

Release Plan Groep 5

Teun Willems, Jules Kreutzer

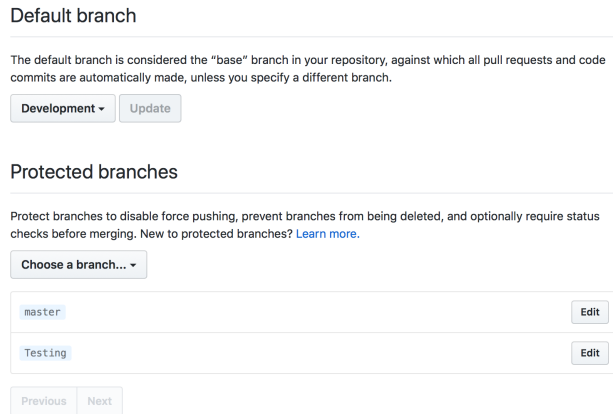
21 februari 2018

Inhoudsopgave

1	Version Control	2
1.1	Branches	2
1.1.1	Master	2
1.1.2	Testing	2
1.1.3	Development	3
1.1.4	Features	3
2	Code Style	4
3	Continuous Integration	5
4	Testing	6
5	Versioning	7
5.1	Major	7
5.2	Minor	7
5.3	Patch	7
5.4	Overige Informatie	7
6	Continuous Delivery	8

1 Version Control

Door middel van version control kunnen meerdere ontwikkelaars aan dezelfde software werken. Tijdens JEA6 zal Git (GitHub) gebruikt worden voor version control. Er is hiervoor gekozen omdat hiermee reeds de meeste ervaring is opgedaan.



The screenshot shows the GitHub repository settings for a branch. The 'Default branch' section indicates that 'Development' is the default branch, with an 'Update' button. The 'Protected branches' section explains that protected branches prevent force pushing and deletion. It lists 'master' and 'Testing' as protected branches, each with an 'Edit' button. Navigation buttons for 'Previous' and 'Next' are at the bottom.

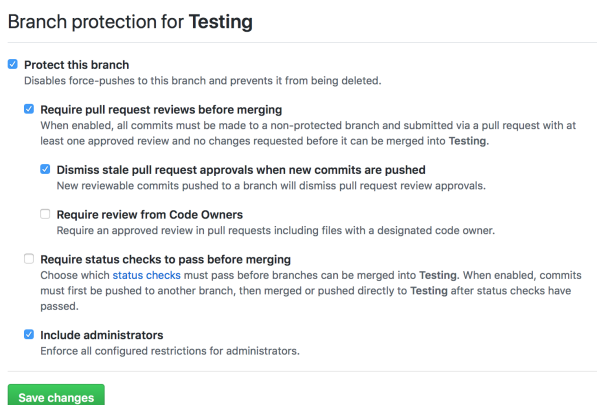
Figuur 1.1: Overzicht van default branch en protected branches

1.1 Branches

Voor de repository waar de code van het vak JEA6 wordt bijgehouden, zijn verschillende branches aangemaakt. Wat het nu is van deze verschillende branches, staat in de volgende alinea's uitgelegd.

1.1.1 Master

De code die beschikbaar is in de master-branch, is de code van de software versie die in productie gebruikt wordt. Nieuwe functies of bug fixes mogen dan ook niet rechtstreeks naar deze branch gecommitt worden. Deze branch is tevens protected. Dit wilt zeggen dat vanuit een git programma (UI of CLI) niet rechtstreeks naar deze branch gecommitt kan worden. Dit zorgt ervoor dat een ontwikkelaar niet per ongeluk naar deze branch kan committen.



The screenshot shows the 'Branch protection for Testing' settings. It includes several checkboxes: 'Protect this branch' (checked), 'Require pull request reviews before merging' (checked), 'Dismiss stale pull request approvals when new commits are pushed' (checked), 'Require review from Code Owners' (unchecked), 'Require status checks to pass before merging' (unchecked), and 'Include administrators' (checked). A 'Save changes' button is at the bottom.

Figuur 1.2: Overzicht van protectie wat ingesteld is op branch

1.1.2 Testing

De Testing-branch zal gebruikt worden om nieuwe functionaliteit die ontwikkeld zijn in de development branch te testen. Wanneer de functionaliteit in de testing branch geaccepteerd wordt, zal deze worden doorgezet naar de master branch zodat de verbeteringen en/of nieuwe wijzigingen beschikbaar zijn voor alle gebruikers op de productie omgeving De Testing-branch heeft dezelfde setup als

de Master-branch. Dit wilt zeggen dat het op deze branch ook niet mogelijk is om rechtstreeks code te committen.

1.1.3 Development

De Development-branch bevat nieuwe functionaliteit of verbeteringen wat ontwikkeld zijn. Deze branch is vrij toegankelijk voor de ontwikkelaar om eventueel kleine(re) verbeteringen door te voeren. Het is als nog de bedoeling dat nieuwe functionaliteit niet rechtstreeks op deze branch wordt gecommitt. Mocht een ontwikkelaar nieuwe functionaliteit of grotere verbeteringen willen ontwikkelen, kan hiervoor een zogenaamde Feature-branch worden aangemaakt. Meer informatie hierover is in de volgende alinea te lezen

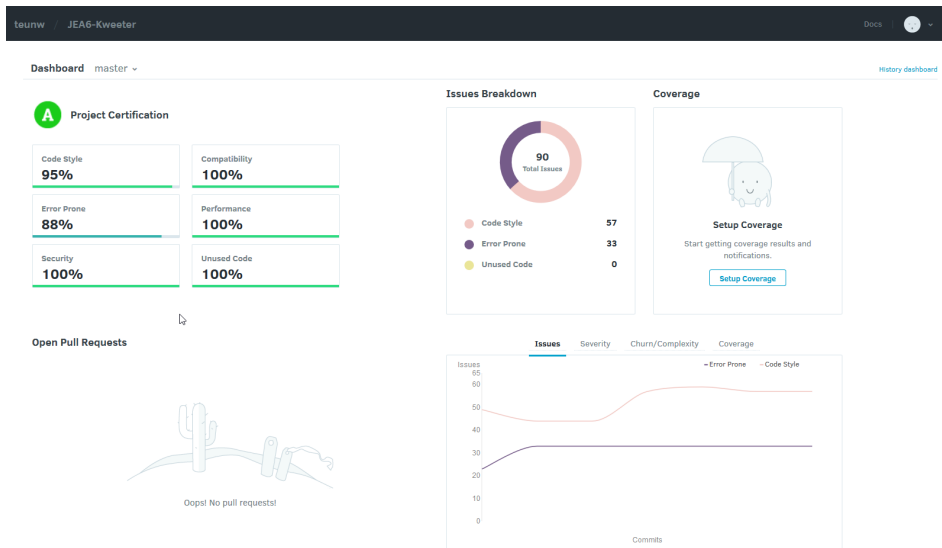
1.1.4 Features

De Features-branch is geen branch die direct aangemaakt is op Github. Dit komt doordat "Features" een overkoepelende term is. Wanneer een ontwikkelaar nieuwe functionaliteit of verbeteringen wilt ontwikkelen, dan zal hiervoor een nieuwe branch moeten worden aangemaakt, bijvoorbeeld "Dev-exampleFeature". Aan de hand van de prefix "Dev-" weet iedereen dat dit een branch betreft waar nieuwe functionaliteit of verbeteringen op worden ontwikkeld. Het gedeelte achter de prefix moet dan ook beschrijven aan welke functionaliteit of verbeteringen worden gewerkt.

Wanneer de ontwikkelaar het implementeren heeft voltooid, zal dit eerst getest moeten worden op zijn eigen, lokale omgeving. Wanneer die nieuwe functionaliteit werkt met de rest van de ontwikkelde software, kan de ontwikkelaar zijn branch, bijvoorbeeld "Dev-exampleFeature" gaan samenvoegen met de Development-branch. Hiervoor maakt de ontwikkelaar een merge request aan wat ervoor zorgt dat diene nieuwe code wordt samengevoegd op de Development-branch.

2 Code Style

Voor de code style worden de standaard code style settings van IntelliJ IDEA gebruikt. De code style wordt ook gecontroleerd door Codacy, een code analyse tool. Codacy controleert je Java (en Javascript) codebase op verschillende errors.

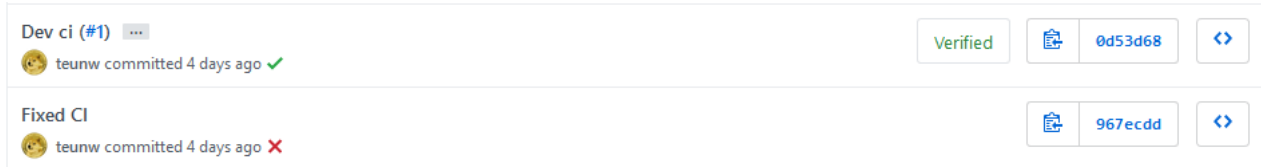


Figuur 2.1: Voorbeeld van code analysis door middel van Codacy

Ook kan er gecontroleerd worden op beveiligingsproblemen in je applicatie, stel dat je bijvoorbeeld SQL-query parameters in je queries hebt staan.

3 Continuous Integration

Continuous Integration is geregeld door TravisCI, voor open source projecten is TravisCI gratis te gebruiken. Omdat het project met Gradle is gebouwd is het makkelijk om CI ondersteuning toe te voegen aan de backend. Het commando "gradle test" draait alle tests in het project, als er iets fout gaat is dit ook meteen in Github zichtbaar.



Figuur 3.1: Overzicht TravisCI

4 Testing

5 Versioning

Door gebruik te maken van versioning kan het voor een ontwikkelaar duidelijk zijn in welke versie van de software een eventueel probleem is aangetroffen. Een manier om versioning toe te passen is door gebruik te maken van Semantic Versioning (versie 2.0.0). Meer informatie en de volledige specificatie van Semantic Versioning is beschikbaar op hun website: <https://semver.org>

Bij Semantic Versioning is een versienummer opgebouwd uit drie verschillende delen. Namelijk:

- Major
- Minor
- Patch

5.1 Major

Het eerste getal van het versie nummer geeft de versie van de software aan. Dit getal begint meestal bij 1, maar in sommige gevallen kan ook versie 0 voorkomen. In deze gevallen geeft de 0 aan dat de software nog in ontwikkeling is, en dat de software nog niet als "stable" gezien kan worden. Wanneer het major nummer van de versie verandert, geeft dit meestal aan dat er wijzigingen in de API zijn gedaan die niet meer compatible zijn met een vorige versie. Zo kan de data die naar een endpoint gestuurd wordt bijvoorbeeld op een andere manier worden terug gegeven.

5.2 Minor

Het tweede getal in het versienummer geeft aan welke wijziging zijn toegevoegd aan het huidige software pakket. Wijzigingen die ervoor zorgen dat het minor getal toeneemt, is meestal nieuwe functionaliteit wat beschikbaar wordt gesteld. Een vereiste van deze functionaliteit is dat deze compatible is met de huidige implementatie van de software. Wanneer een of meerdere functie's niet meer ondersteund worden, moet dit volgens de Semantic Versioning specificatie ook worden weergegeven in het minor nummer van de versie.

5.3 Patch

Wanneer in de huidige versie van de software een fout wordt ontdek en dit verholpen moet worden, een zogenaamde "bugfix". Deze bugfix kan ook worden weergegeven in het versienummer van de software. Dit gebeurt aan de hand van het derde nummer: het patch nummer. Het is vanzelfsprekend dat de huidige functionaliteit niet veranderd mag worden. Mocht dit wel het geval zijn, dan is het ook vereist dat het major of minor nummer van de versie aangepast wordt.

5.4 Overige Informatie

Bij de opbouw van het versie nummer is het niet toegestaan om een nummer te laten beginnen met 0 (bv. 1.01.0 is niet toegestaan). Een uitzondering hierop is de opbouw van het versie nummer wanneer de software nog in ontwikkeling is zoals beschreven in *5.1 Major*.

Wanneer ervoor gekozen wordt om een pre-release versie van de software beschikbaar te stellen, kan dit tevens worden weergegeven in het versie nummer. Dit gebeurt door de versie van de pre-release door middel van een koppelteken te plaatsen achter het daadwerkelijke versie nummer van de software. Een aantal voorbeelden hiervan zijn: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7

Eventuele build metadata mag tevens worden weergegeven in het versie nummer. Dit wordt gebruikelijk direct na het patch nummer of de pre-release versie weergegeven met behulp van het koppelteken +.

6 Continuous Delivery