**Datei-Upload in ASP.NET Core**

ist sehr flexibel und erfolgt typischerweise über ein Formular mit enctype="multipart/form-data". ASP.NET Core stellt dafür das Interface IFormFile bereit, mit dem hochgeladene Dateien serverseitig verarbeitet werden können.

**Voraussetzungen**

Form mit enctype="multipart/form-data"

Controller oder Page mit IFormFile (bzw. List<IFormFile> für mehrere Dateien)

**Beispiel: Einzelner Datei-Upload**

**Formular**

<form method="post" enctype="multipart/form-data" asp-controller="Upload" asp-action="Hochladen">

<input type="file" name="datei" />

<button type="submit">Hochladen</button>

</form>

**Controller**

public class UploadController : Controller

{

[HttpPost]

public async Task<IActionResult> Hochladen(IFormFile datei)

{

if (datei != null && datei.Length > 0)

{

var pfad = Path.Combine("wwwroot/uploads", datei.FileName);

// Sicherstellen, dass der Ordner existiert

Directory.CreateDirectory(Path.GetDirectoryName(pfad)!);

using var stream = new FileStream(pfad, FileMode.Create);

await datei.CopyToAsync(stream);

return Content("Datei erfolgreich hochgeladen!");

}

return BadRequest("Keine Datei ausgewählt.");

}

}

**Mehrere Dateien hochladen**

**Formular**

<form method="post" enctype="multipart/form-data" asp-controller="Upload" asp-action="Mehrere">

<input type="file" name="dateien" multiple />

<button type="submit">Hochladen</button>

</form>

**Controller**

[HttpPost]

public async Task<IActionResult> Mehrere(List<IFormFile> dateien)

{

foreach (var datei in dateien)

{

if (datei.Length > 0)

{

var pfad = Path.Combine("wwwroot/uploads", datei.FileName);

Directory.CreateDirectory(Path.GetDirectoryName(pfad)!);

using var stream = new FileStream(pfad, FileMode.Create);

await datei.CopyToAsync(stream);

}

}

return Content("Dateien erfolgreich hochgeladen!");

}

**Sicherheitshinweise**

* **Dateiname prüfen**: Keine Pfadangaben oder Sonderzeichen übernehmen!
* **Dateitypen validieren** (z. B. .pdf, .jpg)
* **Maximale Dateigröße festlegen** in appsettings.json oder Program.cs:

builder.Services.Configure<FormOptions>(options =>

{

options.MultipartBodyLengthLimit = 10 \* 1024 \* 1024; // 10 MB

});

**Upload-Ziel: wwwroot vs. Speicherort**

* Dateien in wwwroot sind **öffentlich abrufbar**
* Alternativ: Dateien in App\_Data oder externen Ordner speichern

**Übung**

Erstellen Sie ein Formular, mit dem Benutzer eine Datei (nur .txt oder .pdf) hochladen dürfen. Speichern Sie die Datei im Ordner wwwroot/uploads und zeigen Sie den Dateinamen auf einer Bestätigungsseite an.