

Leerlijn	Backend Programming
Kenmerken	Code: BEP Studiepunten: 30 Aantal cursussen: 5 Aantal bijeenkomsten: 18 Toetsvorm: Integrale eindopdracht
Studiegidsbeschrijving	<i>Steeds meer programma's staan tegenwoordig niet enkel meer op je eigen computer, tablet of telefoon. Door informatie en diensten op een server te zetten kunnen deze gemakkelijk gedeeld worden met anderen en kan de gebruiker op elk device bij zijn eigen gegevens. De server zorgt bijvoorbeeld voor het versturen en ontvangen van gegevens, berichten en foto's, het streamen van video's en het spelen van online spellen. In Backend Programming gaan we zulke server applicaties opzetten - met Java en het Spring framework. Op basis van praktijkvoorbeelden koppelen we de server met een database en een web applicatie terwijl we dieper ingaan op de logica en structuren van programmeren.</i>
Beschrijving leerlijn	<p>Zoals koks het voedsel in de keuken beheren, doen servers dat met de data in databases. In deze leerlijn leer je een veilig online platformen maken met het Java Spring framework. Je leert om gebruikersdata in SQL databases te beheren waarbij je zorgt dat gebruikers enkel toegang krijgen tot de informatie en acties waar ze recht op hebben. Je verdiept je kennis van OOP-structuren en design patterns om ook in grotere projecten het overzicht in je code te bewaren. Je vergroot je inzicht in het software ontwikkeltraject door een technisch ontwerp te maken en met behulp van Git je ontwikkelomgeving op te zetten.</p> <p>Tijdens de gehele leerlijn krijg je stap voor stap praktijkvoorbeelden waarin we je de vaardigheden aanleren voor een Junior Backend Developer.</p> <p>Voor de eindtoets doorloop je zelf deze stappen om een beveiligde backend op te zetten. Deze backend mag je koppelen met de website die je maakt bij de leerlijn Frontend Programming, maar je mag ook geheel iets anders maken.</p>
Cursussen	<ol style="list-style-type: none"> <li>1. Java Programmeren</li> <li>2. Backend Documentatie</li> <li>3. Database Development*</li> <li>4. Spring Boot</li> <li>5. Design Patterns &amp; Clean Code</li> </ol> <p>* Deze cursus dient zelfstandig te worden doorgenomen in EdHub.</p>

<b>Cursus</b>	<b>Java Programmeren</b>
Leeruitkomst	De student programmeert in Java, waarbij hij OOP-structuren toepast. Hierbij past de student automatisch testen toe en beheert hij externe code met behulp van Maven, waardoor men in een team aan Java-projecten kan werken.
Toelichting cursus	<p>In de cursus Java programmeren leer je structuren van objectgeoriënteerd programmeren (OOP). Met OOP-structuren leer je ook je code leesbaar, uitbreidbaar en overzichtelijk te houden.</p> <p>Je gaat interfaces en abstracte klassen gebruiken, daarnaast ga je unit-testen schrijven om de code automatisch te kunnen testen. Uiteindelijk leer je gebruik te maken van externe code middels Maven.</p>
Prestatie indicatoren	<ul style="list-style-type: none"> <li>• De student maakt Java applicaties met OOP-structuren als overerving, interfaces en abstracte klassen;</li> <li>• De student test geschreven klassen individueel met unittests;</li> <li>• De student past de principes Maven build lifecycle toe bij het beheren van externe code;</li> <li>• De student beheert zijn libraries met behulp van Maven.</li> </ul>
Omvang	8 EC, 6 bijeenkomsten

<b>Cursus</b>	<b>Backend Documentatie</b>
Leeruitkomst	De student stelt, op basis van de Software Development Life Cycle, technische documentatie op voor de backend van een applicatie, gebruik makend van UML-diagrammen.
Toelichting cursus	<p>Applicaties ontwikkelen is meer dan alleen code schrijven. Wanneer je direct in de code duikt, verlies je snel het overzicht van wat je aan het maken bent en waarom je het aan het maken bent.</p> <p>Daarom ga je in de eerste fase van de Software Development Life Cycle de eisen van de klant in kaart brengen. Deze requirements ga je vertalen naar acties die eindgebruikers met jouw applicatie willen uitvoeren. Vervolgens ontwerp je de structuur van je server zodat jouw applicatie al deze functionaliteiten kan ondersteunen.</p> <p>Tot slot ga je een ontwikkelomgeving opzetten waarin je de voortgang tijdens het ontwikkeltraject vastlegt. Nadat het technisch ontwerp en de ontwikkelomgeving zijn opgezet kan je starten met programmeren en zal je veel beter weten wat je moet programmeren.</p>
Prestatie indicatoren	<ul style="list-style-type: none"> <li>• De student legt de server-client communicatie vast in het sequence diagram.</li> <li>• De student structureert de ontworpen functionaliteiten in een klassendiagram.</li> <li>• De student levert, volgens de fases van de Software Development Life Cycle, de UML-diagrammen op de juiste momenten op.</li> <li>• De student beheert zijn code met gebruik van Git om met versiebeheer de voortgang van het project vast te leggen.</li> <li>• De student schrijft een goede installatiehandleiding zodat de applicatie door derden in een andere omgeving kan worden geïnstalleerd.</li> </ul>
Omvang	2 EC, 2 bijeenkomsten

<b>Cursus</b>	<i>Database Development</i>
Leeruitkomst	De student ontwerpt een relationele database, waarin data met onderlinge relaties veilig opgeslagen en uitgelezen kan worden, aan de hand van een technisch ontwerp document. Tevens beheert de student de data en rechten van databasegebruikers en voert hij CRUD-opdrachten uit op de database.
Toelichting cursus	Data die je later wilt uitlezen, aanpassen of hergebruiken slaan we op in databases. Bijvoorbeeld persoonlijke informatie als login- en adresgegevens en informatie die gedeeld wordt met anderen. Hierbij kun je denken aan berichten, foto's en informatie over het gedrag van gebruikers om statistieken mee te genereren. Deze drie soorten informatieopslag staan in Database Development centraal. We starten met maken van een database ontwerp en zetten vervolgens een database op met behulp van SQL-queries.
Prestatie indicatoren	<ul style="list-style-type: none"> <li>• De student vertaalt een Relationeel Representatie Model naar tabellen in de database.</li> <li>• De student zet, op basis van een tekstuele beschrijving een Relationeel Representatie Model op.</li> <li>• De student leest en bewerkt data (toevoegen, aanpassen, verwijderen) met behulp van SQL.</li> <li>• De student draagt zorg voor de autorisatie en authenticatie in de database.</li> </ul>
Omvang	5 EC, geen bijeenkomsten: zelfstudie

Cursus	Spring Boot
Leeruitkomst	De student zet een backend applicatie op met behulp van het Spring Boot framework en maakt gebruik van verschillende architecturale lagen binnen Spring. De student test zijn applicatie met unit-testen en het <i>mocken</i> van klassen en tevens communiceert de applicatie met een database.
Toelichting cursus	<p>Tijdens deze cursus leer jij een backend applicatie te programmeren die de spil is tussen de frontend en de database. Jouw applicatie zorgt ervoor dat alleen de juiste personen bij de juiste data kunnen komen en geeft vervolgens de gevraagde informatie terug.</p> <p>We gaan dit doen door tijdens deze cursus met een concrete opdracht aan de slag te gaan. Met behulp van het gegeven technisch ontwerp ga jij de backend schrijven waar klanten zich kunnen aanmelden voor een reparatie, waar de planner deze kan inplannen en de monteur de bon kan opmaken.</p>
Prestatie indicatoren	<ul style="list-style-type: none"> <li>• De student benoemt en beschrijft de verschillende architecturale lagen binnen het Spring framework;</li> <li>• De student schrijft code binnen de Spring Architectuur;</li> <li>• De student leest en bewerkt data (toevoegen, aanpassen, verwijderen) met behulp van SQL.</li> <li>• De student voert zijn testen uit met behulp van mocken;</li> <li>• De student gebruikt HTTP-methods om de vertaalslag te maken naar acties met de data;</li> <li>• De student past autorisatie en authenticatie toe binnen het Spring framework.</li> </ul>
Omvang	10 EC, 8 bijeenkomsten

<b>Cursus</b>	<i>Clean Code &amp; Design Patterns</i>
Leeruitkomst	De student schrijft zijn code volgens de afgesproken conventies van Clean Code en ontwikkelt highly cohesive en loose coupled code, door de toepassing van Design Patterns en SOLID.
Toelichting cursus	Applicaties zijn tegenwoordig zo groot dat je daar vrijwel nooit alleen aan werkt. Je werkt altijd samen met andere programmeurs en er komt een moment dat je jouw code moet overdragen aan een andere programmeur. Om deze samenwerking en overdracht goed te laten verlopen dient je code aan bepaalde standaarden en verwachtingen te voldoen. Gedurende deze cursus leren we je om te werken met gemaakte code-afspraken en Java conventies. Daarnaast leer je ook om de code te optimaliseren zodat deze makkelijker te onderhouden is, met behulp van Design Patterns.
Prestatie indicatoren	<ul style="list-style-type: none"> <li>• De student schrijft Java-code op basis van bestaande Java conventies (Clean Code).</li> <li>• De student past Design Patterns toe in zijn Java-code.</li> <li>• De student past de principes van SOLID toe in zijn Java-code.</li> </ul>
Omvang	5 EC, 2 bijeenkomsten

<b>Toetsing</b>
<i>Integrale eindopdracht.</i>