```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
```

1. Δημιουργία Builder

```
var builder = WebApplication.CreateBuilder(args);
```

Τι κάνει

Δημιουργεί τον **builder της εφαρμογής**.

Ο builder:

- φορτώνει ρυθμίσεις
- ετοιμάζει services
- διαβάζει το `appsettings.json`
- παίρνει arguments από command line

Με απλά λόγια

Εδώ **ξεκινά η κατασκευή της web εφαρμογής**.

---

2. Προσθήκη υπηρεσιών MVC

```
builder.Services.AddControllersWithViews();
```

Τι κάνει

Προσθέτει στο σύστημα **MVC υπηρεσίες**.

Αυτό σημαίνει ότι ενεργοποιούνται:

- Controllers

- Views
- Routing για MVC
- Model binding
- Validation

Αν δεν υπάρχει
Το project **δεν μπορεί να χρησιμοποιήσει controllers και views**.

---

3. Δημιουργία της εφαρμογής
```
var app = builder.Build();
```
Τι κάνει
Εδώ ολοκληρώνεται το build και δημιουργείται το πραγματικό **WebApplication object**.
Δηλαδή:
```
builder → προετοιμασία
app → πραγματική εφαρμογή
```

---

4. Έλεγχος περιβάλλοντος
```
if (!app.Environment.IsDevelopment())
```
Τι κάνει
Ελέγχει αν το πρόγραμμα **ΔΕΝ τρέχει σε development mode**.
Τα ASP.NET projects έχουν περιβάλλοντα:
```
Development
Staging
Production
```

---

5. Διαχείριση σφαλμάτων
```
app.UseExceptionHandler("/Home/Error");
```
Τι κάνει
Αν γίνει error στο production:
ο χρήστης μεταφέρεται στη σελίδα
```
/Home/Error
```
Δηλαδή:
```
HomeController → Error action
```
Αυτό προστατεύει από το να εμφανιστούν τεχνικά errors στον χρήστη.

---

6. HSTS (ασφάλεια HTTPS)
```
app.UseHsts();
```
Τι είναι HSTS
HTTP Strict Transport Security.
Λέει στον browser:
```
Αυτό το site πρέπει να ανοίγει μόνο με HTTPS
```
Default
30 ημέρες.

---

7. Redirect HTTP → HTTPS
```
app.UseHttpsRedirection();
```
Τι κάνει
Αν κάποιος μπει:
```
http://site.com
```
τον μεταφέρει σε:
```
https://site.com
```

---

8. Στατικά αρχεία

```
app.UseStaticFiles();
```

Ενεργοποιεί τα αρχεία από τον φάκελο:

```
wwwroot
```

π.χ.

```
css
javascript
images
fonts
```

Αν το αφαιρέσεις:
το site δεν φορτώνει CSS και JS.

---

9. Routing

```
app.UseRouting();
```

Ενεργοποιεί το **σύστημα routing**.
Το routing αποφασίζει:

```
ποιο URL → ποιο controller
```

Παράδειγμα:

```
/home/index
```

---

10. Authorization

```
app.UseAuthorization();
```

Ενεργοποιεί τον έλεγχο **δικαιωμάτων πρόσβασης**.
Χρησιμοποιείται όταν υπάρχουν:

```
users
roles
login system
```

---

11. Default Route MVC

```
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
```

Αυτό είναι **η βασική διαδρομή του MVC**.
Pattern

```
{controller=Home}/{action=Index}/{id?}
```

Σημαίνει:

Τμήμα	Σημασία
controller	ποιος controller
action	ποια μέθοδος
id	προαιρετικό parameter

Παραδείγματα
URL:

```
/
```

πάει στο:

```
HomeController → Index()
```

URL:

```
/Home/About
```
πάει στο:
```
HomeController → About()
```

URL:
```
/Products/Details/5
```
πάει στο:
```
ProductsController → Details(5)
```

---

12. Εκκίνηση εφαρμογής
```
app.Run();
```
Αυτό ξεκινά τον web server.
Η εφαρμογή:

- ακούει HTTP requests
- απαντά σε browser

Χωρίς αυτό **το site δεν τρέχει**.

Συνολική λειτουργία του Program.cs
Η ροή είναι:
```
1. Δημιουργία builder
2. Προσθήκη MVC services
3. Build εφαρμογής
4. Ρύθμιση middleware
5. Routing
6. Εκκίνηση server
```

Δομή ενός MVC project
Με αυτόν τον κώδικα δημιουργείται συνήθως project:
```
CoolWeb
│
├── Controllers
│   └── HomeController.cs
│
├── Models
│
├── Views
│   └── Home
│       ├── Index.cshtml
│       └── Privacy.cshtml
│
├── wwwroot
│   ├── css
│   ├── js
│   └── images
│
└── Program.cs
```

Πώς δουλεύει η ροή MVC
```
Browser request
      ↓
Routing
```

```
      ↓
Controller
      ↓
Model (δεδομένα)
      ↓
View
      ↓
HTML στον browser
```

Παράδειγμα
URL:
`https://localhost/Home/Index`
Flow:
HomeController
↓
Index()
↓
Views/Home/Index.cshtml
↓
HTML page