ivan ip

Fontys ICT Semester 3

Research document

Software development S3 Code Scanners

Table of Contents

[Inleiding 2](#_Toc155313272)

[Wat kunnen code scanners doen 3](#_Toc155313273)

[Hoe werken code scanners 4](#_Toc155313274)

[Hoe kan code scanners mijn applicatie verbeteren 5](#_Toc155313275)

# Inleiding

Voor de individuele project maak ik gebruik van Apex Legends API. Met de api kan ik de speler statistieken ophalen en ook de gespeelde matches terugkijken. De bedoeling hiervan is dat spelers gemakkelijk hun statistieken kunnen terug zien en ook de gespeelde matches. Er is namelijk ook geen mogelijkheid om je games terug te zien vandaar dat ik dit project maak. Ook zorg ik ervoor dat ik niet alleen de api zal gebruiken. De data die ik van de api ophaal zal ik continu opslaan in mijn eigen database zodat ik niet afhankelijk ben van de api. Natuurlijk moet mijn website goed presteren zodat mijn website aantrekkelijker is voor gebruikers. Door gebruik te maken van code scanners kan ik mijn site beter laten presteren. Maar wat zijn code scanners en hoe kunnen code scanners ervoor zorgen dat mijn website beter presteert? Daar ga ik het over hebben in dit onderzoek.

# Wat kunnen code scanners doen

Tijdens het ontwikkelingsproces van een applicatie kom je vaak verschillende soorten fouten tegen, zoals bugs, logische fouten, of zelfs beveiligingsproblemen. Sommige van deze fouten zijn gemakkelijk op te merken, maar andere kunnen door de ontwikkelaar over het hoofd worden gezien, vooral bij grote en complexe codebases. Code scanners spelen hierin een cruciale rol.

Een code scanner is een softwaretool die automatisch de code van een applicatie analyseert om fouten, bugs, beveiligingsproblemen, code-standaarden, en andere potentiële kwaliteitsproblemen op te sporen. Door het gebruik van een code scanner wordt het ontwikkelingsproces gestroomlijnd, omdat het een extra laag van validatie toevoegt aan de code, wat de kans op menselijke fouten vermindert.

Code scanners kunnen onder andere helpen bij het detecteren van:

* **Syntaxfouten**: Fouten die ontstaan doordat de code niet correct is geschreven volgens de regels van de programmeertaal.
* **Logische fouten**: Problemen die optreden wanneer de code wel syntactisch correct is, maar niet het gewenste resultaat oplevert.
* **Beveiligingskwetsbaarheden**: Zaken zoals SQL-injecties, cross-site scripting (XSS), of het opslaan van wachtwoorden in platte tekst.
* **Code duplicatie**: Herhalende stukken code die hergebruikt kunnen worden om onderhoud eenvoudiger te maken.
* **Niet-gedocumenteerde code**: Code die niet goed is gedocumenteerd, wat het moeilijker maakt voor andere ontwikkelaars om te begrijpen.
* **Performance-issues**: Coderegels die mogelijk de prestaties van de applicatie kunnen verminderen, zoals overbodige lussen of geheugenlekken.

Het gebruik van een code scanner maakt het ontwikkelen van software efficiënter en betrouwbaarder, doordat het ontwikkelaars helpt om sneller mogelijke problemen te identificeren en op te lossen.

# Hoe werken code scanners

Code scanners worden vaak geïntegreerd in de **CI/CD (Continuous Integration / Continuous Deployment)** pipeline, wat betekent dat ze automatisch draaien elke keer dat er een wijziging in de code wordt aangebracht. Dit biedt tal van voordelen, omdat het ontwikkelingsproces continu kan worden gecontroleerd zonder dat er handmatige interventie nodig is.

Een code scanner werkt doorgaans als volgt:

1. **Code wordt gepusht naar een Git-repository**: Elke keer dat een ontwikkelaar wijzigingen aanbrengt in de codebase, wordt deze geüpload naar een Git-repository (bijvoorbeeld GitHub of GitLab).
2. **De scanner wordt getriggerd**: Zodra de wijziging is gepusht, wordt de code scanner automatisch gestart via een CI/CD-pijplijn. Dit betekent dat het systeem de code die is gewijzigd of toegevoegd door de ontwikkelaar scant.
3. **Analyse van de code**: De scanner analyseert de code op basis van verschillende vooraf gedefinieerde regels, zoals programmeerstandaarden, best practices, en beveiligingsnormen.
4. **Rapportage van de resultaten**: Na de analyse genereert de code scanner een rapport waarin alle gevonden problemen worden opgesomd. Dit rapport kan zowel als een gedownload bestand worden aangeboden als een visuele weergave in de CI/CD-pijplijn zelf. In sommige gevallen wordt er ook een commentaar toegevoegd op specifieke regels van de code waar een probleem is gevonden.
5. **Terugkoppeling naar de ontwikkelaar**: Als er problemen worden gevonden, kan de ontwikkelaar direct worden gewaarschuwd. Vaak bevat de feedback niet alleen informatie over de fout, maar ook suggesties voor verbeteringen of verwijzingen naar relevante documentatie.

Dit proces is volledig geautomatiseerd en kan in real-time plaatsvinden, zodat je geen tijd verspilt met handmatige controle van je code. Je kunt het resultaat van de code scanner onmiddellijk gebruiken om de code te verbeteren en te optimaliseren.

# Hoe kan code scanners mijn applicatie verbeteren

Door het gebruik van een code scanner kan de kwaliteit van je applicatie op verschillende manieren verbeteren. Hieronder enkele belangrijke manieren waarop dit gebeurt:

1. **Snellere foutopsporing**: Een van de grootste voordelen van code scanners is dat ze ontwikkelaars helpen om fouten sneller te vinden. In plaats van door honderden of duizenden regels code te ploegen, krijgt de ontwikkelaar direct inzicht in waar zich de fouten bevinden. Dit maakt het eenvoudiger om snel de oorzaken van problemen te identificeren en op te lossen, wat het ontwikkelingsproces versnelt.
2. **Verhoogde codekwaliteit**: Code scanners helpen bij het naleven van bepaalde standaarden en best practices. Dit draagt bij aan een schonere, beter leesbare en onderhoudbare codebase. Bijvoorbeeld, de scanner kan detecteren als er onnodige complexe code of code duplicatie is, wat kan worden opgelost door de code te herstructureren of te refactoren.
3. **Verbeterde beveiliging**: Beveiliging is een van de belangrijkste zorgen in softwareontwikkeling. Code scanners kunnen kwetsbaarheden in de code opsporen, zoals onveilige manieren om gebruikersinvoer te verwerken, onvoldoende versleuteling van gevoelige gegevens, of verkeerde configuraties van beveiligingsinstellingen. Door deze kwetsbaarheden vroegtijdig te identificeren, kunnen ze worden verholpen voordat de applicatie live gaat, wat de algehele beveiliging van de applicatie verbetert.
4. **Consistency en samenwerking**: Als meerdere ontwikkelaars aan dezelfde codebase werken, kunnen er inconsistenties ontstaan in de stijl en structuur van de code. Een code scanner kan ervoor zorgen dat iedereen zich aan dezelfde codestandaarden houdt, zodat de code uniform blijft. Dit vergemakkelijkt de samenwerking tussen ontwikkelaars, omdat het gemakkelijker is om de code te begrijpen en aan te passen, ongeacht wie de oorspronkelijke auteur is.
5. **Preventie van regressie**: Regressie houdt in dat nieuwe code ervoor zorgt dat bestaande functionaliteit breekt. Code scanners kunnen helpen voorkomen dat regressies optreden door automatisch te controleren of er wijzigingen zijn die bestaande code kunnen verstoren. Dit verhoogt de stabiliteit van de applicatie en voorkomt dat nieuwe functies oude functies kapotmaken.
6. **Optimalisatie van de prestaties**: Code scanners kunnen ook prestaties verbeteren door code te identificeren die inefficiënt is, zoals onnodige lussen, zware berekeningen, of geheugenlekken. Deze performanceproblemen kunnen snel worden opgelost door de aanbevelingen van de scanner op te volgen, wat uiteindelijk leidt tot een snellere en stabielere applicatie.
7. **Kostenbesparing**: Door sneller bugs te identificeren en op te lossen, kunnen ontwikkelaars tijd besparen die anders verloren zou gaan aan handmatige tests en foutopsporing. Dit kan ook de onderhoudskosten verlagen doordat de codebasis schoner en beter gestructureerd is, waardoor toekomstige wijzigingen eenvoudiger door te voeren zijn.
8. **Verbeterde gebruikerservaring**: Ten slotte heeft het gebruik van code scanners indirect invloed op de gebruikerservaring. Aangezien bugs sneller worden opgelost en de applicatie veiliger en stabieler wordt, kan de eindgebruiker een soepelere en veiligere ervaring hebben met de applicatie.

# Conclusie

Code scanners zijn een onmisbare tool voor moderne softwareontwikkeling. Ze verbeteren niet alleen de kwaliteit van de code door fouten, bugs en beveiligingsproblemen vroegtijdig te detecteren, maar ze dragen ook bij aan een efficiënter ontwikkelingsproces. Het gebruik van code scanners in je CI/CD-pipeline biedt ontwikkelaars de mogelijkheid om sneller en met meer vertrouwen te werken, waardoor applicaties sneller, veiliger en beter worden. Door een code scanner te integreren in je ontwikkelworkflow kun je ervoor zorgen dat je applicatie zowel kwalitatief als functioneel op het hoogste niveau blijft.

# Bronnen

<https://www.privado.ai/glossary/code-scanning#:~:text=Code%20scanning%20tools%20are%20typically,before%20it%20is%20pushed%20live>.

<https://www.fortinet.com/resources/cyberglossary/code-scanning>

<https://www.checkpoint.com/cyber-hub/cloud-security/what-is-code-scanning/>

<https://snyk.io/learn/code-review/code-scanning/>