

ASP Development Associate

Prüfungscode IFW-ASP-200-01 · Fassung 1.0 · Stand 18.09.2026

Webanwendungen mit ASP.NET Core: Projektaufbau und Konfiguration, Dependency Injection, Routing, Middleware und Datenzugriff mit EF Core.

Die Prüfung im Überblick

Bearbeitungszeit 1 Std. 30 Min.	Fragen 70
Bestehensgrenze 70 %	Fragetypen Einfach- und Mehrfachauswahl
Prüfungssprache Deutsch	Zertifikat gültig Unbegrenzt
Durchführung Online, zeitlich überwacht	Preis 49,00 €

Themenblöcke

Block	Thema	Gewichtung	Fragen
1	Projektaufbau, Konfiguration und asynchrone Programmierung	20 %	14
2	Dependency Injection und Lebensdauern	13 %	9
3	Routing, Controller und Model Binding	27 %	19
4	Middleware und Sicherheit	27 %	19
5	Datenzugriff mit EF Core	13 %	9
Gesamt		100 %	70

Die Gewichtung gibt an, welchen Anteil ein Block an den Prüfungsfragen hat. Die Fragen werden beim Start zufällig und anteilig je Block aus dem Fragenpool gezogen.

Projektaufbau, Konfiguration und asynchrone Programmierung

Der Einstiegspunkt der Anwendung, ihre Einstellungen und der Umgang mit asynchronem Code.

- 1.1** Anwendung mit WebApplication aufbauen
CreateBuilder, Dienste registrieren, Pipeline zusammenstellen
- 1.2** Konfiguration lesen
appsettings.json, Umgebungsvariablen und ihre Rangfolge, GetConnectionString()
- 1.3** Options Pattern anwenden
Configure<T> und der Zugriff über IOptions<T>
- 1.4** Asynchron programmieren
async und await, Task<T>, Freigabe des Threads beim Warten

Dependency Injection und Lebensdauern

Dienste registrieren und die richtige Lebensdauer wählen.

- 2.1** Dienste registrieren
AddSingleton, AddScoped, AddTransient
- 2.2** Lebensdauern unterscheiden
ein Scope je HTTP-Request, neue Instanz je Anforderung
- 2.3** Abhängigkeiten auflösen lassen
Konstruktordinjektion und die Captive Dependency als Fehlerquelle

Routing, Controller und Model Binding

Von der URL zur Action: Routen, gebundene Werte, Validierung und Antworten.

- 3.1** Routen definieren
Attribute Routing, Conventional Routing und Minimal API
- 3.2** Routenparameter einschränken
Constraints wie :int als Teil des Matchings
- 3.3** Model Binding nutzen
Werte aus Route, [FromQuery] und [FromBody]
- 3.4** Eingaben validieren
Data Annotations, ModelState, erneutes Rendern einer Razor Page
- 3.5** Antworten erzeugen
ActionResult<T>, Ok(), NotFound() und ihre Statuscodes

- 3.6** Views und Tag Helper verwenden
Razor-Ausdrücke mit @, asp-controller, URL-Erzeugung

BLOCK 4

27 % · 19 Fragen

Middleware und Sicherheit

Die Pipeline und die Middleware, die über Zugriff und Fehlerverhalten entscheidet.

- 4.1** Pipeline verstehen
Reihenfolge der Middleware, Arbeit vor und nach dem Aufruf von next
- 4.2** Eigene Middleware schreiben
app.Use und das Kurzschließen der Kette
- 4.3** Authentication und Authorization einordnen
richtige Reihenfolge, [Authorize], [AllowAnonymous]
- 4.4** Umgebungsabhängige Middleware einsetzen
UseExceptionHandler, UseHsts, UseStaticFiles mit wwwroot

BLOCK 5

13 % · 9 Fragen

Datenzugriff mit EF Core

Daten über den DbContext lesen, ändern und das Schema pflegen.

- 5.1** DbContext und DbSet einsetzen
Zuordnung zur Tabelle und Einstieg in LINQ-Abfragen
- 5.2** Verzögerte Ausführung beachten
SQL entsteht erst beim Materialisieren der Abfrage
- 5.3** Änderungen speichern
Change Tracking und SaveChangesAsync()
- 5.4** Migrations anwenden
dotnet ef migrations add und database update

Institut für Webentwicklung · unabhängiges Lernangebot ohne Verbindung zu den genannten Technologieanbietern. Maßgeblich ist die Fassung auf der Zertifizierungsseite.