





Sezione 06

ASP.NET Core MVC: I controller

Catalogo corsi

[+ Crea nuovo](#)

Titolo	Valutazione ▼	Prezzo	
 Web marketing facile <i>di Mario Rossi</i>	★★★★★	EUR 17.99 EUR 19.99	Dettagli
 L'ABC del fai da te <i>di Mario Rossi</i>	★★★★★	EUR 7.99	Dettagli
 Come fare gli origami <i>di Mario Rossi</i>	★★★★☆	EUR 30.99 EUR 34.99	Dettagli
 Corso di ceramica <i>di Mario Rossi</i>	★★★★☆	USD 10.99 USD 11.99	Dettagli

Punto n° 2

Gli utenti devono poter visualizzare l'intero catalogo dei corsi da una pagina di elenco.

Come fare gli origami

Iscriviti

Lorem ipsum dolor sit amet consectetur adipiscing 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

Mario Rossi



EUR 30.99

~~EUR 34.99~~



Punto n° 5

Cliccando uno dei risultati nell'elenco, si accede alla pagina di dettaglio del corso.

Realizzare la UI di un'applicazione web

- Due tecnologie ugualmente idonee.

ASP.NET Core
MVC

ASP.NET Core
Razor Pages

Routing, dependency injection, Razor, ...

ASP.NET Core MVC

- Promuove la "separazione delle responsabilità";
- Model, View e Controller: 3 componenti per 3 differenti responsabilità.



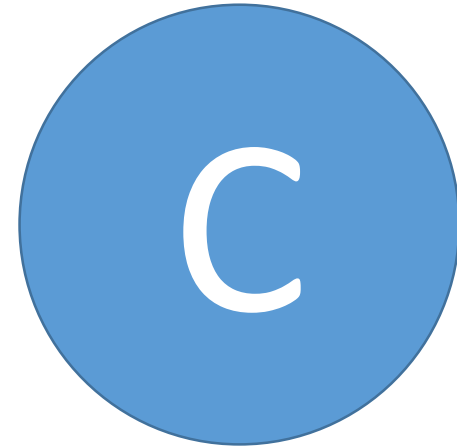
Model

Offre servizi applicativi
Contiene la logica di business



View

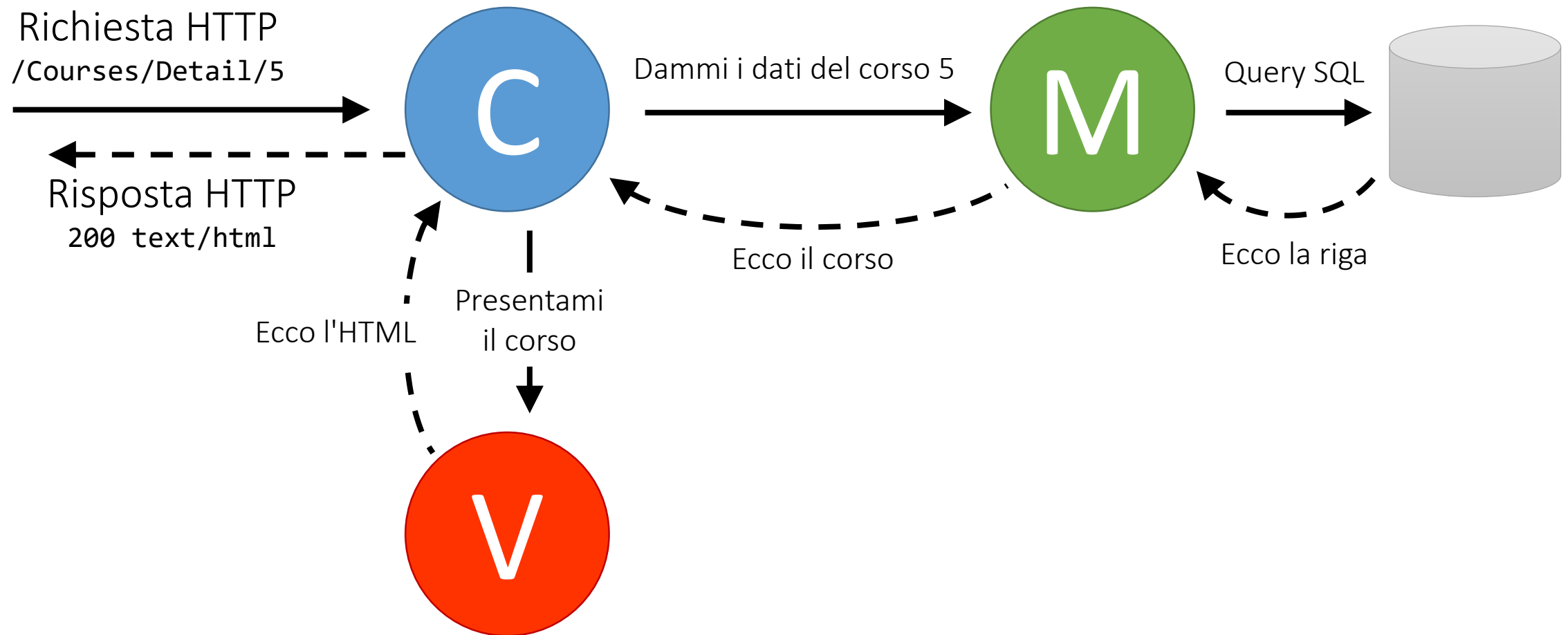
Presenta i dati in HTML
Sfrutta l'engine Razor



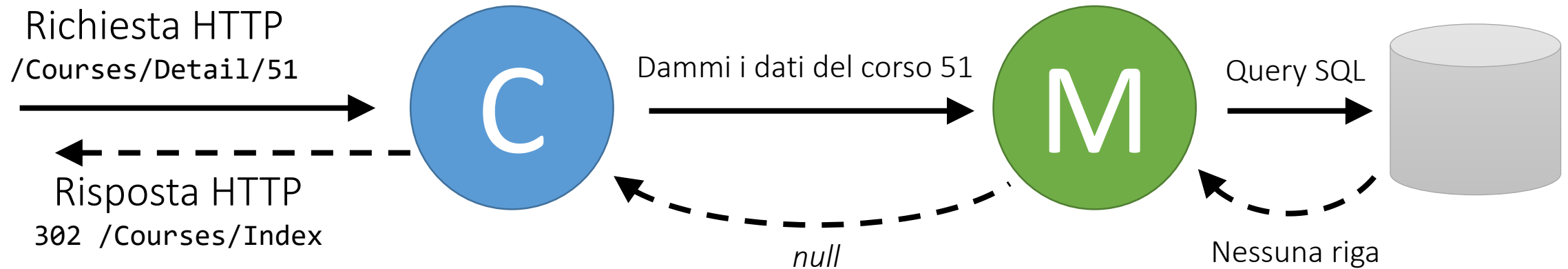
Controller

Riceve l'input della richiesta
Coordina la creazione della risposta

Il pattern MVC



Il pattern MVC: a volte la view non è necessaria

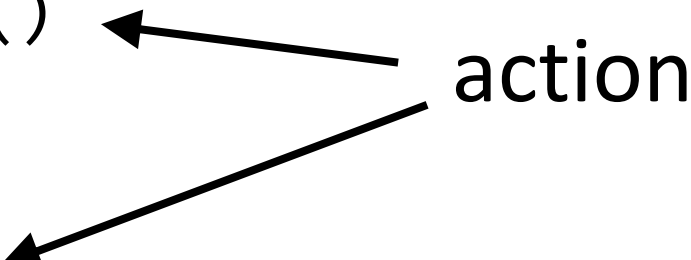


Il pattern MVC: le responsabilità

- La responsabilità del **Model** è quella di **offrire dei servizi applicativi**;
 - Contiene la logica di business che regola l'accesso ai dati;
- La responsabilità della **View** è quella di **presentare dati in HTML**;
 - È "stupida": deve solo presentare quello che le viene passato dal controller;
- La responsabilità del **Controller** è quella di **coordinare**;
 - Deve restare snello: quindi non accede ai database, non invia email, ecc.
 - Ottiene dati dal model e li passa alla view per farglieli presentare in HTML.

Il controller

```
public class CoursesController : Controller
{
    public IActionResult Index()
    {
        return View(oggetto);
    }
    public IActionResult Detail(string id)
    {
        return View(oggetto);
    }
}
```



The word "action" is positioned to the right of the code. Two black arrows originate from it: one points to the `Index()` method signature, and the other points to the `Detail(string id)` method signature, indicating that these methods represent actions in the MVC pattern.

Tipi di risultato che un'action può restituire

Istruzione	Quando usarlo
<code>return View(<i>oggetto</i>);</code>	Per restituire del contenuto HTML generato dinamicamente
<code>return Content(<i>testo</i>);</code>	Per avere l'esatto controllo sul contenuto della risposta
<code>return File(<i>percorsoFile</i>);</code>	Per restituire il contenuto binario di un file presente su disco
<code>return Redirect(<i>url</i>);</code>	Per reindirizzare ad un altro URL
<code>return Json(<i>oggetto</i>);</code>	Per restituire un oggetto serializzato in JSON

ASP.NET Core MVC: suddivisione in directory

Controllers

 CoursesController.cs

Models

 CourseService.cs

 CourseDetailViewModel.cs

← In realtà la classi del Model potete organizzarle come volete, anche su più directory o in progetti separati.

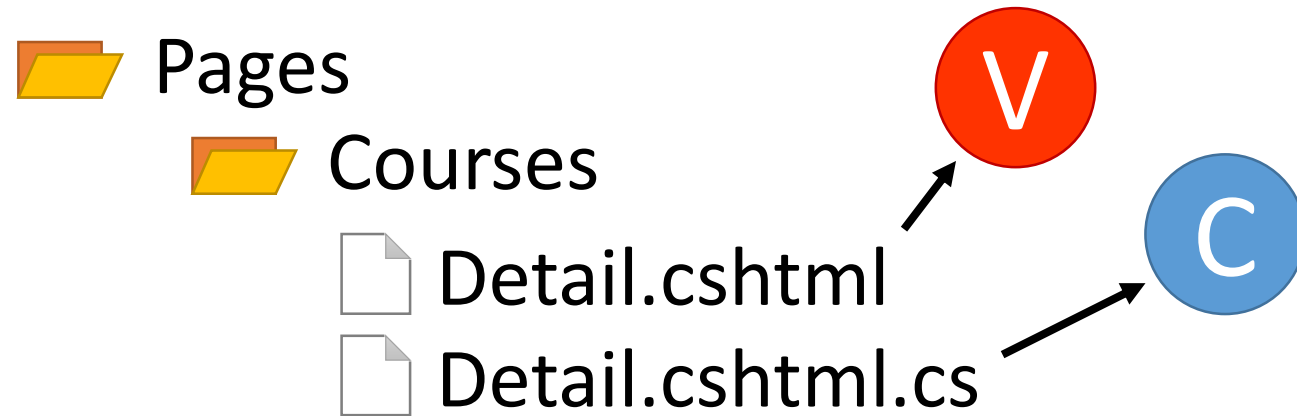
Views

Courses

 Detail.cshtml

Suddivisione dei file
per responsabilità

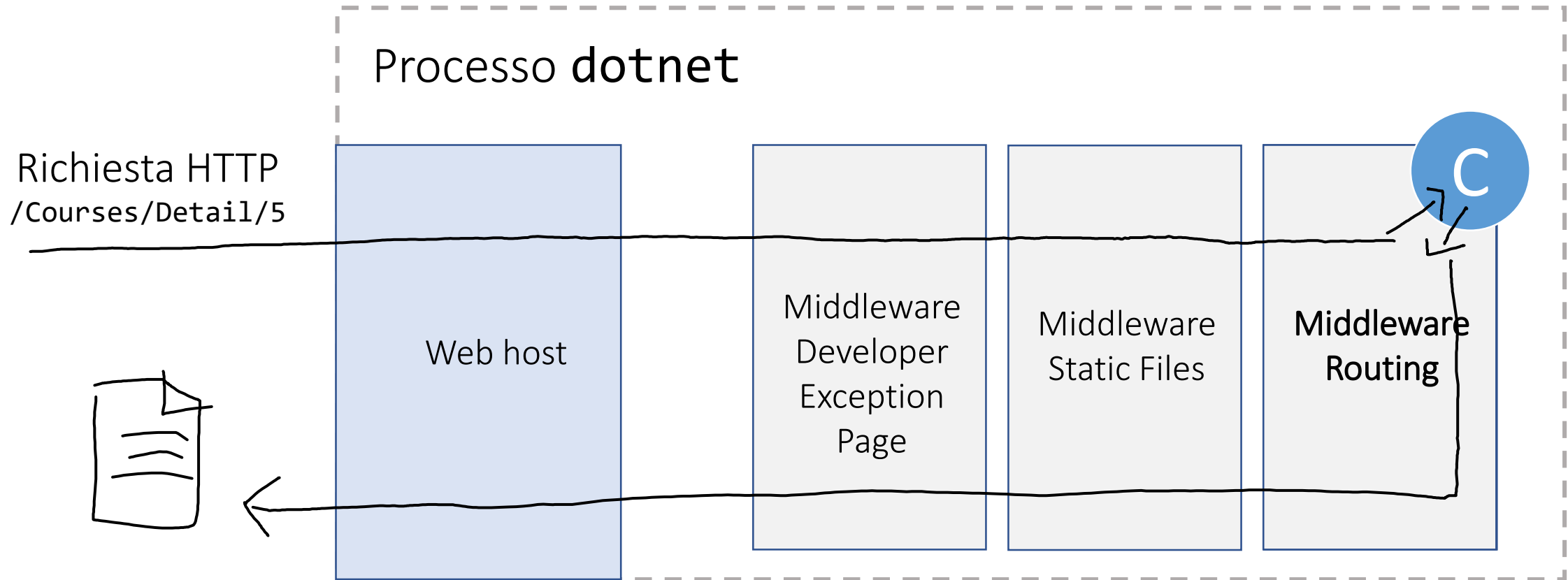
Mentre in ASP.NET Core Razor Pages...



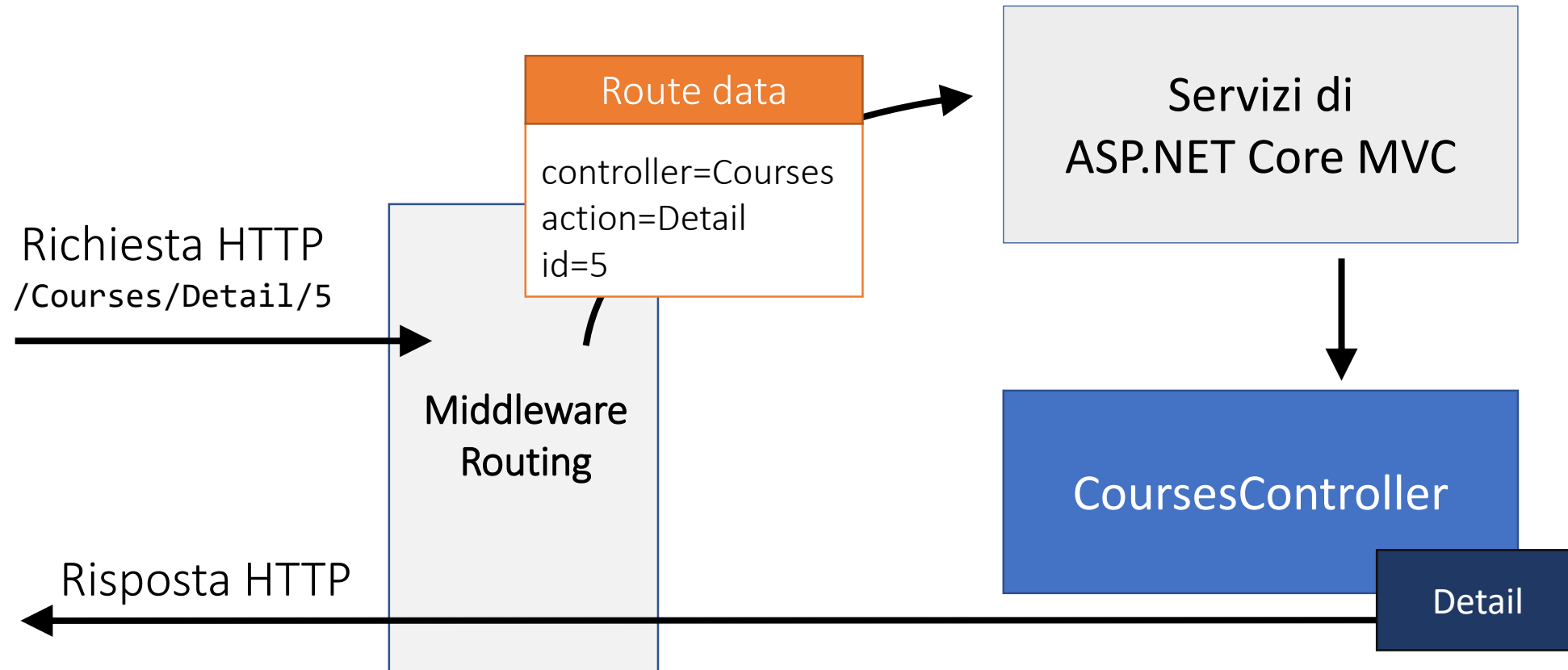
Models

Suddivisione dei file
per obiettivo

La nostra applicazione



Il middleware di routing



Impostare la Compatibility Version

- Serve a scegliere quale comportamento desideriamo usare per i servizi di ASP.NET Core MVC. Le versioni più nuove sono spesso più performanti e ottimizzate, ma potrebbero anche avere degli effetti collaterali che portano l'applicazione a comportarsi diversamente rispetto al passato. Per i dettagli, consultiamo il [blog di ASP.NET](#).

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc()
        .SetCompatibilityVersion(CompatibilityVersion.Version_2_2);
}
```