data annotation.

```
public class CourseCreateInputModel
{
    [Required(ErrorMessage = "Il titolo è obbligatorio"),
    MinLength(10, ErrorMessage = "Il titolo dev'essere di almeno {1} caratteri"),
    MaxLength(100, ErrorMessage = "Il titolo dev'essere di al massimo {1} caratteri"),
    RegularExpression(@"^[\\w\\s\\.]+$", ErrorMessage = "Titolo non valido")]
    public string Title { get; set; }
}
```

- Per siti multilingua è preferibile tenere i testi su file di risorse esterni e usare invece le proprietà ErrorMessageResourceName e ErrorMessageResourceType.

asp-validation-summary

È un tag helper che mostra TUTTI gli errori di validazione.

```
<div asp-validation-summary="All"></div>
```

VALORI AMMESSI

- **All**: visualizza tutti gli errori di validazione;
- **None**: non visualizza alcun errore di validazione;
- **ModelOnly**: visualizza solo gli errori del modello (cioè non legati a una specifica proprietà).

Aggiungere errori di validazione personalizzati

```
ModelState.AddModelError("Title", "Il titolo già esiste");
```

Riguarda una proprietà, perciò non verrebbe visualizzato con ModelOnly

```
ModelState.AddModelError("", "Non è possibile creare questo corso");
```

Riguarda il modello nel suo complesso e perciò verrà visualizzato con ModelOnly

asp-for

```
<input type="text" asp-for="Title" placeholder="Digita un titolo">
```

- Imposta gli attributi id e name;
- Imposta gli attributi di validazione (secondo le data annotation);
- Imposta l'attributo value.

```
<input type="text" class="form-control form-control-lg" placeholder="Digita  
un titolo che attiri gli studenti..." data-val="true" data-val-maxlength="Il titolo  
dev'essere di al massimo 100 caratteri" data-val-maxlength-max="100" data-val-  
minlength="Il titolo dev'essere di almeno 10 caratteri" data-val-minlength-min="10"  
data-val-regex="Titolo non valido" data-val-regex-pattern="^[\\w\\s\\.]+$" data-val-  
required="Il titolo è obbligatorio" id="Title" name="Title" maxlength="100"  
value="!!!!!">
```

Librerie necessarie per validazione client

- **jQuery**
- **jQuery validation**

*Usando il
provider unkpg*

<https://cdnjs.com/libraries/jquery-validate>

```
libman install jquery-validation@1.19.1 --destination wwwroot/lib/jquery-validation --files  
dist/jquery.validate.min.js --files dist/localization/messages_it.min.js --files dist/additional-  
methods.min.js
```

- **jQuery validation unobtrusive**

<https://cdnjs.com/libraries/jquery-validation-unobtrusive>

```
libman install jquery-validation-unobtrusive@3.2.11 --destination wwwroot/lib/jquery-  
validation-unobtrusive --files dist/jquery.validate.unobtrusive.min.js
```

Data Annotation personalizzata (definizione)

- Si crea una classe che deriva da `ValidationAttribute` e si fa l'override del metodo `IsValid`.

```
public class NotNullAttribute : ValidationAttribute
{
    protected override ValidationResult IsValid(object value, ValidationContext validationContext)
    {
        if (value != null)
        {
            return ValidationResult.Success;
        }
        return new ValidationResult(ErrorMessage, new[] { validationContext.MemberName });
    }
}
```

Data Annotation personalizzata (uso)

- Si crea una classe che deriva da `ValidationAttribute` e si fa l'override del metodo `IsValid`.

```
public class CourseCreateInputModel
{
    [Required(ErrorMessage = "Il titolo è obbligatorio"),
    MinLength(10, ErrorMessage = "Il titolo dev'essere di almeno {1} caratteri"),
    MaxLength(100, ErrorMessage = "Il titolo dev'essere di al massimo {1} caratteri"),
    RegularExpression(@"^[\\w\\s\\.]+$", ErrorMessage = "Titolo non valido"),
    NotNull(ErrorMessage = "Il titolo non deve essere null")]
    public string Title { get; set; }
}
```

Data Annotation personalizzata

- Ideale per controlli sintattici quando nessuna delle altre Data Annotation ci aiuta.

Esempio: controllo della validità della partita IVA

0123456789C

Una Data Annotation personalizzata non supporta dependency injection né esecuzione asincrona

```
public class UniqueCourseTitleAttribute : ValidationAttribute
{
    protected override ValidationResult IsValid(object value, ValidationContext validationContext)
    {
        //Ottengo un DbContext
        var context = validationContext.GetService(typeof(MyCourseDbContext)) as MyCourseDbContext;
        string title = value?.ToString().ToLower();

        //Verifico se il titolo esiste ma in maniera sincrona :(
        bool titleExists = context.Courses.AnyAsync(course => course.Title.ToLower() == title).Result;
        if (!titleExists)
        {
            return ValidationResult.Success;
        }
        return new ValidationResult(ErrorMessage, new[] { validationContext.MemberName });
    }
}
```


Validazione personalizzata a livello di modello

- Si implementa l'interfaccia `IValidatableObject` e si definisce il metodo `Validate`.

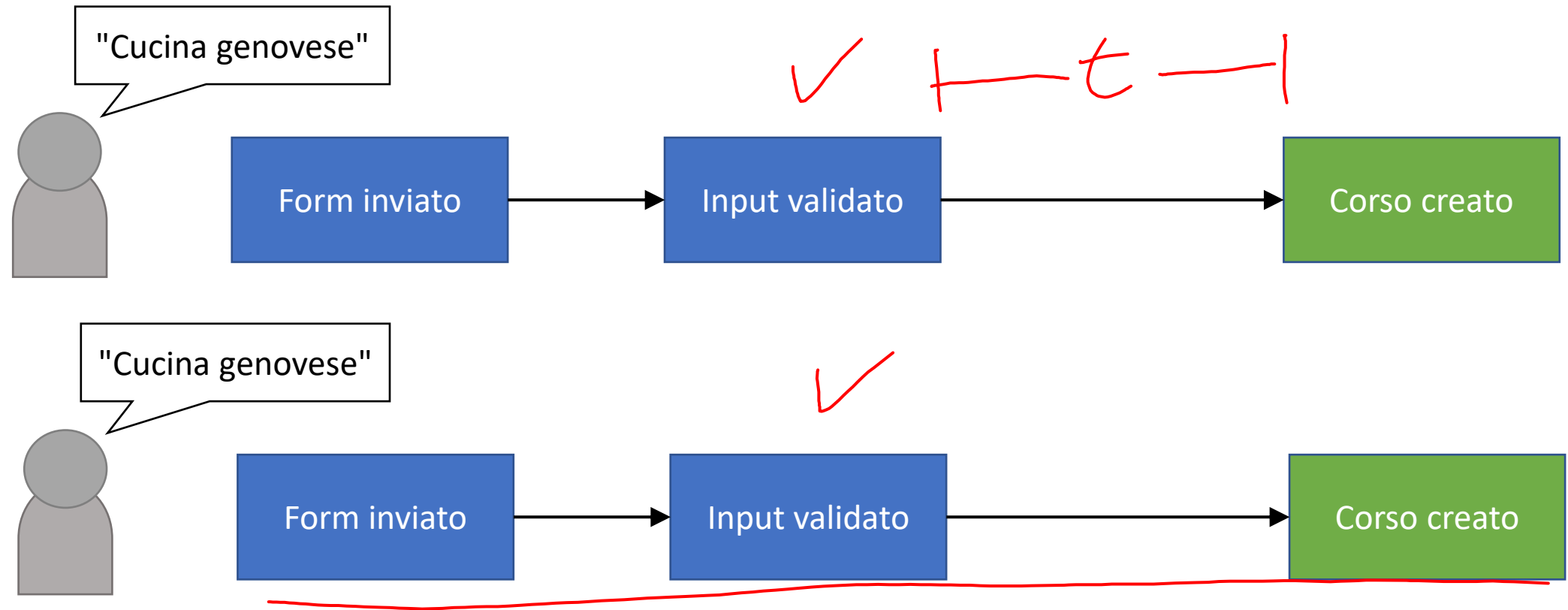
```
public class CourseCreateInputModel : IValidatableObject
{
    public string Title { get; set; }

    public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
    {
        if (Title != null)
        {
            yield return ValidationResult.Success;
        }
        yield return new ValidationResult("Il titolo non può essere null", new[] { nameof(Title) });
    }
}
```

Validazione personalizzata a livello di modello

- Ideale quando abbiamo logica complessa per il controllo sintattico di molteplici proprietà;
- Per preferenza personale, quando vogliamo tenere la logica di validazione all'interno dell'input model.

Problemi di concorrenza



Una soluzione pragmatica

Se vogliamo che nel database non possano esserci titoli duplicati, rendiamo impossibile la presenza di titoli duplicati.

→ Poniamo un vincolo UNIQUE sulla colonna Title della tabella Courses.

```
CREATE UNIQUE INDEX ux_title ON Courses (  
    Title COLLATE NOCASE  
);
```

Valido solo su SQLite

Verifica esistenza del titolo lato client: la Data Annotation Remote

Invierà una richiesta ajax per eseguire una verifica lato server.

- il form **non** verrà inviato e la pagina **non** sarà ricaricata.

```
public class CourseCreateInputModel
{
    [Remote(action: "IsTitleAvailable", controller: "Courses")]
    public string Title { get; set; }
}
```

Che edizione di jQuery stiamo usando?

Uncaught TypeError: Cannot read property 'apply' of undefined. Exception occurred when checking element Title, check the 'remote' method.

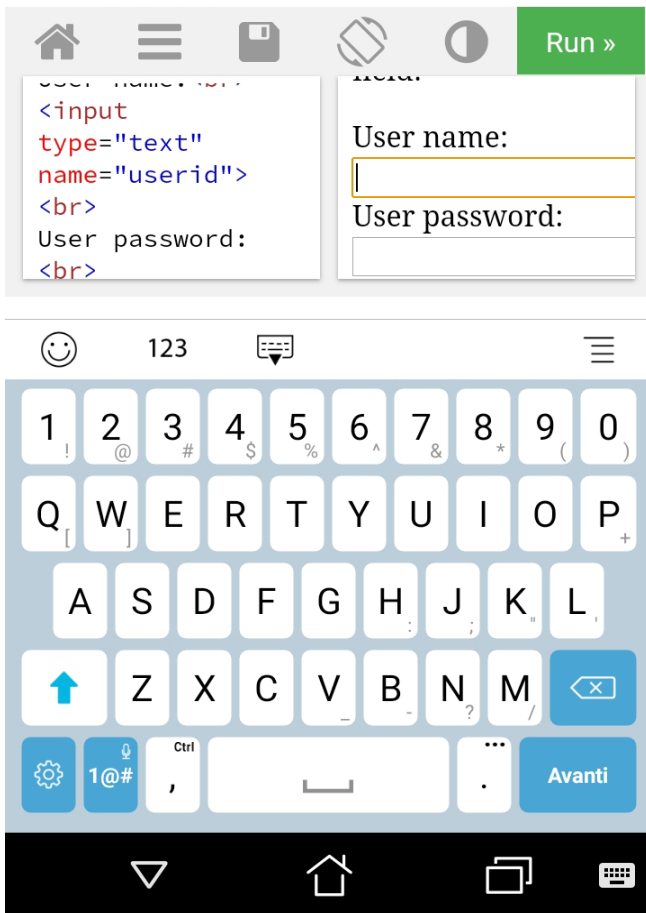
	jQuery	jQuery slim
Peso	30KB	24KB
Manipolazione del DOM	✓	✓
Modifica degli stili CSS	✓	✓
Gestione degli eventi	✓	✓
Utilità varie	✓	✓
Invio di richieste Ajax	✓	✗
Effetti di animazione	✓	✗

La validazione client **non** sostituisce quella server

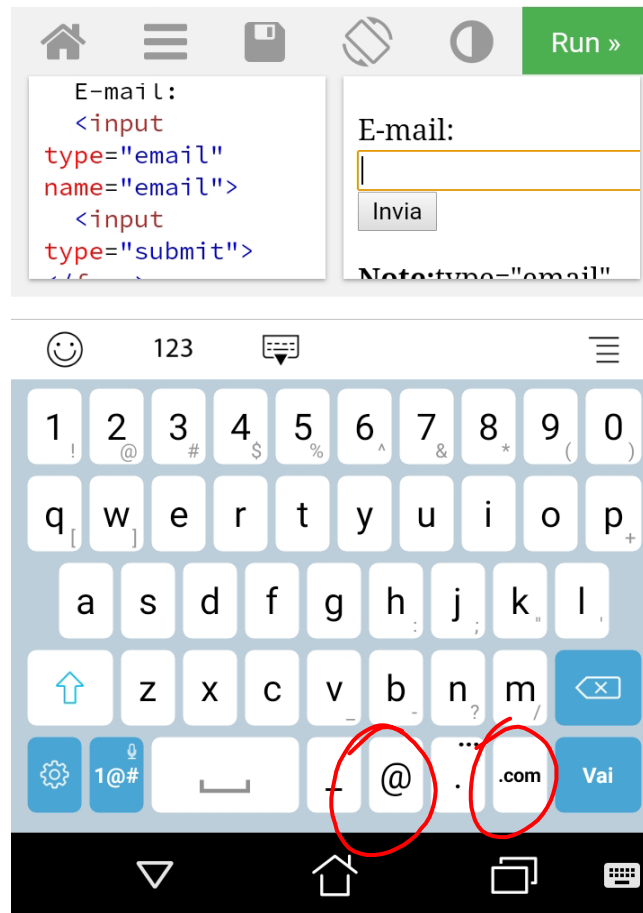
- Serve per aiutare l'utente ed evitargli trasferimenti inutili;
- È sempre necessario avere anche la validazione lato server;
- La validazione lato client può essere bypassata alterando il contenuto della pagina;
- Un malintenzionato può "forgiare" la richiesta HTTP come preferisce.

Su mobile: diverse tastiere virtuali in base al type

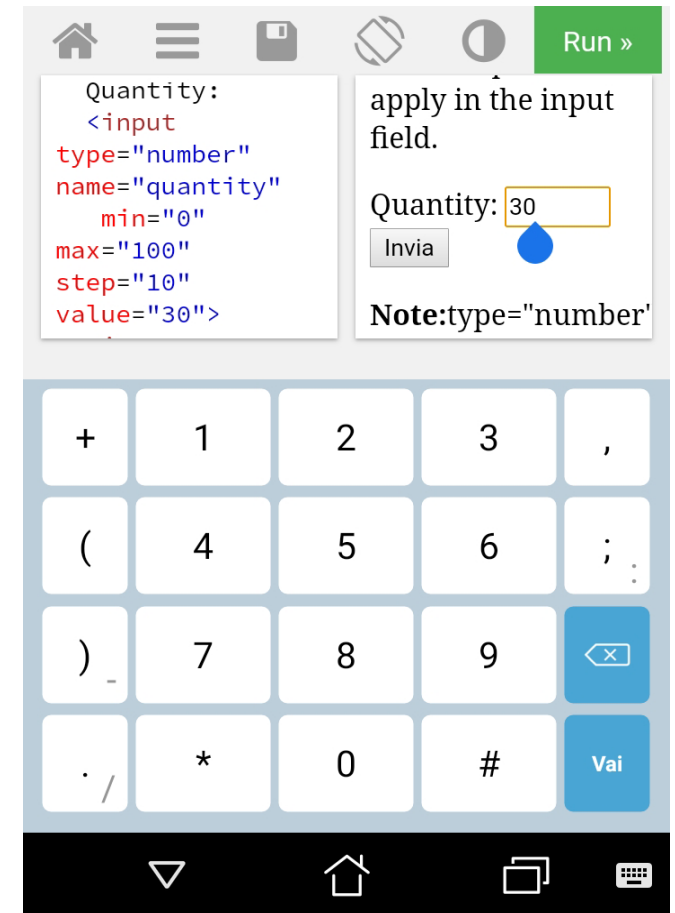
type="text"



type="email"



type="number"



Elenco dei `type` di una casella `input`

Testo

<code>text</code>	Testo semplice
<code>email</code>	Indirizzo e-mail
<code>tel</code>	Numero di telefono
<code>search</code>	Casella di ricerca
<code>password</code>	Testo mascherato da asterischi
<code>hidden</code>	Casella nascosta

Numeri

<code>number</code>	Numero intero o decimale
<code>range</code>	Selezione in un intervallo

Date e orari

<code>date</code>	Giorno, mese e anno
<code>datetime-local</code>	Data e ora
<code>week</code>	Settimana e anno
<code>month</code>	Mese e anno

Selezione

<code>checkbox</code>	Casella di spunta (scelta multipla)
<code>radio</code>	Casella di spunta (scelta singola)
<code>color</code>	Codice colore
<code>file</code>	Caricamento di un file

Bottoni

<code>button</code>	Bottone (non invia il form)
<code>submit</code>	Bottone (invia il form)
<code>image</code>	Immagine (invia il form)
<code>reset</code>	Resetta il form

Vari tipi di casella per i form HTML

```
<input type="text" asp-for="Title">
```

Web marketing facile

```
<input type="number" min="0" max="1000"  
step="any" asp-for="CurrentPrice.Amount">
```

17,99

```
<textarea asp-for="Description" rows="10">  
</textarea>
```

Testo descrittivo

```
<input type="email" asp-for="Email">
```

tutor@example.com

```
<select  
asp-items="@Html.GetEnumSelectList<Currency>()"  
asp-for="CurrentPrice.Currency"></select>
```

EUR ▼

EUR

USD

GBP

Input Model per la modifica

```
public class CourseEditInputModel
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Description { get; set; }
    public string ImagePath { get; set; }
    public string Email { get; set; }
    public Money FullPrice { get; set; }
    public Money CurrentPrice { get; set; }
}
```



```
public class CourseDetailViewModel
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Description { get; set; }
    public string ImagePath { get; set; }
    public string Author { get; set; }
    public double Rating { get; set; }
    public Money FullPrice { get; set; }
    public Money CurrentPrice { get; set; }
    public List<LessonViewModel> Lessons { get; set; }
    public TimeSpan TotalCourseDuration => //...
}
```

Display

È una Data Annotation che serve a indicare l'"etichetta".

Descrizione



Display

È una Data Annotation che serve a indicare l'"etichetta".

NELL'INPUT MODEL

```
public class CourseEditInputModel
{
    [Display(Name = "Descrizione")]
    public string Description { get; set; }
}
```

NELLA VIEW

```
<label asp-for="Description"></label> Produce
      ↘
      <label for="Description">Descrizione</label>
```

IValidatableObject

Per logiche di validazione complesse che coinvolgono più proprietà.

```
public class CourseEditInputModel : IValidatableObject
{
    public Money FullPrice { get; set; }
    public Money CurrentPrice { get; set; }

    public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
    {
        if (FullPrice.Amount < CurrentPrice.Amount)
        {
            yield return new ValidationResult("Prezzo errato", new [] { nameof(FullPrice) });
        }
    }
}
```

Aggiornare una riga con SQL

Si usa il comando UPDATE

```
UPDATE Courses SET Title = "Titolo", Description = "Testo"  
WHERE Id = 1;
```

Aggiornare una riga con Entity Framework Core

```
public async Task<CourseDetailViewModel> EditCourseAsync(CourseEditInputModel inputModel)
{
    var course = await dbContext.Courses.FindAsync(inputModel.Id);
    course.Title = inputModel.Title;
    dbContext.Update(course);
    await dbContext.SaveChangesAsync();
    //...
}
```


Invalidare la cache al salvataggio

IMemoryCache

```
memoryCache.Remove("Chiave");
```

IDistributedCache

```
distributedCache.Remove("Chiave");
```

```
await distributedCache.RemoveAsync("Chiave");
```

Ci sono solo due cose difficili in Computer Science:
invalidare la cache e dare i nomi alle cose.

Phil Karlton

TempData

- Funziona come il ViewData: serve a passare messaggi dal controller alla view;
- "Sopravvive" a un redirect, e poi il messaggio viene rimosso automaticamente.

NELL'ACTION DEL CONTROLLER

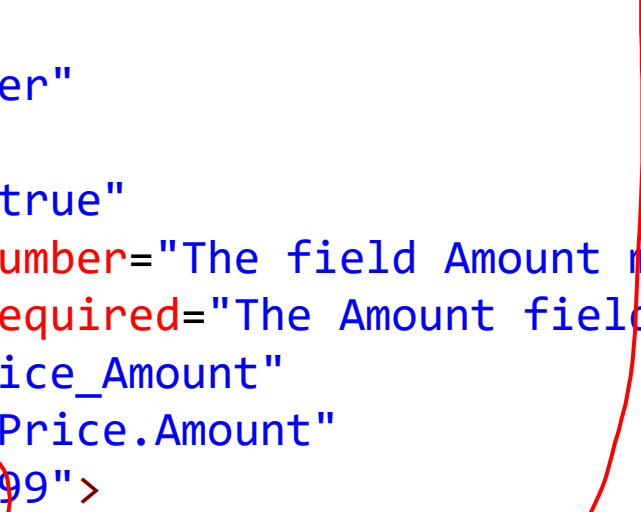
```
CourseDetailViewModel course = await courseService.EditCourseAsync(inputModel);  
TempData["ConfirmationMessage"] = "I dati sono stati salvati con successo";  
return RedirectToAction(nameof(Detail), new { id = inputModel.Id });
```

Attenzione alla *Culture*

```
<input type="number" step="any" asp-for="FullPrice.Amount">
```

*asp-for potrebbe usare
incorrettamente la virgola
come separatore dei
decimali*

```
<input  
  type="number"  
  step="any"  
  data-val="true"  
  data-val-number="The field Amount must be a number."  
  data-val-required="The Amount field is required."  
  id="FullPrice_Amount"  
  name="FullPrice.Amount"  
  value="17,99">
```



Culture dell'applicazione

Una *Culture* è l'insieme delle convenzioni usate da una particolare lingua, in una particolare regione del mondo.

	it-IT	it-CH	en-US
Lingua	Italiano	Italiano	English
Regione	Italia	Svizzera	United States
Separatore migliaia	.	'	,
Separatore decimali	,	.	.
Valuta	€	CHF	\$
Calendario	Gregoriano	Gregoriano	Gregoriano
Formato Data	g/m/a	g.m.a	m/g/a

Culture dell'applicazione

Un'applicazione ASP.NET Core, salvo diversa disposizione, usa la stessa *Culture* dell'utente del sistema operativo.

🏠 Language

Windows display language

English (United States)



17.99



🏠 Lingua

Lingua di visualizzazione di Windows

Italiano (Italia)



17,99



Soluzione 1. Impostare la Culture predefinita

Aggiungere il RequestLocalizationMiddleware dal metodo Configure della classe Startup.

```
var appCulture = new CultureInfo("en-US");  
app.UseRequestLocalization(new RequestLocalizationOptions  
{  
    SupportedCultures = new[] { appCulture },  
    DefaultRequestCulture = new RequestCulture(appCulture)  
});
```

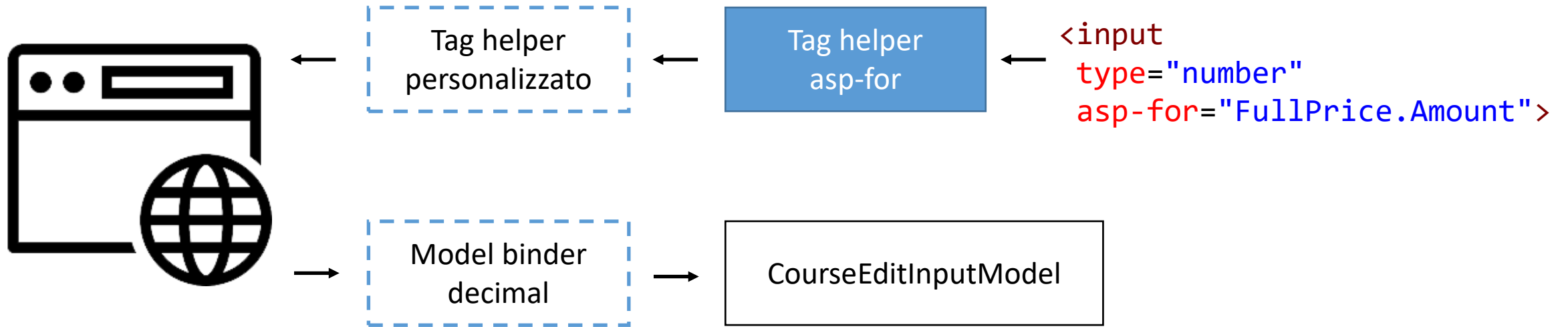
Soluzione 1. Impostare la Culture predefinita

Aggiungere il RequestLocalizationMiddleware dal metodo Configure della classe Startup.

```
var appCulture = CultureInfo.InvariantCulture;  
app.UseRequestLocalization(new RequestLocalizationOptions  
{  
    SupportedCultures = new[] { appCulture },  
    DefaultRequestCulture = new RequestCulture(appCulture)  
});
```

E' una Culture artificiale, le cui convenzioni non cambieranno mai

Soluzione 2. Tag helper e model binder personalizzati



Installare Summernote

Da CDN

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/summernote-bs4.min.css">  
<script src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/summernote-bs4.min.js"></script>  
<script src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/lang/summernote-it-IT.js"></script>
```

Con Libman

```
libman install summernote@0.8.15 --destination wwwroot/lib/summernote --files dist/font/summernote.eot --  
files dist/font/summernote.ttf --files dist/font/summernote.woff --files dist/font/summernote.woff2 --  
files dist/summernote-bs4.min.css --files dist/summernote-bs4.min.js --files dist/lang/summernote-it-IT.js
```

*Usando il
provider unkpg*

@Html.Raw

Stampa una stringa "tale e quale", senza fare l'HTML Encoding.

SENZA Html.Raw

```
@{  
    string testo = "<b>Ciao</b>";  
}  
<div>@testo</div>
```



<div>Ciao</div>

HTML Entities

Con Html.Raw

```
@{  
    string testo = "<b>Ciao</b>";  
}  
<div>@Html.Raw(testo)</div>
```



<div>Ciao</div>

Cross-Site Scripting (XSS)

OWASP TOP 10

Injection

Broken Authentication

Sensitive Data Exposure

XML External Entities (XXE)

Broken Access Control

Security Misconfiguration

Cross-Site Scripting XSS

Insecure Deserialization

Using Components with Known Vulnerabilities

Insufficient Logging & Monitoring

<https://owasp.org/www-project-top-ten/>

Cross-Site Scripting (XSS)

È una vulnerabilità che affligge siti web dinamici che impiegano un **insufficiente controllo** dell'input nei form.

https://it.wikipedia.org/wiki/Cross-site_scripting

<script> —————> <script>

Cross-Site Scripting (XSS)

È una vulnerabilità che affligge siti web dinamici che impiegano un insufficiente controllo dell'input nei form.

https://it.wikipedia.org/wiki/Cross-site_scripting



Furto d'identità



Ads



Malware



Sfruttare risorse

Mozilla Security

[Advisories](#)[Known Vulnerabilities](#)[Mozilla Security Blog](#)[Security Bug Bounty](#)

Client Bug Bounty

[Frequently Asked Questions](#)[Hall of Fame](#)

Web Bug Bounty

[Eligible Websites](#)[Frequently Asked Questions](#)[Hall of Fame](#)

Mozilla Foundation Security Advisory 2020-03

Security Vulnerabilities fixed in Firefox 72.0.1 and Firefox ESR 68.4.1

Announced January 8, 2020

Impact

critical

Products

Firefox, Firefox ESR

Fixed in

Firefox 72.0.1

Firefox ESR 68.4.1

CVE-2019-17026: IonMonkey type confusion with StoreElementHole and FallibleStoreElement

Reporter

Qihoo 360 ATA

Impact

critical

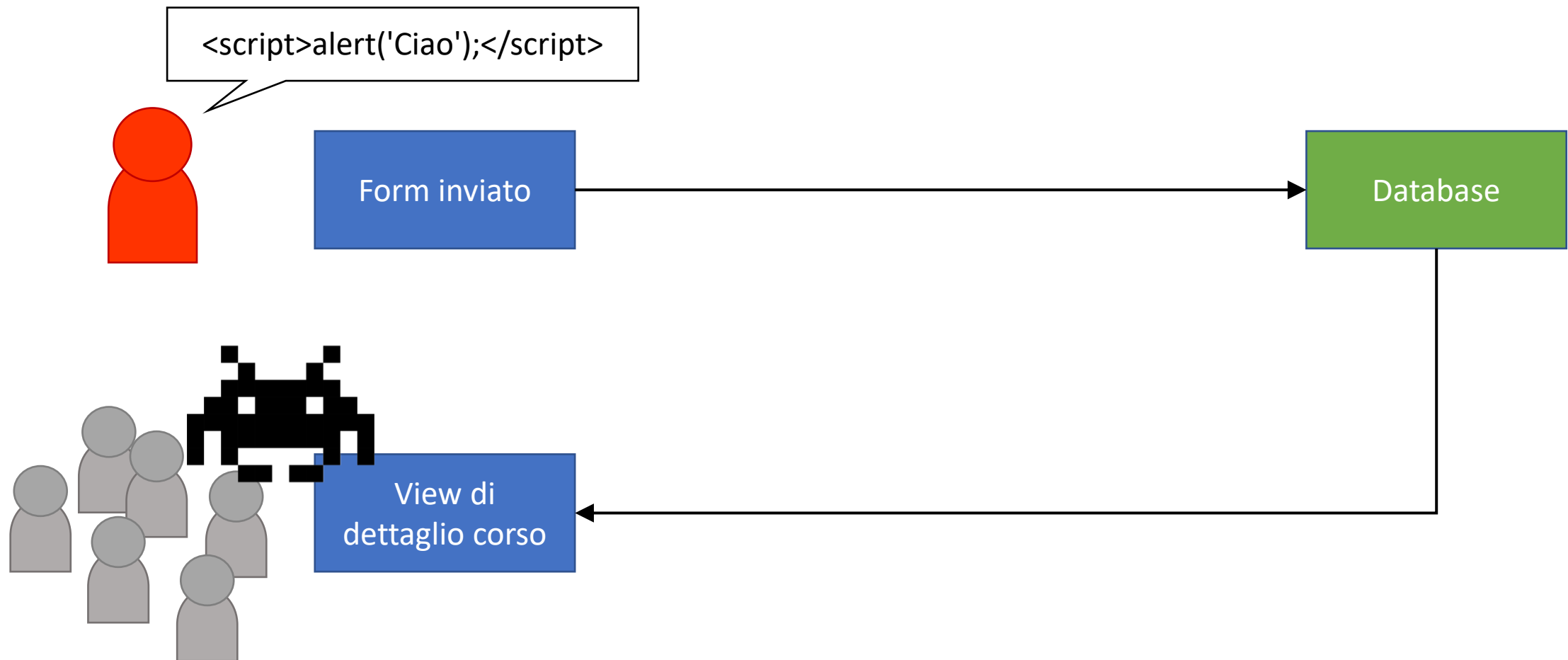
Description

Incorrect alias information in IonMonkey JIT compiler for setting array elements could lead to a type confusion. We are aware of targeted attacks in the wild abusing this flaw.

References

[Bug 1607443](#)

Proteggere l'applicazione da attacchi XSS



Form "normali" e "multipart"

Richiesta inviata da un form "normale"

senza attributo enctype

```
Id=32&Title=L%27arte+dell%27Ikebana&Description  
=Lorem+ipsum+dolor+sit...
```

Richiesta inviata da un form "multipart"

con enctype="multipart/form-data"






```
-----boundary  
Content-Disposition: form-data; name="Id"  
  
32  
  
-----boundary  
Content-Disposition: form-data; name="Title"  
  
L'arte dell'Ikebana  
  
-----boundary  
Content-Disposition: form-data; name="Image";  
filename="foto.jpg"  
Content-Type: image/jpeg  
  
???e?Exif???II*??
```

Caricare un'immagine (o altro tipo di file)

- Aggiungere al form l'attributo `enctype="multipart/form-data"`
- Aggiungere un elemento `<input type="file" asp-for="Image">`
- Aggiungere all'input model una proprietà di tipo `IFormFile`
- Validare e salvare l'immagine

IFormFile

- È un tipo di oggetto che rappresenta un file caricato dall'utente;
- Possiede vari membri pubblici:

-  FileName
-  Length
-  ContentType
-  CopyToAsync
-  OpenReadStream

Cache "busting"

Basta aggiungere l'attributo `asp-append-version="true"` ad ``

VIEW DETAIL.CSHTML

```

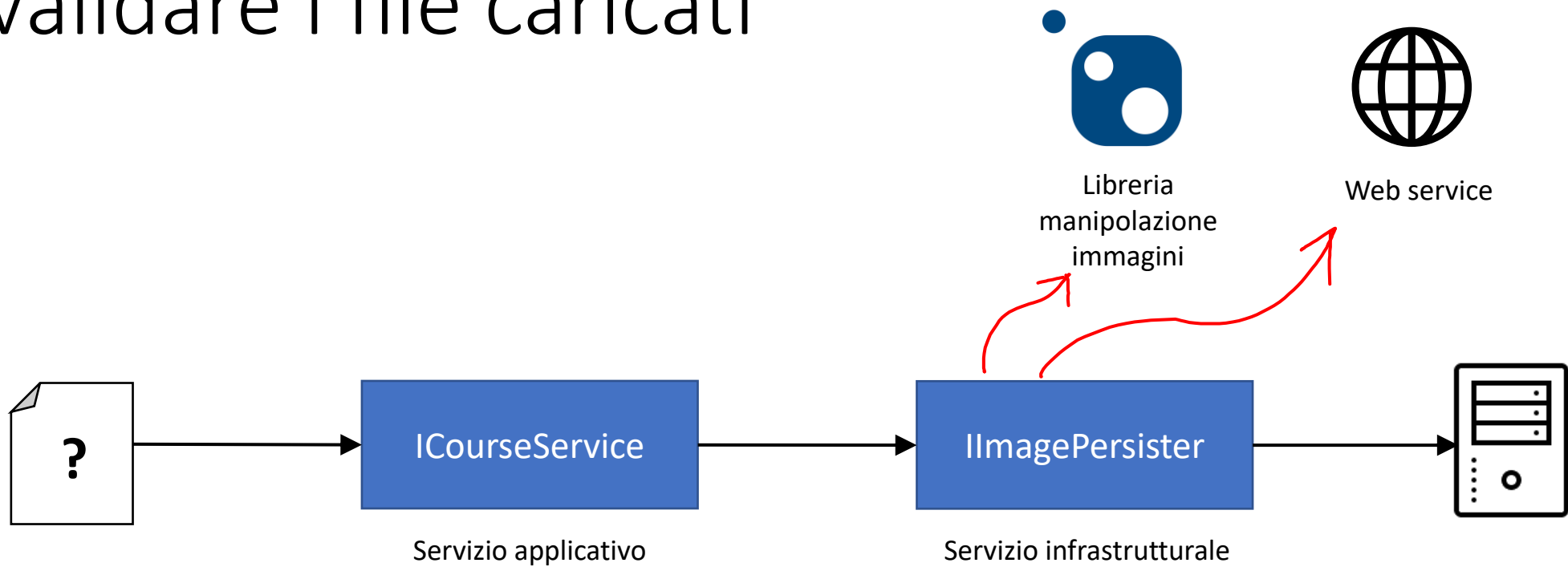
```

BROWSER

```

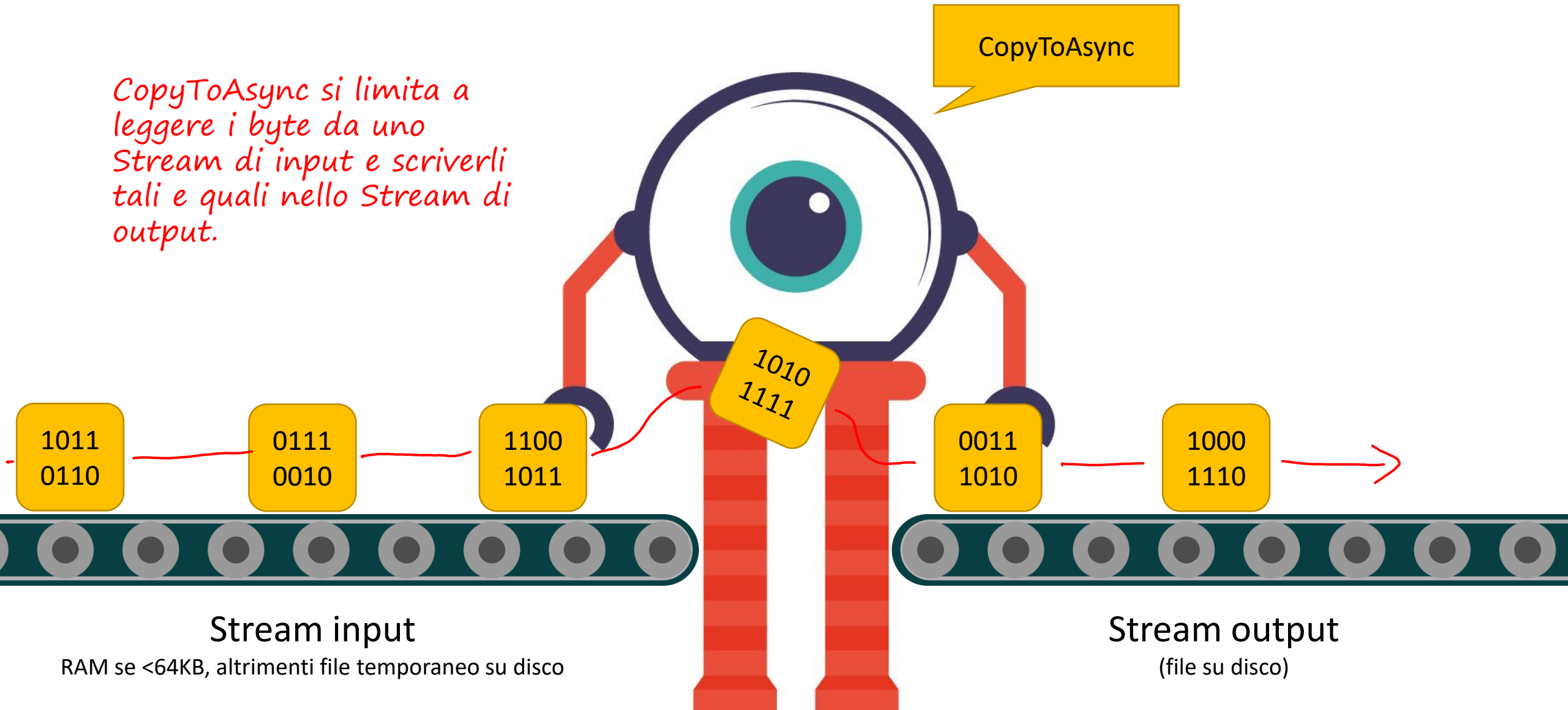
```

Validare i file caricati

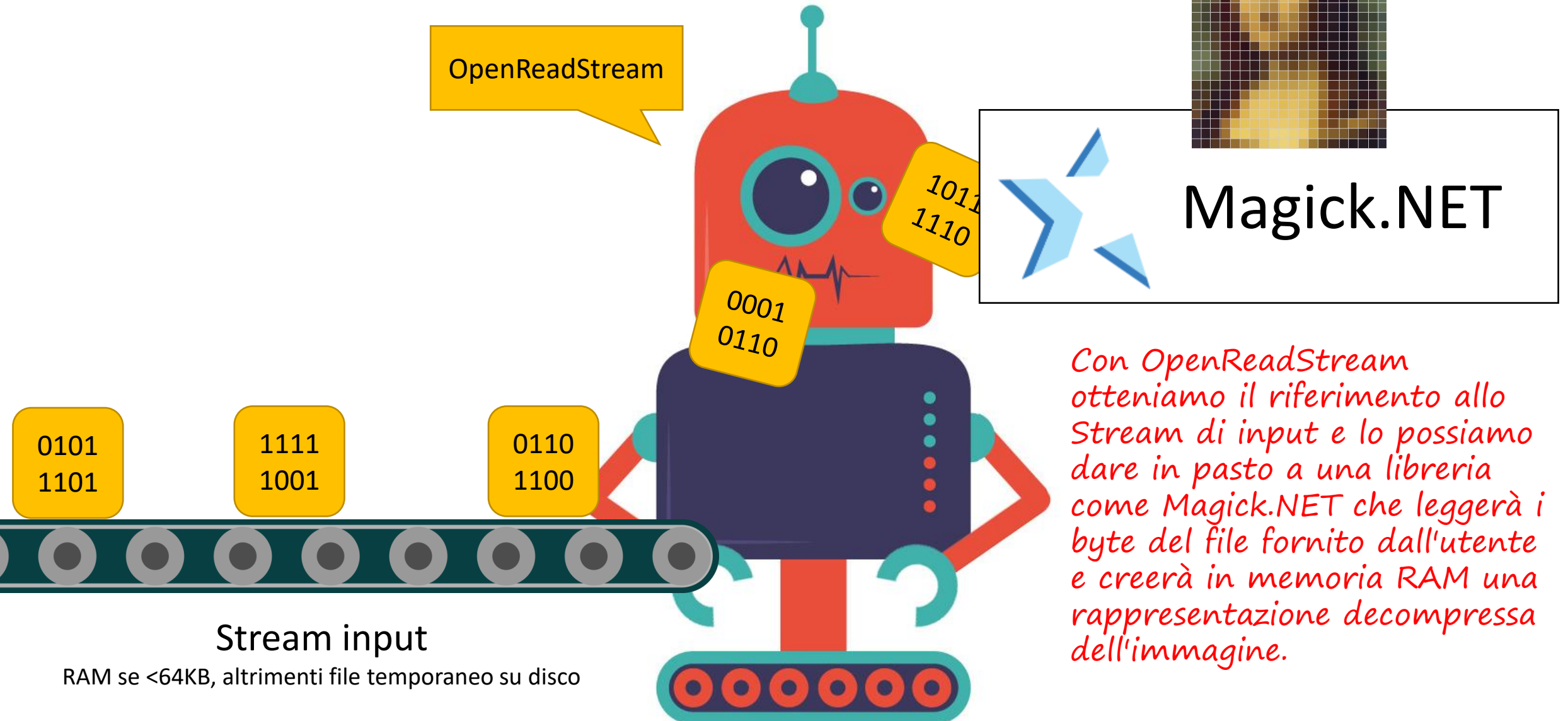


Stream

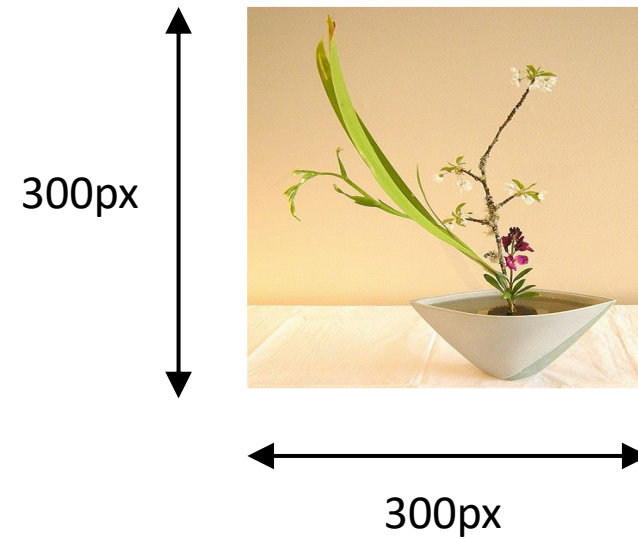
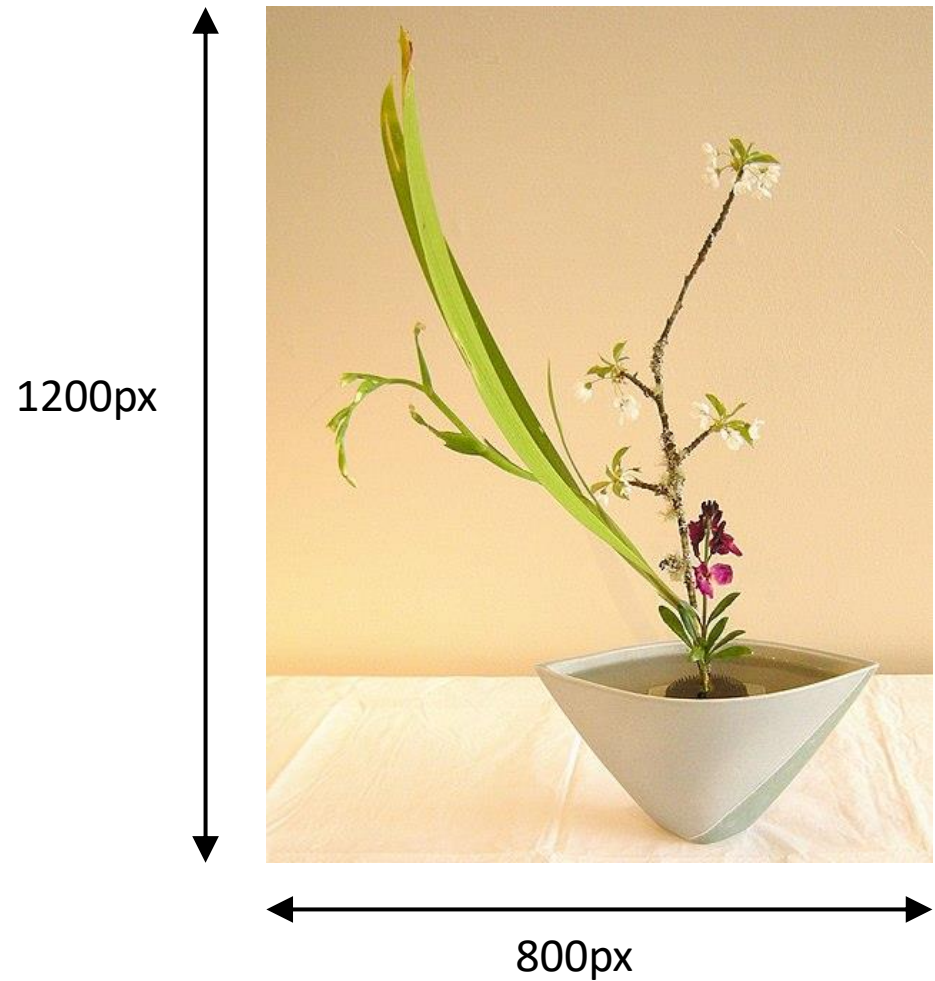
CopyToAsync si limita a leggere i byte da uno Stream di input e scriverli tali e quali nello Stream di output.



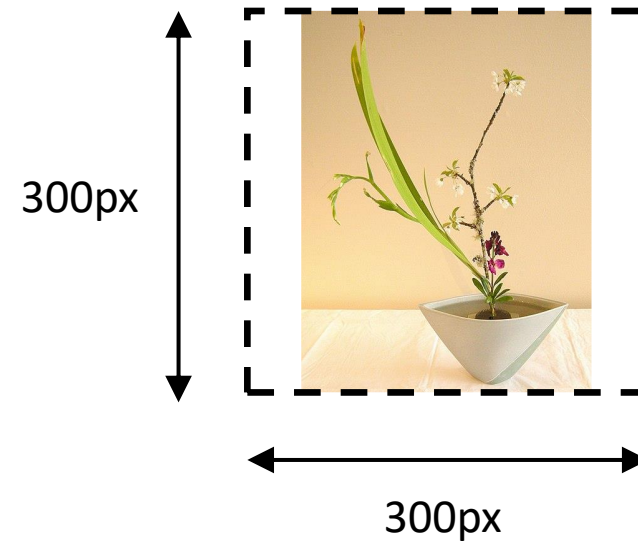
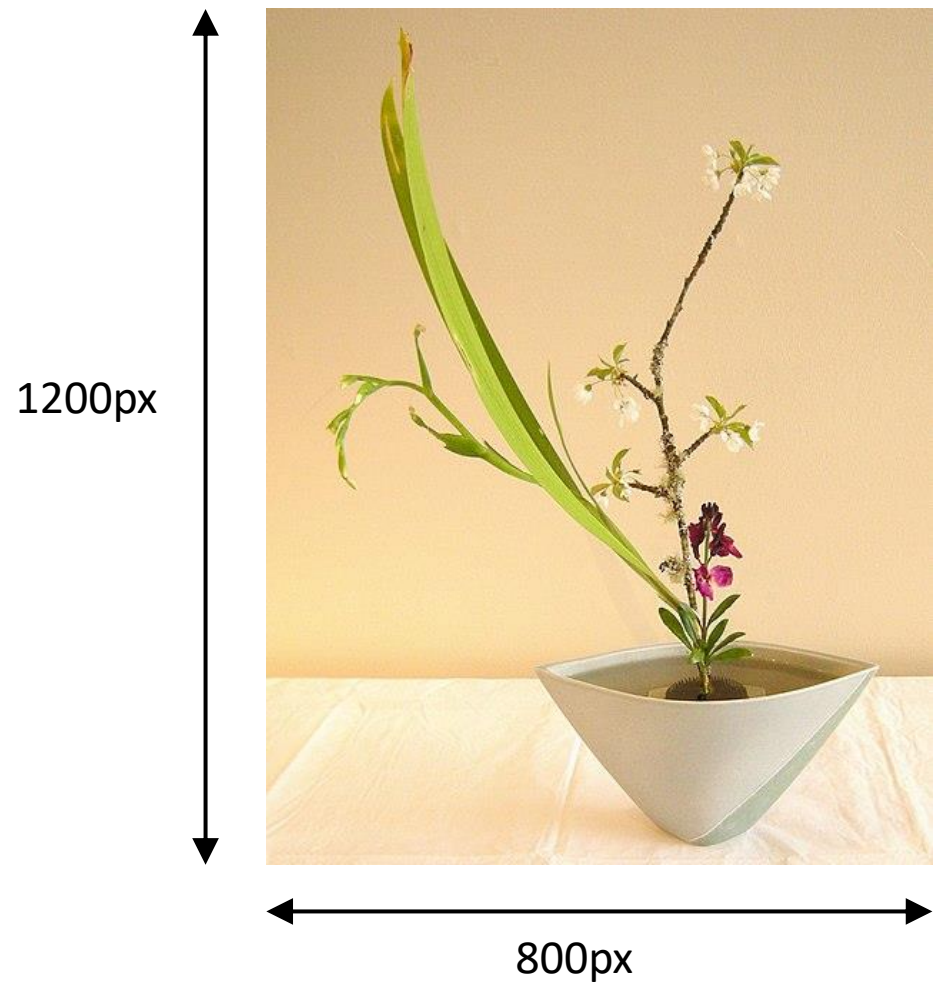
FileStream OpenReadStream



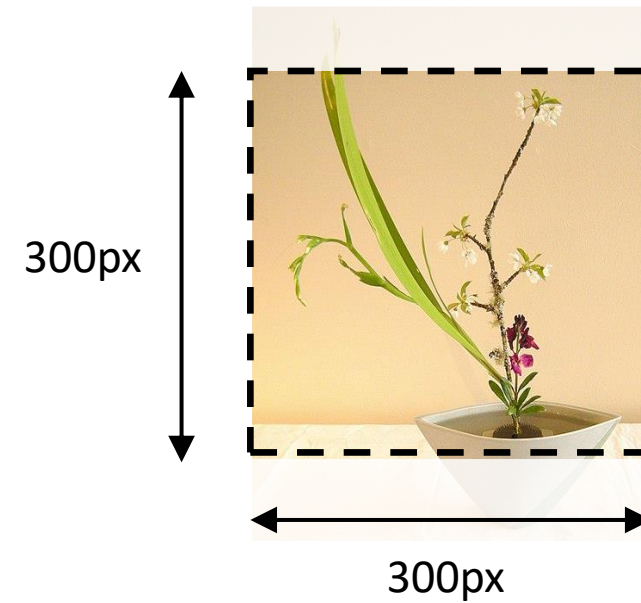
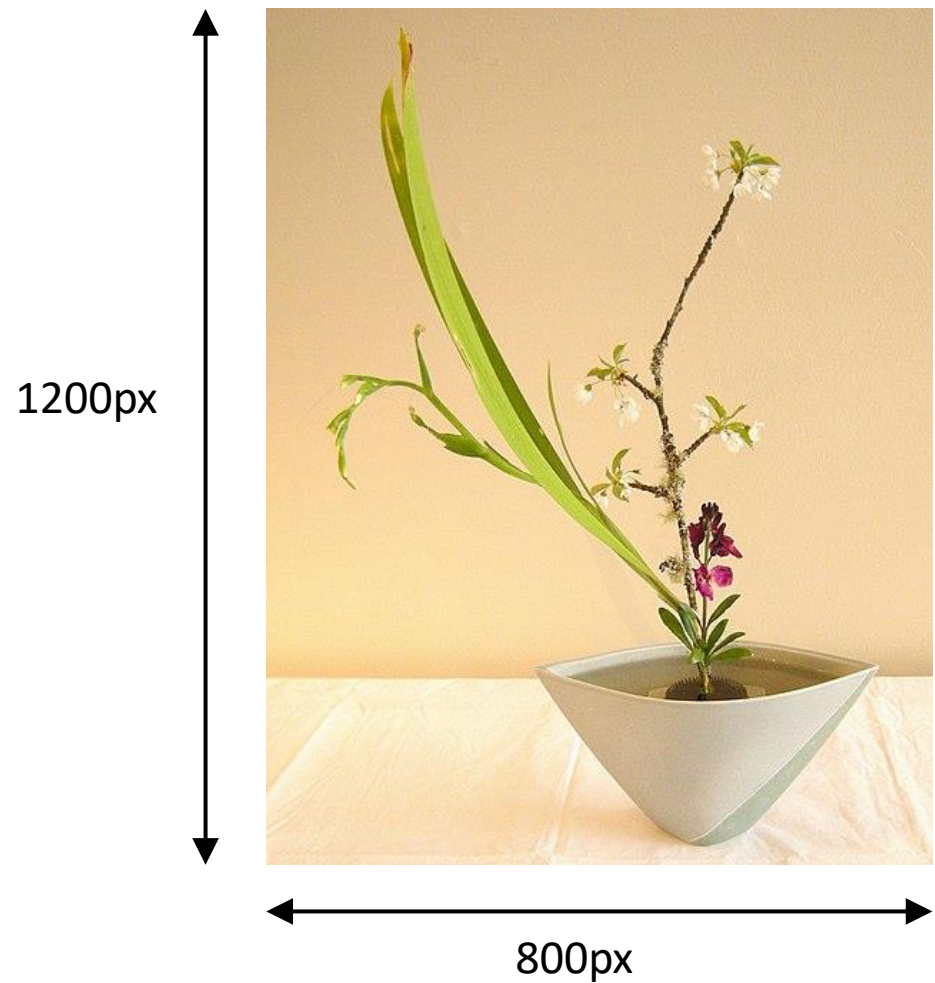
Ridimensionamento di un'immagine



Ridimensionamento di un'immagine



Ridimensionamento di un'immagine



Ridimensionamento con ImageMagick.NET

`Resize(int width, int height)`

`Resize(MagickGeometry geometry)`

RIDIMENSIONARE ALTERANDO LE PROPORZIONI DELL'IMMAGINE



```
using var stream = formFile.OpenReadStream();  
using var image = new MagickImage(stream);  
var resizeGeometry = new MagickGeometry(300, 300)  
{  
    IgnoreAspectRatio = true  
};  
image.Resize(resizeGeometry);
```

Ridimensionamento con ImageMagick.NET

Resize(`int` width, `int` height)
Resize(`MagickGeometry` geometry)

*Con un `MagickGeometry`
così configurato, equivale
a `Resize(width, height)`*

RIDIMENSIONARE MANTENENDO LE PROPORZIONI E L'INTERA FIGURA



```
using var stream = formFile.OpenReadStream();  
using var image = new MagickImage(stream);  
var resizeGeometry = new MagickGeometry(300, 300)  
{  
    FillArea = false  
};  
image.Resize(resizeGeometry);  
image.Extent(300, 300, MagickColor.FromRgb(255, 255, 255));
```

Ridimensionamento con ImageMagick.NET

`Resize(int width, int height)`

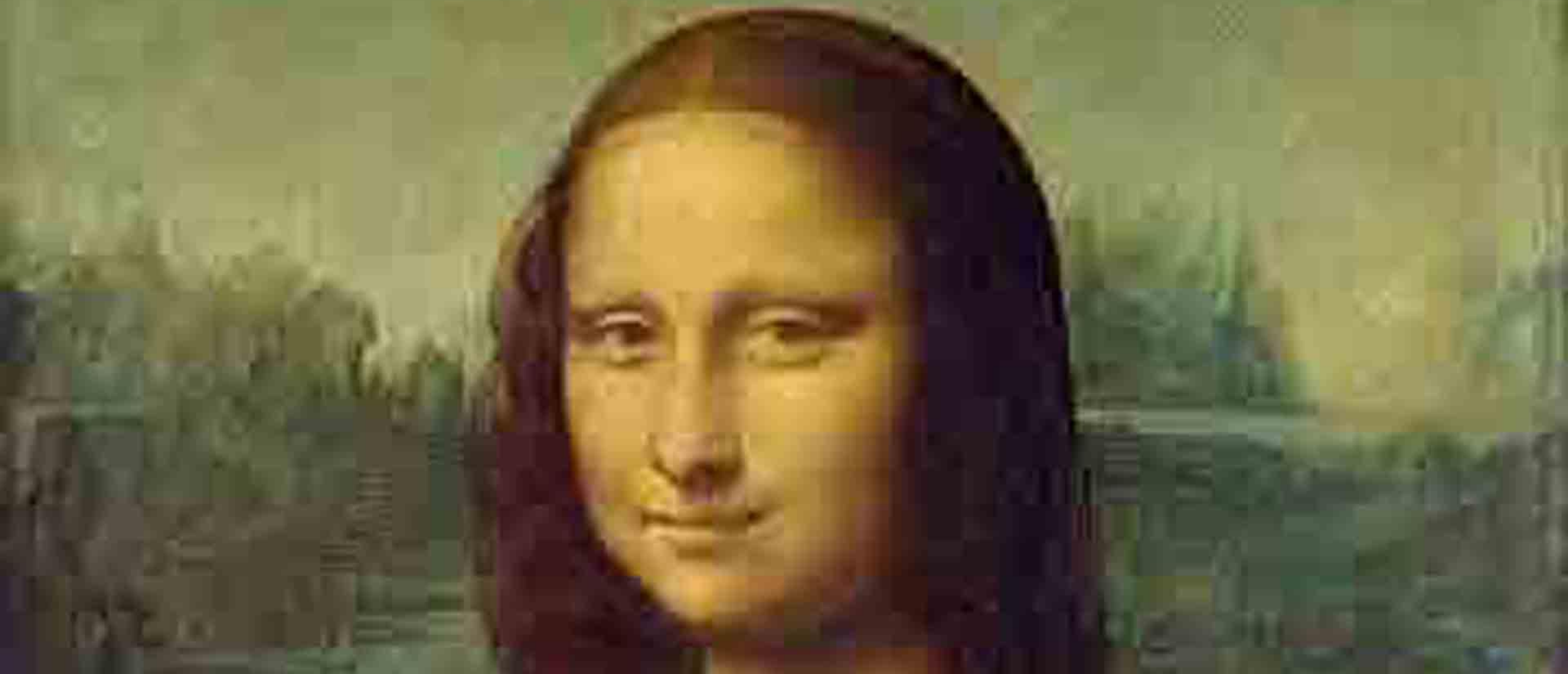
`Resize(MagickGeometry geometry)`

RIDIMENSIONARE MANTENENDO LE PROPORZIONI E TAGLIANDO L'ECESSO



```
using var stream = formFile.OpenReadStream();  
using var image = new MagickImage(stream);  
var resizeGeometry = new MagickGeometry(300, 300)  
{  
    FillArea = true  
};  
image.Resize(resizeGeometry);  
image.Crop(300, 300, Gravity.Northwest);
```

Qualità di un'immagine JPG (10% - 63KB)



Qualità di un'immagine JPG (50% - 130KB)



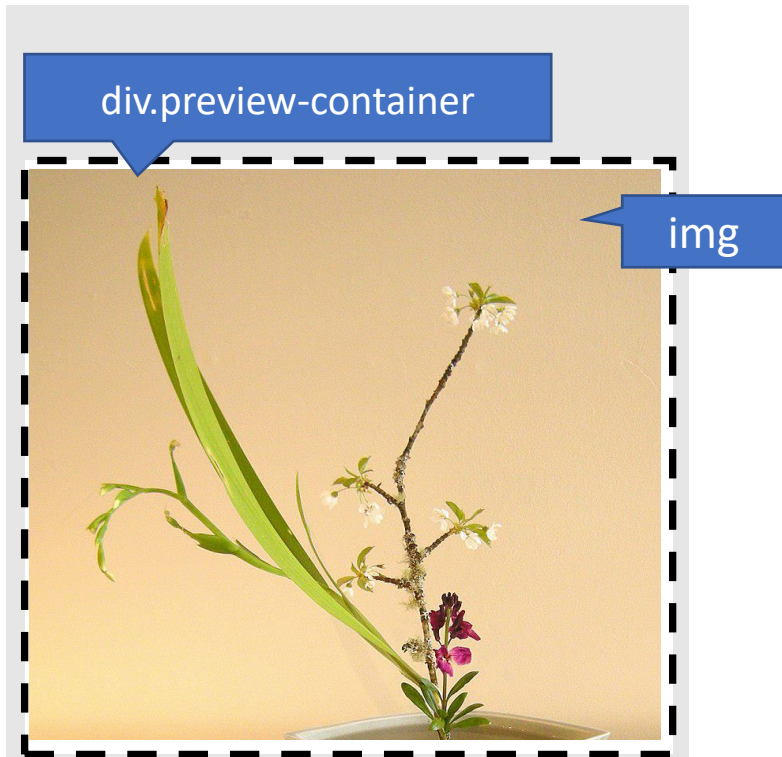
Qualità di un'immagine JPG (90% - 430KB)





CSS
IS
AWESOME

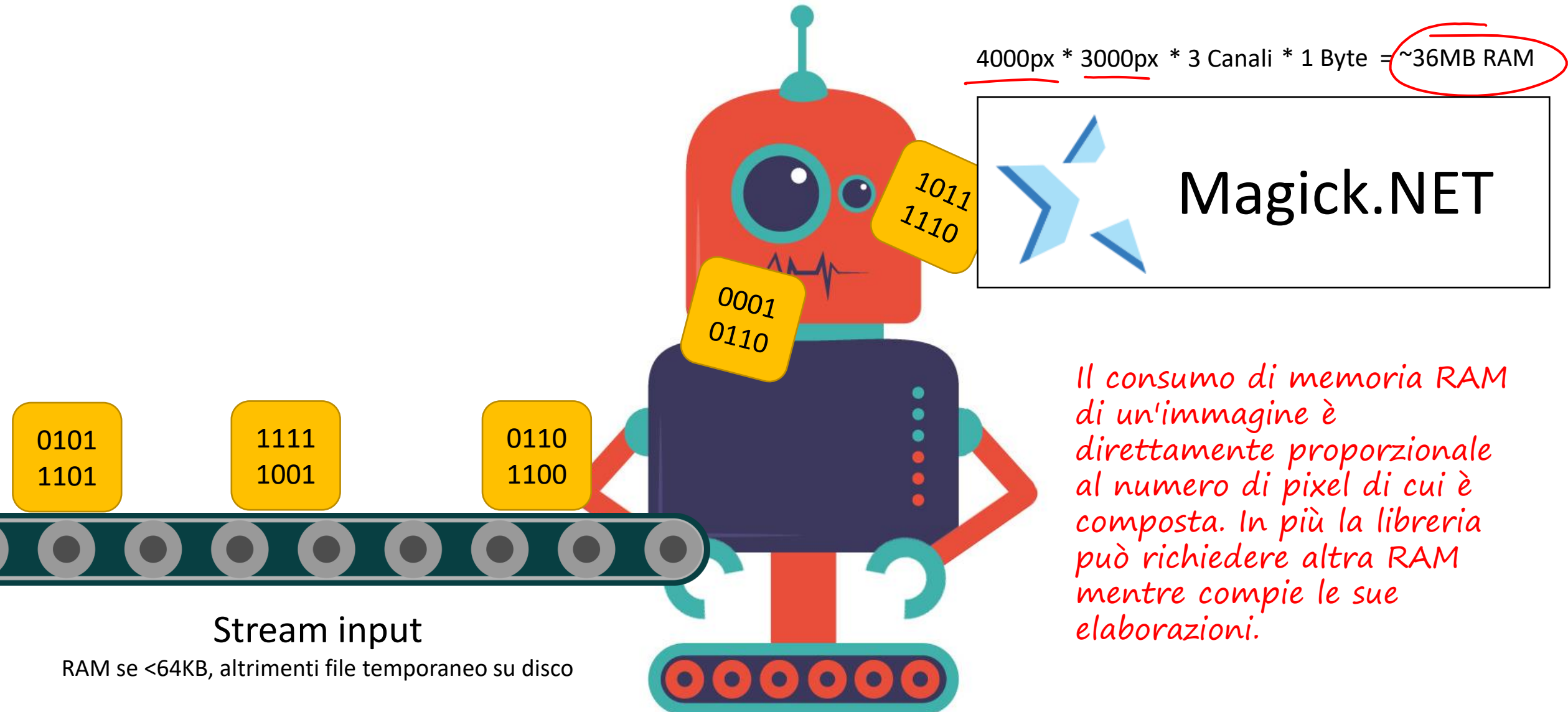
Anteprima del ridimensionamento



CSS

```
.preview-container {  
  padding-top: 100%;  
  position: relative;  
}  
  
.preview-container img {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
  object-position: top left;  
}
```

Decompressione delle immagini



36MB
NOT GREAT
NOT TERRIBLE



Limitare l'uso della RAM con Magick.NET

```
private readonly SemaphoreSlim semaphore;
```

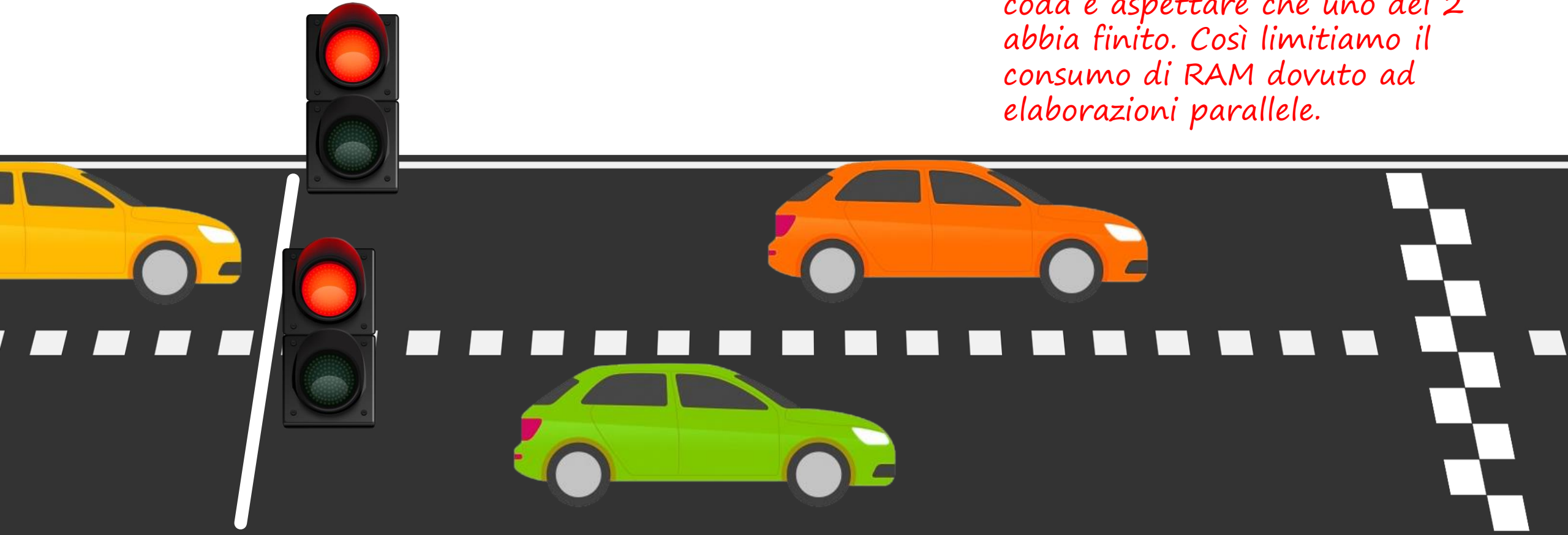
```
public MagickNetImagePersister() {  
    ResourceLimits.Width = 4000;  
    ResourceLimits.Height = 4000;  
    semaphore = new SemaphoreSlim(2);  
}
```

← Registrare come Singleton nella dependency injection, in modo che tutti i thread insistano su questa istanza di SemaphoreSlim

←

new SemaphoreSlim(2)

In questo esempio, al massimo 2 thread alla volta possono eseguire il codice "critico". Il terzo thread deve mettersi in coda e aspettare che uno dei 2 abbia finito. Così limitiamo il consumo di RAM dovuto ad elaborazioni parallele.



Limitare il peso dei file caricati

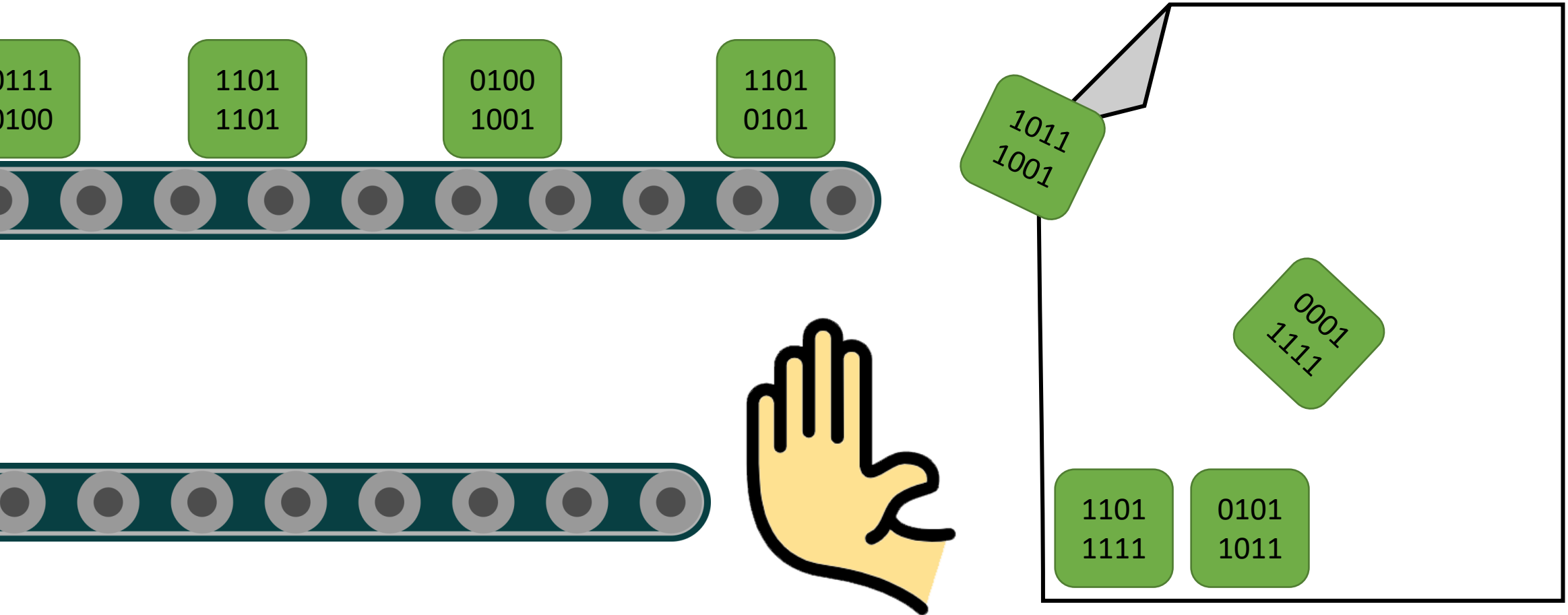
- Nel file `appsettings.json`

```
"Kestrel": {  
  "Limits": {  
    "MaxRequestBodySize": 5242880  
  }  
}
```

- Nel metodo `ConfigureServices` della classe `Startup`

```
services.Configure<KestrelServerOptions>(Configuration.GetSection("Kestrel"));
```

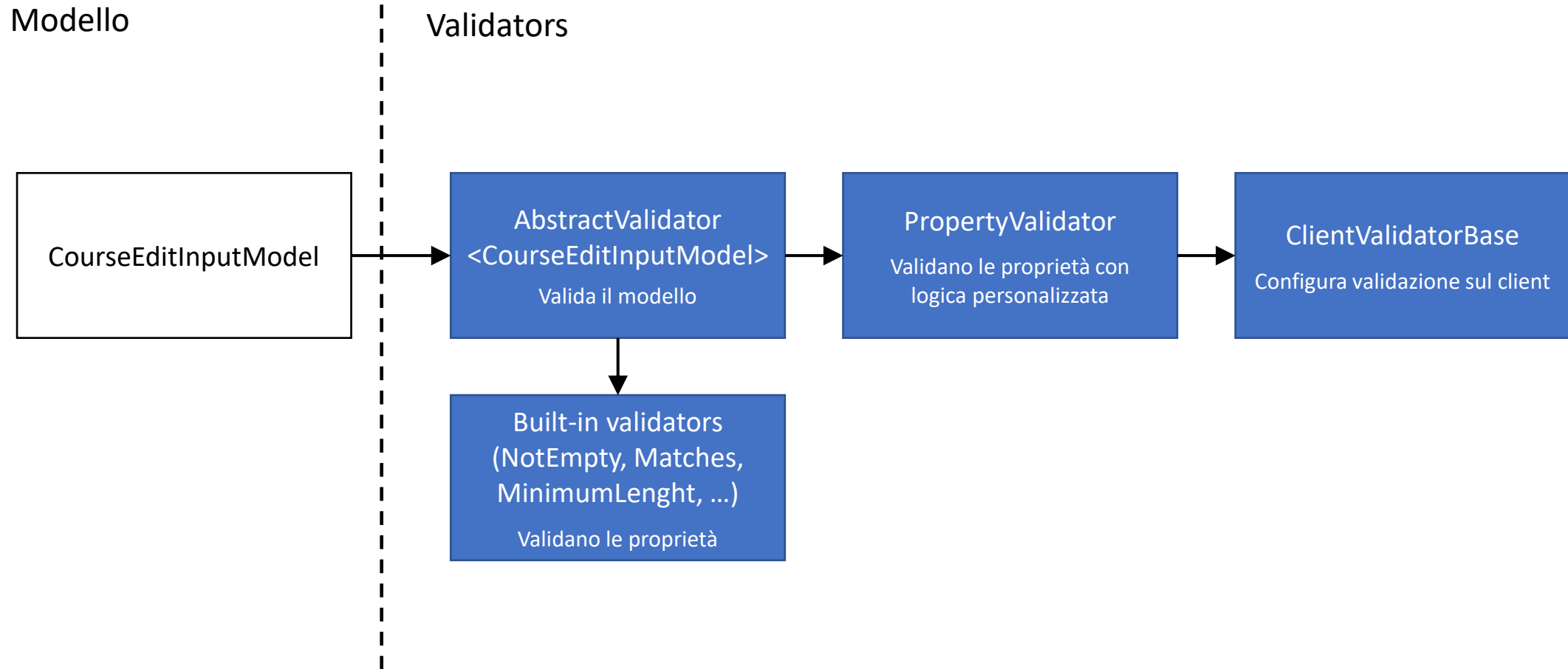
Il SO nega la scrittura contemporanea su un file



Da Data Annotation a FluentValidation

| Data Annotation | Built-in validator di FluentValidation |
|---|---|
| Required | NotEmpty(), NotNull() |
| Range(1900, 2019) | InclusiveBetween(1900, 2019), ExclusiveBetween(1899, 2020) |
| MinLength(10), MaxLength(100) | MinimumLength(10), MaximumLength(100) |
| RegularExpression("^[a-z]+\$") | Matches("^[a-z]+\$") |
| Compare(nameof(<i>AltraProprietà</i>)) | Equal(model => model.<i>AltraProprietà</i>) |
| CreditCard, EmailAddress, Phone, Url | CreditCard(), EmailAddress(), --, -- |
| Remote("Action", "Controller") | -- |

Panoramica dei validator di FluentValidation



Progresso nella specifica

- Punto 3: prezzo corrente \leq prezzo intero;
- Punto 11: link al form di creazione;
- Punto 12: form di creazione;
- Punto 13: link al form di modifica;
- Punto 14: form di modifica;
- Punto 15: editor WYSIWYG;
- Punto 16: protezione da attacchi XSS.

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 |
|---|---|---|---|