

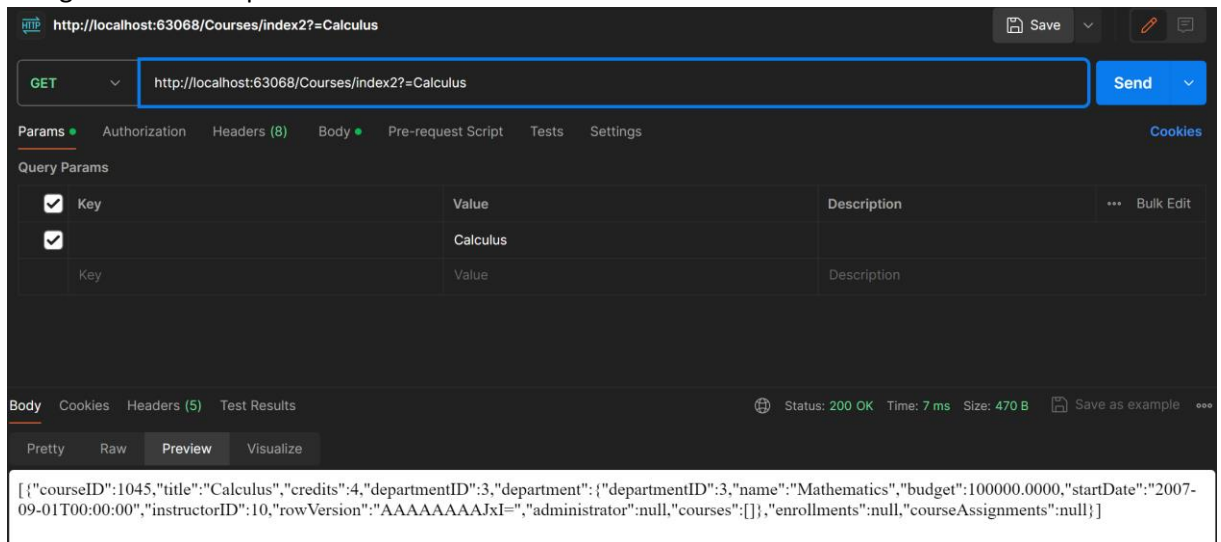
CST1 practicum software security les 3

Magic url opsporen in de CourseController:

Gebruik voor de volgende opdrachten Postman ipv de browser om een http request te maken.

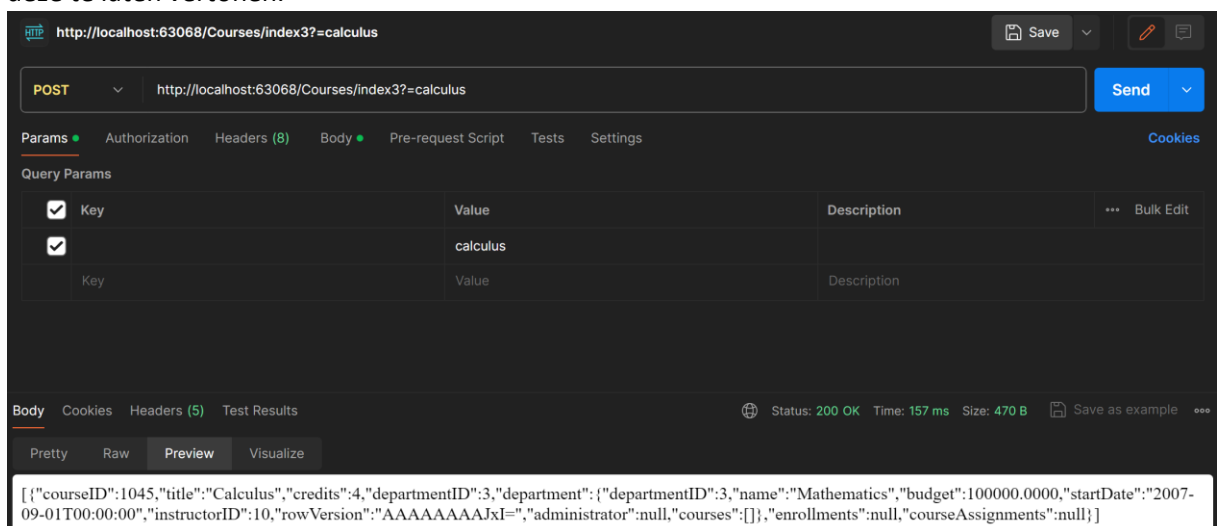
1. *De methode Index2 heeft de annotatie[HttpGet] met welke http method (GET, POST, enz) kan je deze aanroepen. Deze methode heeft een parameters searchString geef deze een waarde in het http request*

Je kunt deze methode aanroepen met de get methode. Je kunt in de searchString courses meegeven om hierop te zoeken.



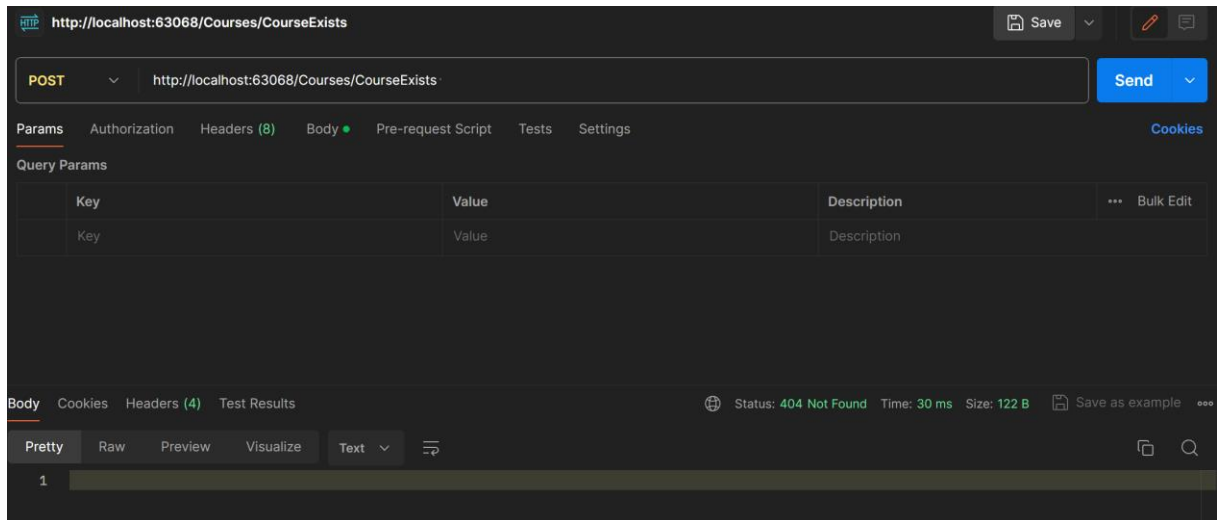
2. *Dezelfde vraag voor Index3*

Je roept deze methode aan met een post methode. Je kunt hierbij een course meegeven om deze te laten vertonen.



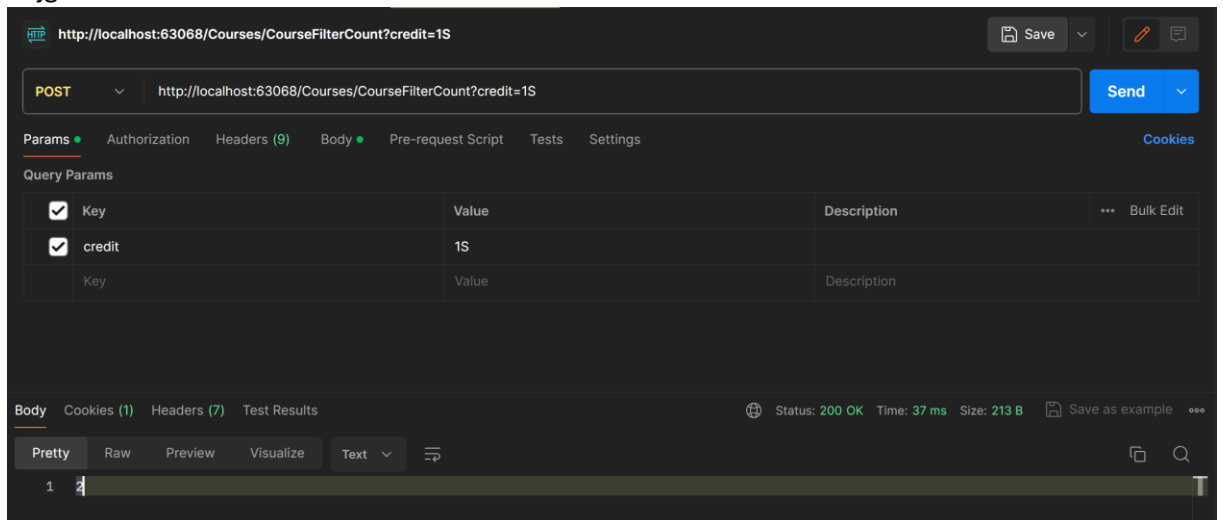
3. *Roep de methode CourseExists aan. Wordt de code in de methode in de code aangeroepen? Leg ook uit waarom de code wel of niet wordt aangeroepen.*

Je roept deze methode aan met een post method. De code wordt niet aangegeven aangezien deze onder een private bool staat.



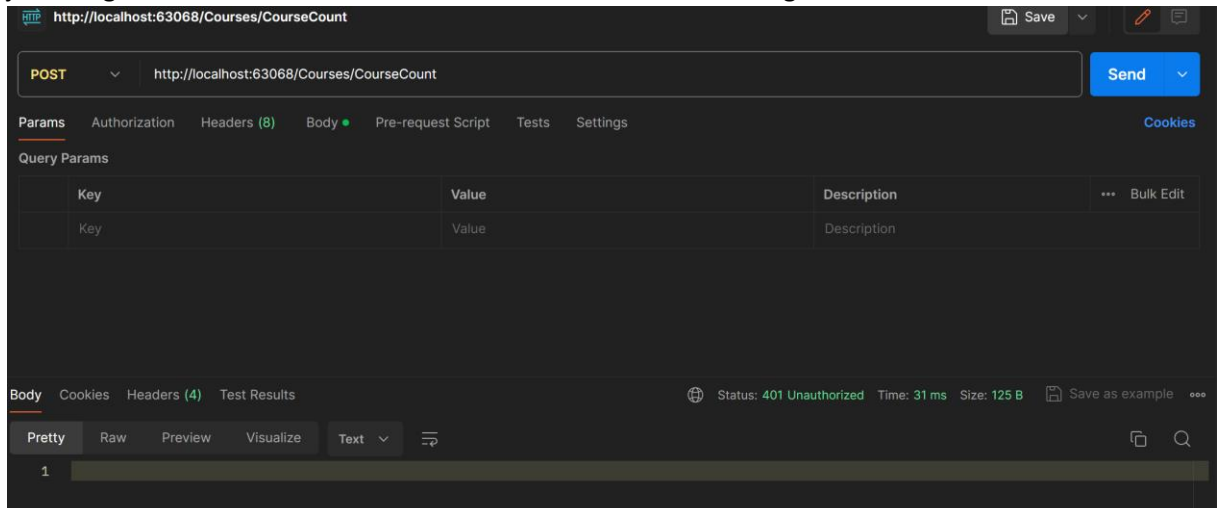
4. Dezelfde vraag voor de methode: CourseFilterCount

Je roept deze methode aan met een post method. Je kunt een credit nummer meegeven. Je krijgt dan de hoeveelheid courses met dit aantal credits te zien.



5. Dezelfde vraag voor de methode: CourseCount

Je roept deze methode aan met een post method. De code wordt niet weergegeven aangezien je eerst geautoriseerd moet worden voordat deze code wordt uitgevoerd.



6. *Wat doet de methode: UpdateCourseCredits*

Met deze methode kun je de credits waarde van de courses aanpassen.

De methode ontvangt een optionele int vanuit de input die multiplier wordt genoemd.

Vervolgens wordt er gekeken of de waarde van de multiplier niet leeg is. Als deze waarde niet leeg is, wordt de SQL opdracht uitgevoerd. De course credits worden geupdate door de huidige credits te vermenigvuldigen met de multiplier.

Vervolgens wordt View teruggegeven.

```
[HttpPost]
0 references
public async Task<IActionResult> UpdateCourseCredits(int? multiplier)
{
    if (multiplier != null)
    {
        ViewData["RowsAffected"] =
            await _context.Database.ExecuteSqlCommandAsync(
                "UPDATE Course SET Credits = Credits * {0}",
                parameters: multiplier);
    }
    return View();
}
```

7. *Waarom is deze een magic url? Hint probeer deze methode "te vinden" via de homepage*

Omdat deze URL te benaderen is maar niet geïndexeerd staat.

Hidden field hacken:

Kijk naar de html code van het formulier voor het wijzigen van een student

1. *Welke hidden fields wordt er gebruikt?*

```
... <input type="hidden" data-val="true" data-val-required="The ID field is required." id=
"ID" name="ID" value="1" > == $0
```

2. *Verander de waarde van het 1^e hidden field en wat is het effect? Aanpassen kan in Chrome via F12 (in het volgende practicum komt het andere hidden field die een token bevat aanbod)*

Zodra je hidden aanpast naar visible zie je dat er een nieuw input veld tevoorschijn komt. In dit veld staat de waarde 1 al ingevuld. Dit is het ID van de student.

Student

Last Name

Alexander <script>alert('Hello')</script>

First Name

Carson

Enrollment Date

01-09-2010

```
<nav class="navbar navbar-inverse navbar-fixed-top">...</nav>
<div class="container body-content">
  <h2>Edit</h2>
  <form action="/Students/Edit/1" method="post" novalidate="novalidate">
    <div class="form-horizontal">
      <div>Student</div>
      <div>
        <input type="visible" data-val="true" data-val-required="The ID field is required."
        id="ID" name="ID" value="1" > == $0
      </div>
      <div class="form-group">...</div>
      <div class="form-group">...</div>
      <div class="form-group">...</div>
    </div>
    <input name="__RequestVerificationToken" type="hidden" value="CFD18AM9GAP39Q81uG90retPmH
    Xap2PffecCMwV5t4588kunenloup18pov9HMKJ0f3t5pafw488Npp8MK-43phoByUJ1M_218w3sr7cmnqfxf6F
    Yuh4b-2CTV185VNBt1t1e21e1K1VX739mH5TKLA">
  </form>
</div>
```

Deze kan je aanpassen naar een ander ID. Zodra je dit doet schrijf je de data van de getoonde student over de student waarvan je het ID invult.

Last Name	First Name	Enrollment Date	
Alexander <script>alert('Hello')</script>	Carson	2010-09-01	Edit Details Delete
Alexander <script>alert('Hello')</script>	Carson	2010-09-01	Edit Details Delete
Anand	Arturo	2013-09-01	Edit Details Delete

Previous

Next

© 2017 - Contoso University

Client en server side validatie

Kijk naar de code voor het creëren van een nieuwe course. Dit formulier is iets aangepast tov de vorige versie.

1. Welke validatie er zit in de server side code en welke in client side code

Server side: “`if (course.Credits<=10 && course.Title.Length>3)`”. Dit oud in dat dat de course credits 10 of lager moeten zijn en ook de titel lengte groter moet zijn dan 3

Client side: zodra een input niet correct is, is de create knop disabled. Zodra de input wel volgens de validatie correct is, is deze disabled weg. Er is een client side validatie bij credits die op 5 en lager staat.

Course

Number
101

Title
TestCourse

Credits
7
Credits kan maximaal 5 zijn

Department
Mathematics

Create

[Back to List](#)

```

field is required." id="DepartmentID" name="DepartmentID"></select>
<span class="text-danger field-validation-valid" data-valmsg-for="DepartmentID"
data-valmsg-replace="true"> </span>
</div>
::after
</div>
<div class="form-group">
::before
<div class="col-md-offset-2 col-md-10">
<input type="submit" id="submitButton" value="Create" disabled>
</div>
::after
</div>
</div>
</form>
<script></script>
</div>
</div>
<hr>
<footer></footer>
::after
</div>
<script src="//lib/jquery/dist/jquery.js"></script>

```

2. Kan je een course aanmaken met 7 credits? Kan een normale gebruiker dit via de browser

Nee, dit wordt geblokkeerd door de input validatie. Je krijgt een melding dat credits maximaal 5 kan zijn. Dit komt door de client side validatie

3. Kan je een course aanmaken met 7 credits? Als een gebruiker via F12 het document object model aanpast.

Ja je kan een course aanmaken met 7 credits als gebruiker zodra je het document object model aanpast. Zodra je disabled weg haalt of veranderd naar enabled kan de course aangemaakt worden. Dit komt omdat je hiermee de client side validatie negeert.

```

<div class="col-md-offset-2 col-md-10">
...
<input type="submit" id="submitButton" value="Create" enabled> == $0
</div>

```

4. *Dezelfde vraag maar dan voor 12 studiepunten? Leg uit waarom er een verschil is met de vorige vraag.*

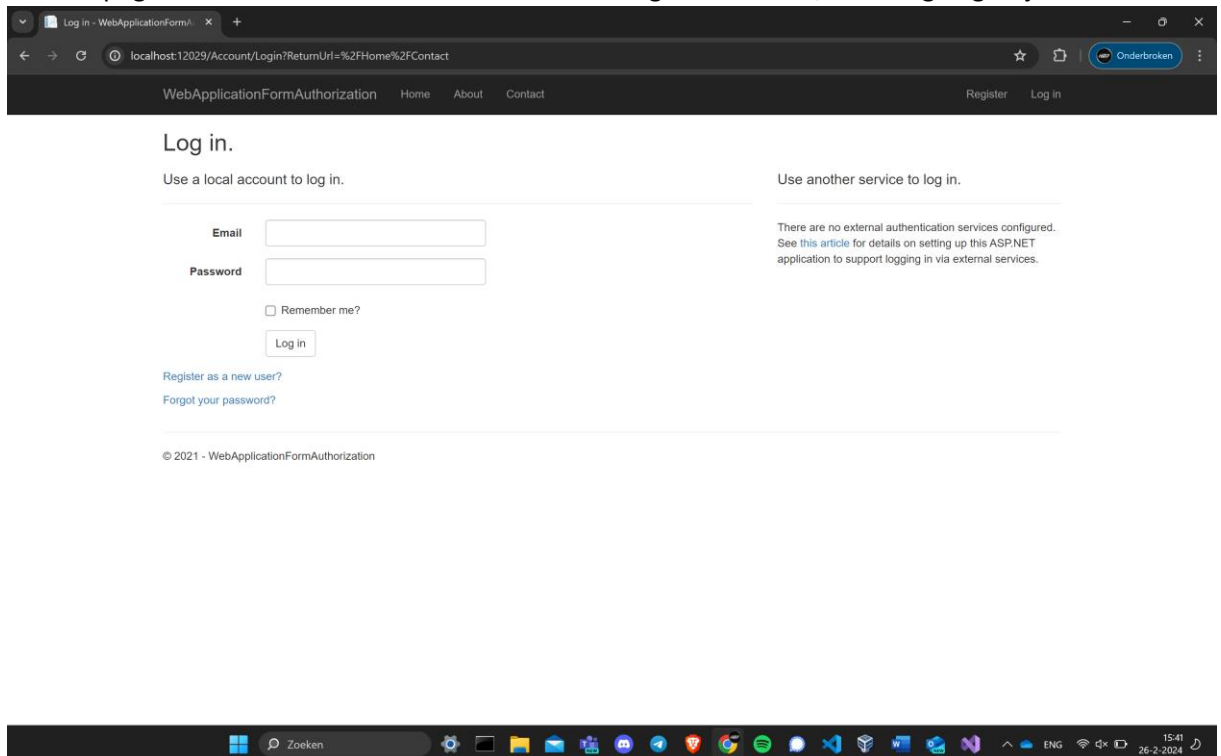
Dit werkt niet. Dit komt omdat de server side validatie op 10 en lager staat. Zodra je van 5 naar 7 gaat wordt dit in het begin tegengehouden door de client side validatie. Zodra je dit negeert kan het nog aangezien de server side validatie 7 nog wel toestaat. 12 is hoger dan de server side validatie. Hierdoor wordt deze input tegengehouden.

WebapplicationFormAuthentication

Open nu dit andere Visual Studio project voor deze week. Deze applicatie is 100% gegenereerde code met functionaliteit voor het inloggen. Voordat je de applicatie start, maak eerst de database aan met het bekende commando: dotnet ef database update

1. *Ga naar de home pagina en daarna naar de contact pagina. Wat zie je?*

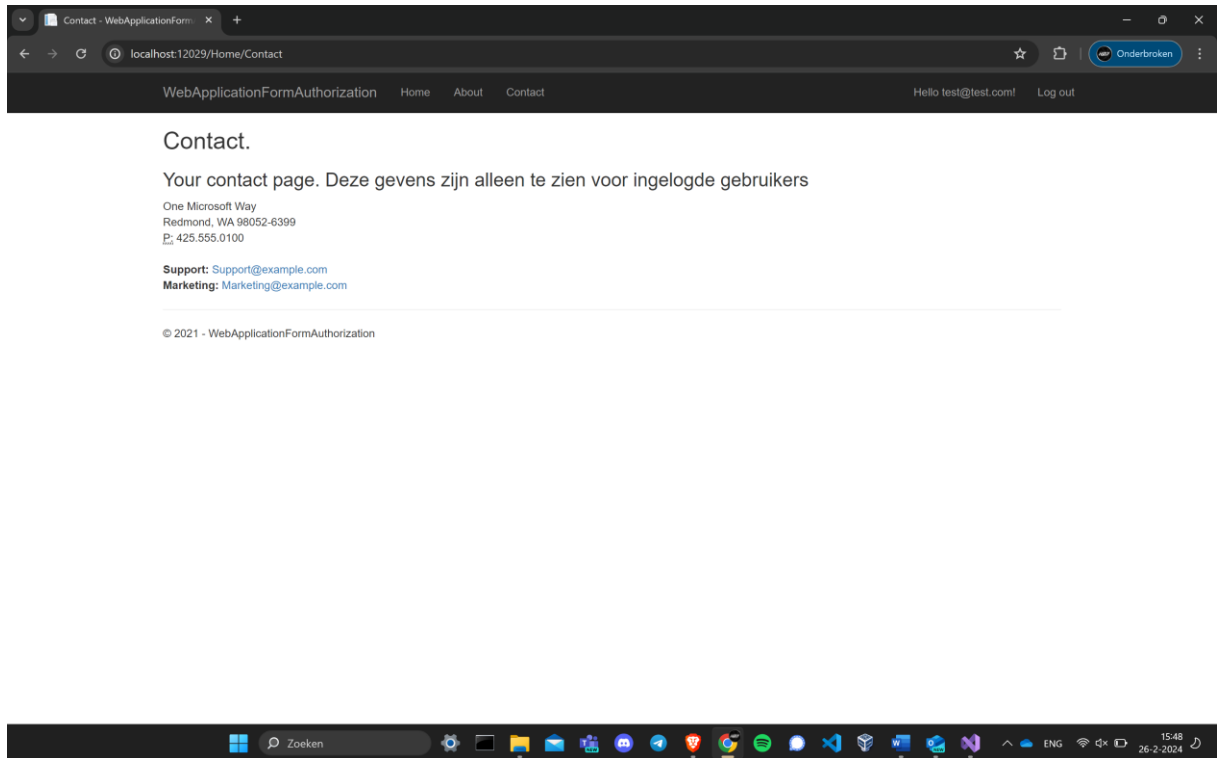
Je ziet een pagina waarbij kan inloggen, een nieuwe gebruiker kan registreren en een functie om een vergeten wachtwoord op te vragen. Dit is een login pagina. Ik verneem dat de contact pagina alleen beschikbaar is voor mensen die geautoriseerd, aldus ingelogd zijn.



2. *Kijk naar de code en leg uit waarom je de home pagina wel kan zien en de contact pagina niet*
Door het stukje "[Authorize]" moet je geautoriseerd zijn om het volgende stukje code te kunnen gebruiken. In dit geval is het de contact pagina. Er staat ook dat de contact gegevens alleen te zien zijn voor ingelogde gebruikers.

```
[Authorize]
public IActionResult Contact()
{
    ViewData["Message"] = "Your contact page. Deze gegevens zijn alleen te zien voor ingelogde gebruikers";
    return View();
}
```

3. *Registreer een nieuwe gebruiker en ga daarna naar de contact pagina. Wat gebeurt er nu?*
Als je ingelogd bent kan je de contact gegevens nu wel bekijken. Dit komt omdat je na het inloggen geautoriseerd bent voor deze pagina.



4. *Log uit en log weer in. Welke action methode is er verantwoordelijk voor om te controleren dat je wachtwoord klopt? Hoe ontvangt deze methode de gebruikersnaam en wachtwoord? Laat dit zien door een breakpoint te plaatsen in de code.*

Er wordt gebruik gemaakt van een post request om te controleren of je wachtwoord klopt.

```
// POST: /Account/Login
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
```

De methode ontvangt de gebruikersnaam en wachtwoord via deze code.

```
// To enable password failures to trigger account lockout, set lockoutOnFailure: true
var result = await _signInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, lockoutOnFailure: false);
if (result.Succeeded)
{
    _logger.LogInformation(1, "User logged in.");
    return RedirectToLocal(returnUrl);
}
```

5. *Open de bijbehorende database (via SQL Server Object Explorer) en noteer welke gegevens je in de database kan vinden over de geregistreerde user.*

Hieronder zie je de zichtbare gegevens over de geregistreerde user.

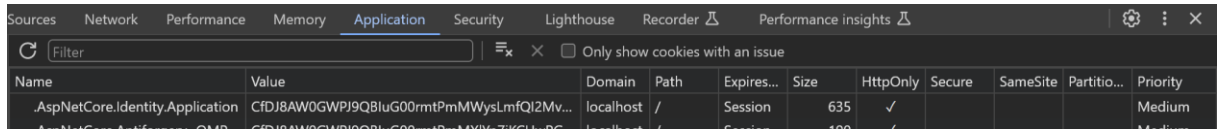
AccountController.cs													
dbo.AspNetUsers [Data]													
ID	AccessFailedCount	ConcurrencyStamp	Email	EmailConfirmed	LockoutEnabled	LockoutEnd	NormalizedEmail	NormalizedUserName	PasswordHash	PhoneNumber	PhoneNumberConfirmed	SecurityStamp	TwoFactorEnabled
18-65e804ab666	0	1ae17614-1be9...	test@test.com	False	True	NULL	TEST@TEST.COM	TEST@TEST.COM	AQAAAAEAAACc...	NULL	False	Sefacd11-9763...	False
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. *Leg uit waarom deze website eigenlijk TLS zou moeten gebruiken?*

De website zou TLS moeten gebruiken aangezien het feit dat TLS-encryptie kan helpen webapplicaties te beschermen tegen datalekken en andere aanvallen.

7. Welke cookie is na afloop van het inloggen geplaatst? Zie F12

Hieronder zie je welke cookie is geplaatst na het inloggen.

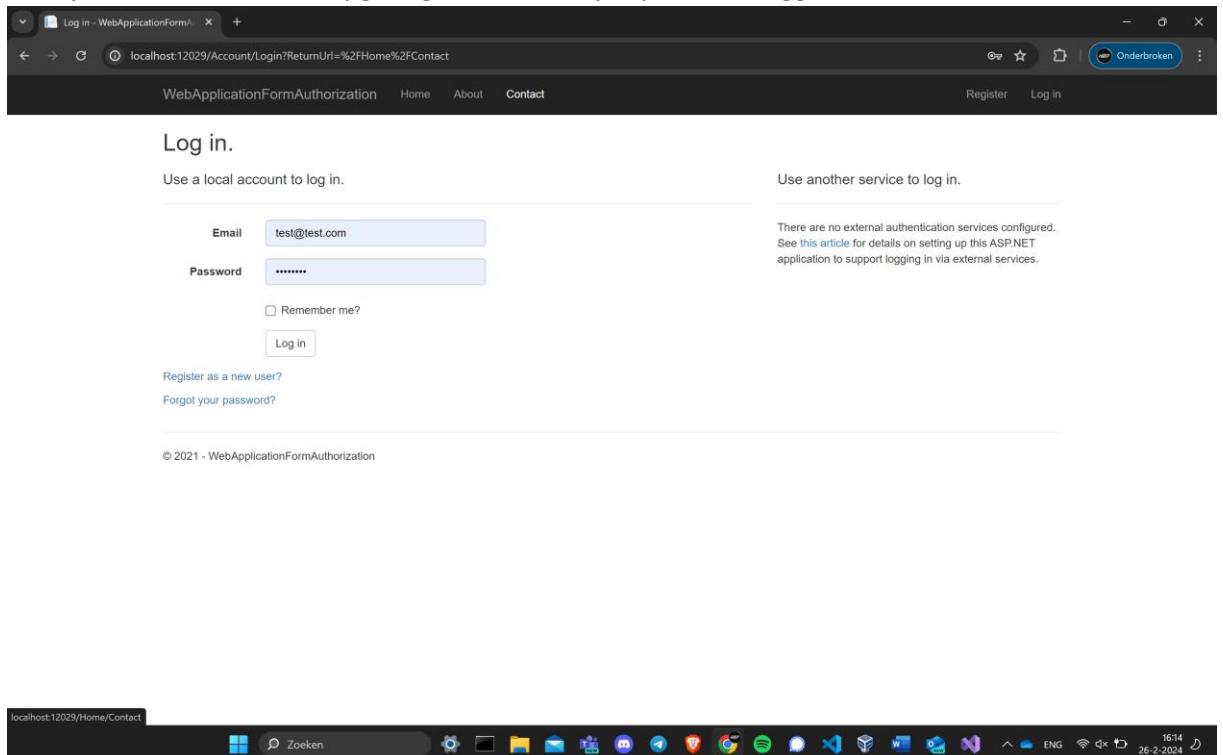


The screenshot shows the Chrome DevTools Application tab with the 'Cookies' section selected. A table lists cookies for the current page. The first cookie is highlighted.

Name	Value	Domain	Path	Expires...	Size	HttpOnly	Secure	SameSite	Partitio...	Priority
.AspNetCore.Identity.Application	CfDJ8AW0GWPJ9Q8luG00rmtPmMWysLmfQI2Mv...	localhost	/	Session	635	✓				Medium

8. Wat gebeurt er als je deze cookie verwijdert en naar de contact pagina gaat?

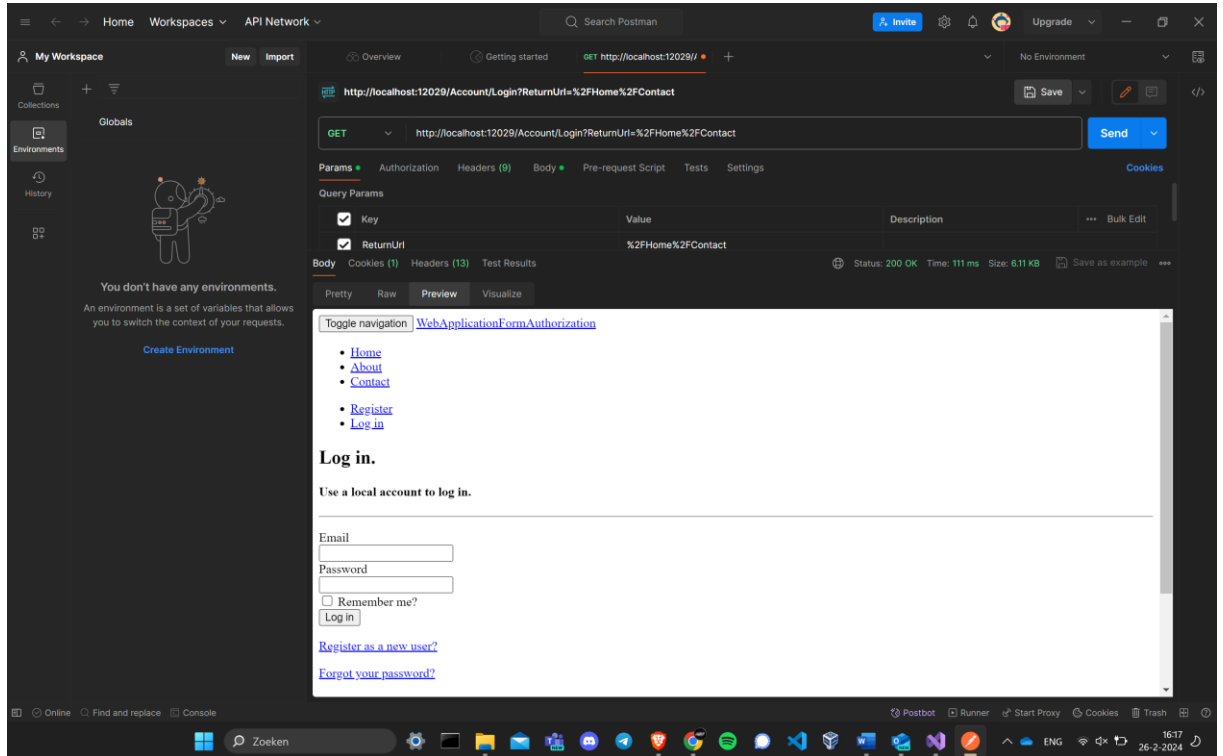
Als je deze cookie verwijdert en dan naar de contact pagina gaat, krijg je de inlogpagina te zien. Dit komt omdat de cookie je login voor een bepaalde tijd bewaard. Hiermee kan je dan over de website navigeren zonder dat je constant opnieuw hoeft in te loggen. Zodra je deze verwijderd is dit niet meer opgeslagen, dus moet je opnieuw inloggen.



The screenshot shows a web browser window at localhost:12029. The page title is 'WebApplicationFormAuthorization'. The main heading is 'Log in.' Below it, there are two sections: 'Use a local account to log in.' and 'Use another service to log in.' The 'Use a local account to log in.' section contains a form with 'Email' (test@test.com) and 'Password' (masked with dots) fields, a 'Remember me?' checkbox, and a 'Log in' button. Below the form are links for 'Register as a new user?' and 'Forgot your password?'. The 'Use another service to log in.' section has a message: 'There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.' The footer shows '© 2021 - WebApplicationFormAuthorization'. The browser's address bar shows 'localhost:12029/Account/Login?ReturnUrl=%2FHome%2FContact'. The Windows taskbar at the bottom shows the time as 16:14 on 26-2-2024.

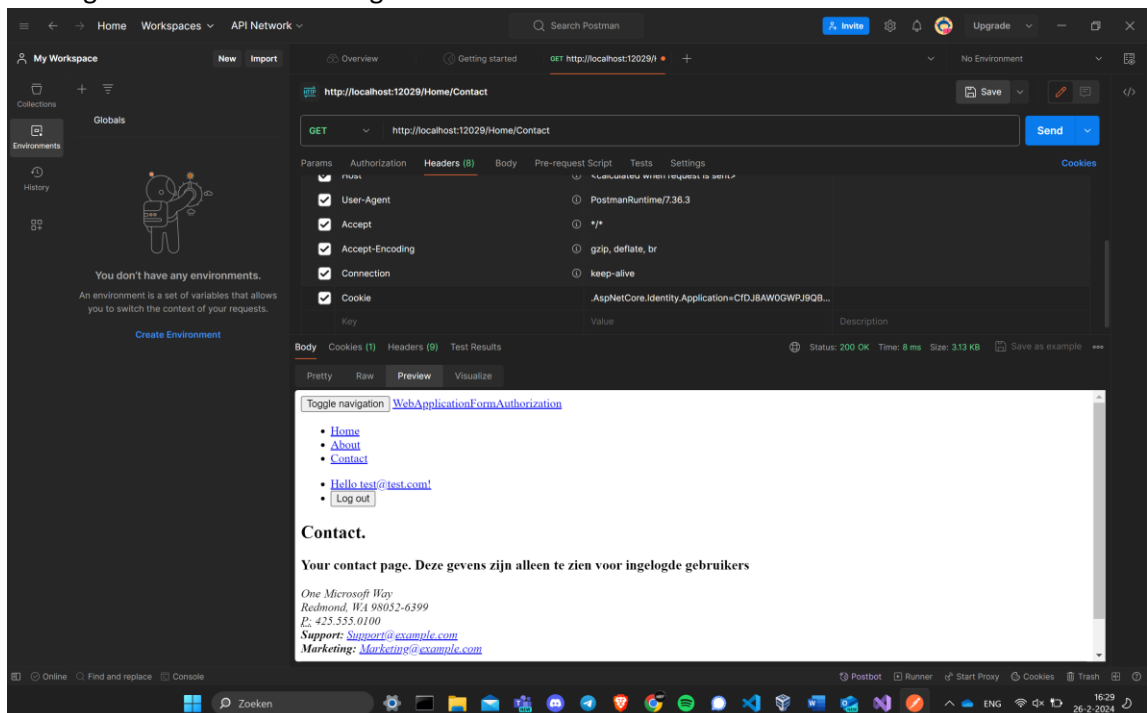
9. Open de contact pagina via postman (GET request naar de url van de contactpagina) Wat zie je? Als je bij de response (klikt op preview ipv pretty) is het makkelijker te zien welke html pagina je hebt gekregen als response op het http request

Je ziet een preview van de login pagina. Ik ben nog niet ingelogd dus vandaar dat ik ook via postman geen toegang heb tot deze pagina.



10. Voeg in postman deze cookie toe als header field. Je hebt nu zowel de key als de value van de cookie nodig. Vraag weer de contactpagina op. Wat krijg je nu als response?

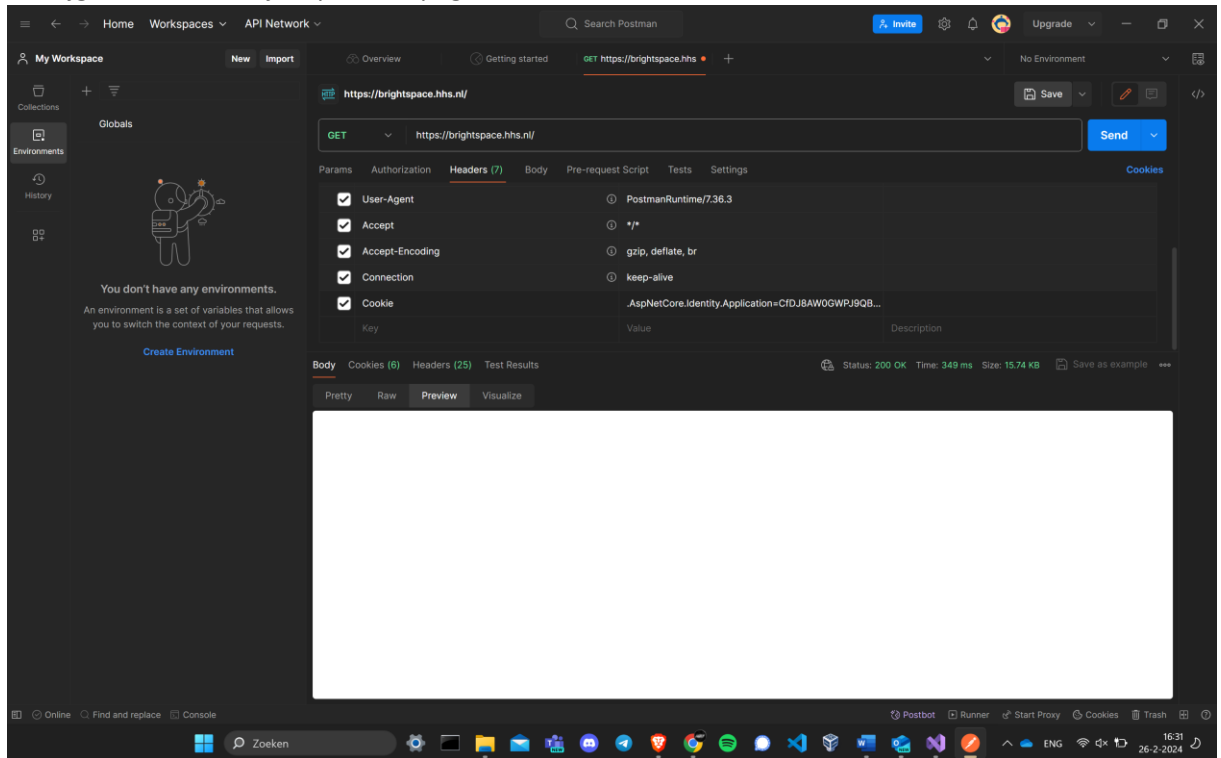
Als je de cookies invult krijg je de contact pagina te zien die je ook ziet als je bent ingelogd. Ik zie de gebruiker die ik heb aangemaakt en de contactinformatie van de website.



Toegang tot brightspace

11. Open in postman de url: <https://brightspace.hhs.nl>. Welke response krijg je? Als je bij de response (klikt op preview ipv pretty) is het makkelijker te zien welke html pagina je hebt gekregen als response op het http request

Je krijgt niks te zien bij de preview pagina.

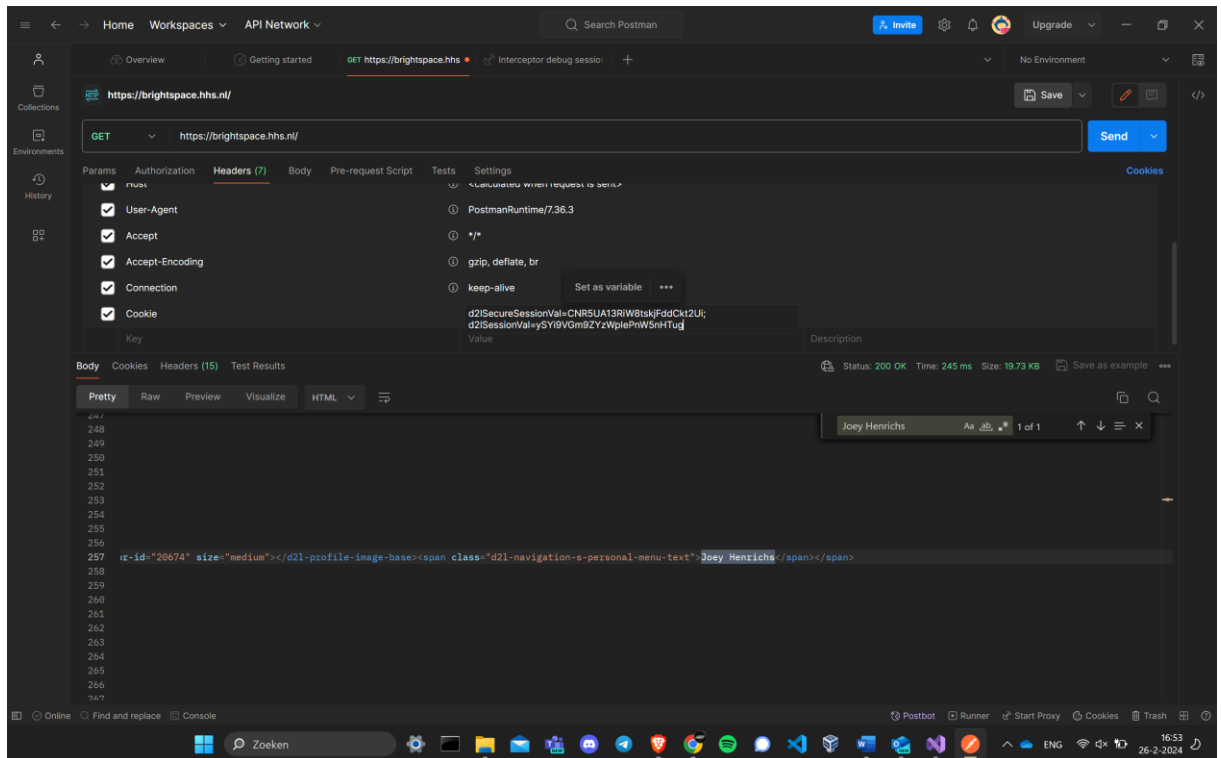


12. Open nu in de browser brightspace en log zonodig in. Ga naar F12 en kijk naar de cookies. Kopieer de value van d2ISessionVal= en d2ISecureSessionVal=

Name	Value	Domain	Path	Expire...	Size	HttpO...	Secure	Same...	Partiti...	Priority
ShibbolethSSO	Shibboleth	bright...	/	2024...	23		✓	None		Mediu...
_shibsession_64656661756c7468...	_b42421b9864f1a964e94f768d8f280ef	bright...	/	Session	140	✓	✓	None		Mediu...
d2ISecureSessionVal	1KgIrlQJcqS3fZr5be8G8D4gu	bright...	/	Session	44	✓	✓	None		Mediu...
d2ISessionVal	97hRT6wfK2OmESQ0LMUMNR71K	bright...	/	Session	38	✓	✓	None		Mediu...

13. Voeg in postman deze cookie toe als header field. Je hebt nu zowel de key als de value van de cookie nodig. Vraag weer de homepage van brightspace op. Wat krijg je nu als response? Zoek in de response maar naar je eigen naam.

Ik heb beide toegevoegd als value aan cookie. Om ze van elkaar te scheiden wordt “;” gebruikt. Zodra je dit invult als cookie en op send drukt, krijg je de html code die je zou krijgen als je bent ingelogd. Hierbij kan je zoeken op je naam en wordt dit getoond.



14. Waarom krijg je nu een hele andere response? Leg dit uit.

Door de cookies in te vullen bezoek je de website alsof je al bent ingelogd. De cookies zijn een soort vervanging voor het inloggen. Anders zou je op elke nieuwe pagina opnieuw moeten inloggen. Doordat je de cookies invult, kun je over de website browsen alsof je bent ingelogd.