

# ASP.NET Core MVC Ajax Tag Helper

Ingo Köster

Diplom Informatiker (FH)

# Ajax

---

- › Für die Verwendung von AJAX mit ASP.NET MVC gibt es eigene Attribute von Form von Tag Helfern
- › Um diese nutzen zu können, ist das entsprechende clientseitige Paket notwendig

# Paket mittels LibMan Installieren

- › Als Anbieter cdnjs auswählen
- › Den Namen vollständig eingeben
  - › Ggf. eingeschränktes IntelliSense
  - › **jquery-ajax-unobtrusive**
- › Version wird automatisch angehängt

Clientseitige Bibliothek hinzufügen

Anbieter: cdnjs

Bibliothek: jquery-ajax-unobtrusive@3.2.6

☒ Alle Bibliotheksdateien einbeziehen

☐ Spezifische Dateien auswählen:

▲ ☒ Dateien:

- ☒ JS jquery.unobtrusive-ajax.js
- ☒ JS jquery.unobtrusive-ajax.min.js

Zielspeicherort: wwwroot/lib/jquery-ajax-unobtrusive/

Installieren Abbrechen

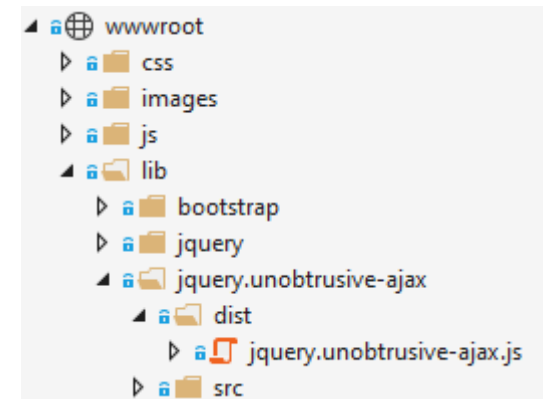
# Einbinden des Frameworks für den Client

---

- › Dieses kann z.B. mittels der `_Layout.cshtml` Datei in die gesamte Anwendung eingebunden werden
- › Oder nur die Views in denen die Tag Helper verwendet werden sollen

- › Mittels des passenden Pfades einbinden. Beispiel:

```
<script src="~/lib/jquery-ajax-unobtrusive/dist/jquery.unobtrusive-ajax.js"></script>
```



# ActionMethode

---

- › Da per Ajax HTML empfangen werden soll, welches anschließend in eine bestehende Seite eingefügt werden soll, wird eine ActionMethode verwendet, welche eine PartialView zurückliefert

```
public IActionResult PatientSuchen(string patientenName) {  
    var patients = ...  
    return PartialView("_patientList", patients);  
}
```

# Tag Helper für <form>

---

- › Im MVC Framework existieren die Tag Helper für das <form> Element
- › Die Attribute asp-controller & asp-action geben Controller und ActionMethode an für den Post an

```
<form asp-controller="Home" asp-action="PatientSuchen" ...>
```

...

```
</form>
```

# Ajax Helper

---

- › Damit die Daten nicht per POST sondern per Ajax versendet werden, wird das data-ajax Attribut verwendet
- › Um das per Ajax empfangene HTML in die Webseite einzufügen, wird die Id eines Elementes zum Einfügen angegeben

```
<form asp-controller="Home" asp-action="PatientSuchen"  
data-ajax="true" data-ajax-update="#results">
```

...

```
</form>
```

```
<div id="results"></div>
```

# JavaScript für Events

---

- › Der Helper kann mit JavaScript-Funktionen für gewisse Events kombiniert werden
- › In der View oder JavaScript-Dateien werden entsprechende Funktionen angelegt
- › In dem `<form>` Element werden für die Events die Namen der JavaScript Funktionen angegeben
- › Beispiele
  - › `data-ajax-begin`
  - › `data-ajax-complete`
  - › `data-ajax-failure`
  - › `data-ajax-success`



# JavaScript für Events - Beispiel

---

```
<script>  
let onBegin = function() {  
    ...  
};  
</script>
```

```
<form asp-controller="Home" asp-action="Index"  
    data-ajax-begin="onBegin" data-ajax="true"> ...  
</form>
```

# Ajax für Links mittels <a> und Tag Helfern

---

- › Erzeugt einen Link zur angegebenen ActionMethode
- › Beim Anklicken wird die Aktionsmethode asynchron mit JavaScript aufgerufen
- › Z.B. eine partielle View asynchron laden und anzeigen
  - › `<a asp-controller="Home" asp-action="PatientSuchen" data-ajax-update="#PatientenListe" data-ajax="true" asp-route-patientName="F">Anfangsbuchstabe F</a>`
- › ActionMethode
  - › `public IActionResult PatientSuchen(string patientName) { ... }`