

ASP.NET Core MVC

Upload & Download von Dateien

Ingo Köster

Diplom Informatiker (FH)

Upload von Dateien

- › Für den Upload einer Datei kann in einem Form ein `input` Element vom Typ `file` verwendet werden
- › Die Daten werden mittels der `Post` Methode an eine `ActionMethode` des `Controllers` übertragen
- › Der `Encoding` Typ (`enctype`) muss dazu auf `multipart/form-data` gesetzt werden
 - › Ohne diesen wird nur der Dateiname übertragen, nicht der Inhalt

In der View

› Formular für das Posten der Datei

FileUploadControl
Datei auswählen Keine ausgewählt

```
@using (Html.BeginForm ("Index",  
    "Home",  
    RequestMethod.Post,  
    new { enctype = "multipart/form-data" }))  
{  
    <input type="file" name="dieDatei" />  
    <input type="submit" />  
}
```

Mit Tag Helfern

```
<form asp-controller="Home" asp-action="Index" method="post"
  enctype="multipart/form-data">
  <input type="file" name="dieDatei" />
  <input type="submit" />
</form>
```

Action Methode

- › Die ActionMethode, welche für den Upload der Datei zuständig ist, benötigt einen Übergabeparameter vom Typ `IFormFile`
- › `[HttpPost]`
- › `public IActionResult UploadFile(IFormFile dieDatei) { ... }`

Die Schnittstelle IFormFile

```
public interface IFormFile
{
    string ContentType { get; }
    string ContentDisposition { get; }
    IDictionary Headers { get; }
    long Length { get; }
    string Name { get; }
    string FileName { get; }
    Stream OpenReadStream();
    void CopyTo(Stream target);
    Task CopyToAsync(Stream target, CancellationToken cancellationToken = null);
}
```

ContentDisposition	"form-data; name=\"file\"; filename=\"C:\\Users\\ik\\Downloads\\vswhere.exe\""
ContentType	"application/octet-stream"
FileName	"C:\\Users\\ik\\Downloads\\vswhere.exe"
Headers	{Microsoft.AspNetCore.Http.HeaderDictionary}
Length	411136
Name	"file"
Static members	
Non-Public members	

Die wichtigsten Eigenschaften von IFormFile

Name	Beschreibung
Length	Die Größe einer hochgeladenen Datei in Bytes
ContentType	Der MIME-Inhaltstyp einer hochgeladenen Datei
FileName	Der vollqualifizierte Name der Datei auf dem Client
Name	Der Name aus dem <code><input type="file" name=""></code>

Mehrere Dateien hochladen

- › Das `input` Element kann mit dem Attribut `multiple` versehen werden
- › Dadurch können in dem Dialog durch gedrückt halten der STRG-Taste mehrere Dateien ausgewählt werden

- › Beispiel

```
<form action="/Home/Upload" method="post" enctype="multipart/form-data">  
  <input type="file" name="alleDateien" multiple="multiple" />  
  <input type="submit" />  
</form>
```


Mehrere Dateien hochladen – Action Methode

› Der Übergabeparameter ist vom Typ `IEnumerable<IFormFile>`

› Beispiel

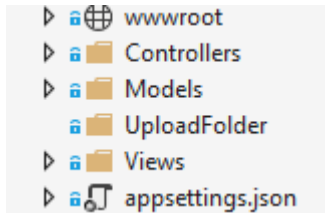
```
public ActionResult Upload(IEnumerable<IFormFile> alleDateien)
{
    if (alleDateien != null) {
        foreach (var datei in alleDateien) {
            if (datei.Length > 0) { ...
```

Hochgeladene Datei im Dateisystem ablegen

› Datei im Ordner „UploadFolder“ in der Wurzel der Anwendung speichern

› Pfad anlegen:

```
string path = Path.Combine(Directory.GetCurrentDirectory(),  
    "UploadFolder", Path.GetFileName(dieDatei.FileName));
```



› Datei im Pfad speichern:

```
using (FileStream stream = new FileStream(path, FileMode.Create))  
{  
    file.CopyTo(stream);  
}
```

Download von Dateien

- › Für den Download einer Datei wird eine ActionMethode mit einem Rückgabewert vom Typ `FileResult` benötigt
- › Der Rückgabewert dieser Methode ist vom Typ `File`

```
public FileResult DateiDownload() {  
    return File(...);  
}
```
- › In einer View kann der Download z.B. über einen Helper angesprochen werden
 - › `@Html.ActionLink("Herunterladen", "DateiDownload")`
 - › `<a asp-action="DateiDownload">Herunterladen`

Download von Dateien

- › Innerhalb der ActionMethode können z.B. die Klassen für Dateien und Ordner aus dem .NET-Framework verwendet werden
 - › DirectoryInfo, FileInfo, etc.
- › Die File Methode des Controllers ist überladen
- › Eine PDF-Datei kann z.B. wie folgt zurückgegeben werden
 - › `FileInfo fi = ...`
 - › `return File(fi.OpenRead(), "application/pdf", fi.Name);`