

# Deployment

## 1 Toegang tot PLESK

[datasource](#)

Ga na bij een Software Development docent en meld dat je met deze module start.

De docent zal alles voor je klaarzetten!

Om te beginnen heb je toegang nodig tot PLESK. Hiervoor ga je met je browser naar het volgende adres:

<https://jouw.website/>

Vervolgens krijg je een inlogscherf. Klik hier op 'Wachtwoord vergeten?' om een link te krijgen voor het aanmaken van een wachtwoord. Je inlognaam is je studentnummer.

Om een wachtwoord te resetten (of op te vragen) gebruik je je email adres

[studentnummer@talnet.nl](mailto:studentnummer@talnet.nl),

bijvoorbeeld : [123456@talnet.nl](mailto:123456@talnet.nl)

Vergeet niet je SPAM-box te checken.

Als je bent ingelogd in PLESK, kom je gelijk in de beheeromgeving van jouw website. De url van jouw website is:

<jouw studentnummer>.jouw.website

Als je deze url voor de eerste keer intyped zie je een standaard (tijdelijke) pagina. Deze ga je in de volgende onderdelen vervangen door de door jou gemaakte website- / applicatie. Veel browsers zullen deze pagina niet direct laten zien. Dit komt omdat de site nog niet beveiligd is met een SSL certificaat (slotje bij je browser url).

## ? Inleveren

- Screenshot van je browser met de PLESK-omgeving en zichtbare domeinnaam/URL.

# 2 Beveilig het domein met een SSL? certificaat

Start met het bekijken van je domein in de browser: `<studentNummer>.jouw.website`. Let op het slot-icoon in de adresbalk.

## Stap 1

Voer jouw **domeinnaam** in, in je browser. Er kunnen afhankelijk van je browser, computer, virusscanner verschillende dingen gebeuren. Kijk wat je ziet. En kijk ook of er naast de domeinnaam / url in je adresbalk een slotje te zien is. Hoe ziet dat er uit?

### Wat was ook alweer een domeinnaam?

Een **domeinnaam** is het adres van een website dat je intoetst in de browser, zoals:

[www.google.com](http://www.google.com)

## Stap 2

Log in op je PLESK-omgeving en ga naar het onderdeel 'SSL/TLS-certificaten' onder 'Beveiliging'. Hier zie je verschillende mogelijkheden die je hebt om certificaten te installeren. Dit is per beheersysteem verschillend en ligt er aan hoe een beheersysteem is ingericht. Wij maken gebruik van het soort certificaat van 'Let's Encrypt'. Dit is een gratis open source certificaat dat vaak is geïnstalleerd op een beheersysteem (PLESK)

### Wat is een certificaat?

Een **certificaat** op internet is een soort digitaal paspoort voor een website. Het bewijst dat de website een veilige verbinding maakt.

Bijvoorbeeld:

Als je `https://` ziet en een slotje in de adresbalk, dan gebruikt de website een certificaat.

“ Een certificaat zorgt ervoor dat de verbinding tussen jou en de website veilig is (versleuteld), maar het garandeert **niet** dat de website zelf te vertrouwen is.

Een nepwebsite (zoals `ing-bank-login.xyz`) kan ook gewoon een geldig certificaat hebben en een slotje tonen. Daarom moet je **altijd goed naar het adres kijken** en alleen websites vertrouwen die je kent of via een betrouwbare link bezoekt.

ssl-1.png

**Stap 3:** Klik onder aan de certificatenpagina op ' **Installeren** ' bij het onderdeel 'Let's Encrypt'

ssl-2.png

Hierna krijg je opties voor wat je op je domein wilt beveiligen. Het belangrijkste voor nu is de domeinnaam. Maar bijvoorbeeld mail kun je hier ook laten versleutelen met een certificaat.

## Stap 4

Klik aan wat je nodig hebt. Je kunt alles aanvinken behalve de 'Wildcard'optie. Hiervoor zijn meerdere handelingen nodig.

### Wat is een wildcard?

Een wildcard-optie betekent dat je met één instelling o meerdere variaties in één keer dekt.

Een \* staat bij een wildcard voor "alles" of "maakt niet uit wat hier staat".

## Stap 5

Bekijk de site nogmaals in je browser (eventueel refreshen) en kijk of je verschil ziet en of het domein is beveiligd.

## Stap 6

Maak een screenshot van je browser met hier in de adresbalk waar jou domeinnaam + het beveiligingsslotje goed zichtbaar is. Stuur ook jouw url in.

## Werkt het niet?

Het kan zijn dat er in het geheugen van jouw browser nog staat dat de website niet veilig is. Met het volgende commando kan je dat geheugen leeg maken.

Ga naar CMD (command box) en type daar in:

```
ipconfig /flushdns
```

image.png

## ? Reflectie – SSL-certificaat installeren

- Wat is het doel van een SSL-certificaat?
- Wat verandert er zichtbaar in je browser als een site beveiligd is?
- Wat kan er misgaan als een website géén SSL-certificaat heeft (zoek dit eventueel op)?
- Welk type certificaat heb je gekozen en waarom was de 'Wildcard' optie niet geschikt?

## ? Inleveren

- Lever je reflectie in als .txt of .pdf-bestand.
- Screenshot van de browser met het beveiligde domein (slotje zichtbaar).
- URL van je domein.

## *3 Maak een FTP-account aan*

Maak via PLESK een FTP-account met zelfgekozen gebruikersnaam en wachtwoord. Stel de hoofdmap in op de root-directory van je (live) website.

## ? Reflectie – FTP-account aanmaken

- Wat is het doel van een FTP-account bij het beheren van je website?
- Welke gegevens moet je onthouden of noteren bij het aanmaken van een FTP-account?

## ? Inleveren

- Lever je reflectie in als .txt of .pdf-bestand.

## *4 FTP-verbinding maken en bestanden plaatsen*

### ↓ **FTP-client downloaden en installeren**

Je mag natuurlijk alle programma's die je fijn vindt gebruiken. In de uitleg gaan we kijken naar de client van FileZilla (zelfde organisatie als FireFox). Dit is een gratis programma dat je hier kunt downloaden:

<https://filezilla-project.org/download.php?type=client>

! let er op dat je de **clientversie** gebruikt.

Installeer dit programma op je computer.

---

## ☐☐FTP-verbinding maken

Nu je de FTP-client hebt geïnstalleerd gaan we een verbinding maken met jouw domein op de server. Om dit te doen heb je 3 gegevens nodig:

- **Host** (server)  
Hier kun je je domeinnaam invoeren: <studentnummer>.jouw.website
- **Gebruiker**  
(zoals je deze hebt aangemaakt bij het maken van een ftp-account op de server)
- **Wachtwoord**  
(zoals je deze hebt aangemaakt bij het maken van een ftp-account op de server)

Volg onderstaande stappen om een verbinding te maken:

- open FileZilla
- open 'sitebeheer' via:  
bestanden -> sitebeheer
- klik op 'nieuwe site'
- geef de site een naam in het linker overzichtsscherf
- Voer je host, je gebruikersnaam en wachtwoord in
- klik je op 'verbinden'

filezilla.png

Als het goed is, ben je nu ingelogd en zie je de inhoud van je servermap waar je een site kunt plaatsen.

filezilla2.png

Verwijder de bestanden die je ziet op de server.

Maak een eigen index.html bestand aan op je computer. Maak hier een leuke pagina van met html-code. Plaats deze vervolgens op de server via FTP. Als je FileZilla gebruikt zie je in het linker gedeelte van je scherm, bestanden op jouw computer en rechts die op de server.

Bekijk je nieuwe pagina in je browser door naar je site te gaan!

Mooi, he!?

## ? Reflectie – FTP-account aanmaken

- Wat kan er gebeuren als je een verkeerde hoofdmap instelt bij je FTP-account?
- Wat zou je een volgende keer anders aanpakken bij het instellen van FTP-toegang?

## ? Inleveren

- URL van jouw domein met de nieuwe indexpagina.
- Lever je reflectie in als .txt of .pdf-bestand.

## *5 Database aanmaken*

Maak in PLESK een nieuwe database én gebruiker aan:

Geef de database de naam:

Je ziet de database in het overzicht; open phpMyAdmin daarvoor.

## ? Inleveren

- Screenshot van phpMyAdmin in je browser inclusief adresbalk en database-naam.

## *6 Webapplicatie installeren*

In de [CRUD Challenge](#) heb je een applicatie gemaakt, waarmee geregistreerd kon worden wie te laat in de klas kwam en de reden daarvan. Weet je nog!? Dit ga je nu live zetten op jouw server.

image.png

Nu is het tijd om jouw applicatie 'live' te zetten op de server, zodat iedereen via Internet er gebruik van kan maken. ! Hou er rekening mee dat het kan zijn dat je code moet aanpassen om een goede

verbinding te maken met de database. Tip: Net als in xampp, kun je als hostname / servername 'localhost' gebruiken. Inleveren : De url van jouw (werkende) applicatie

☐ Let op: je moet mogelijk de database-connectie in je code aanpassen (bijv. hostname = 'localhost').

## ? Inleveren

- URL van je live, werkende applicatie.

# 7 Test je Kennis

### Wat is Plesk?

**Plesk** is een soort **bedieningspaneel** voor je website. Je kunt het vergelijken met het dashboard van een auto, maar dan voor je webserver.

#### ☐ Wat kun je ermee doen?

- Je eigen **domeinnaam** instellen (zoals www.jouwsite.nl)
- **Bestanden uploaden** van je website
- Een **database aanmaken**
- Een **SSL-certificaat** installeren (voor de ☐ in je adresbalk)
- **E-mailadressen beheren** (zoals info@jouwsite.nl)

☐ Plesk maakt het dus makkelijk om je website online te zetten en te beheren, zonder dat je moeilijke servercommando's hoeft te leren.

### Wat doet een SSL-certificaat en waarom is dat belangrijk?

Een SSL-certificaat zorgt ervoor dat gegevens tussen bezoeker en website versleuteld zijn.

Je herkent het aan de ☐ in de adresbalk. Het beschermt tegen afluisteren en verhoogt vertrouwen, maar zegt niets over de betrouwbaarheid van de site zelf.

### Wat betekent 'wildcard'-optie bij certificaten en waarom kies je die niet meteen?

Een wildcard-certificaat dekt alle subdomeinen (bijv. \*.jouw.website). Voor eenvoudige sites heb je die vaak niet nodig, en het vereist extra configuratie, dus kies je meestal alleen jouw domein.

### Waarom moet je een FTP-account aanmaken en waar is dit goed voor?

Met een FTP-account kun je verbinding maken met de server en bestanden (zoals je website) uploaden of bijwerken. Het is dé manier om jouw code 'live' te zetten.

### Waarom is het nodig de database-connectie opnieuw in te stellen bij deployment van je CRUD-applicatie?

In je lokale development omgeving gebruik je meestal gebruiker 'root' zonder wachtwoord (lekker makkelijk!).

Op de productieserver moet je de het wachtwoord en liefst ook de gebruikersnaam aanpassen om te voorkomen dat je database makkelijker wordt gehakkeld en dat de informatie in verkeerde handen komt.

## ? Reflectie-opdracht – Van lokaal naar live

Tijdens deze module heb je een webapplicatie **van je eigen computer naar een echte server** gebracht. Dat betekent dat je hebt gewerkt met Plesk, FTP, SSL en een database op een server.

## ?? Opdracht

Maak een reflectie en beantwoord de volgende vragen.

1. Wat vond je het lastigste onderdeel van deze opdracht?  
Leg uit waarom dit lastig was en hoe je het uiteindelijk hebt opgelost.
2. Waar heb je AI ingezet en waarbij heeft dat jou goed geholpen?
3. Wat is volgens jou het belangrijkste verschil tussen een website draaien op je eigen computer (bijv. XAMPP) en op een echte server?
4. Vind je dat elke software developer moet begrijpen hoe deployment werkt (servers, databases, SSL)?  
Leg uit waarom wel of niet.
5. Stel dat je deze webapplicatie opnieuw zou moeten deployen.

1. Welke stappen zou je hetzelfde doen?
2. Welke stappen zou je anders aanpakken of beter voorbereiden?

# ? Inleveren

Persoonlijke reflectie in PDF

--

---

---

Revision #19

Created 2025-06-18 12:11:30 UTC by Max

Updated 2026-04-10 12:45:36 UTC by Max