Im ersten Schritt wird ein Spring Boot Projekt in IntelliJ erstellt.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Jetzt erstelle ich ein Repo auf GitHub

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Das Repo und das Projekt in IntelliJ werden jetzt miteinander verknüpft und die ersten Sachen auch gleich committed.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift enthält.

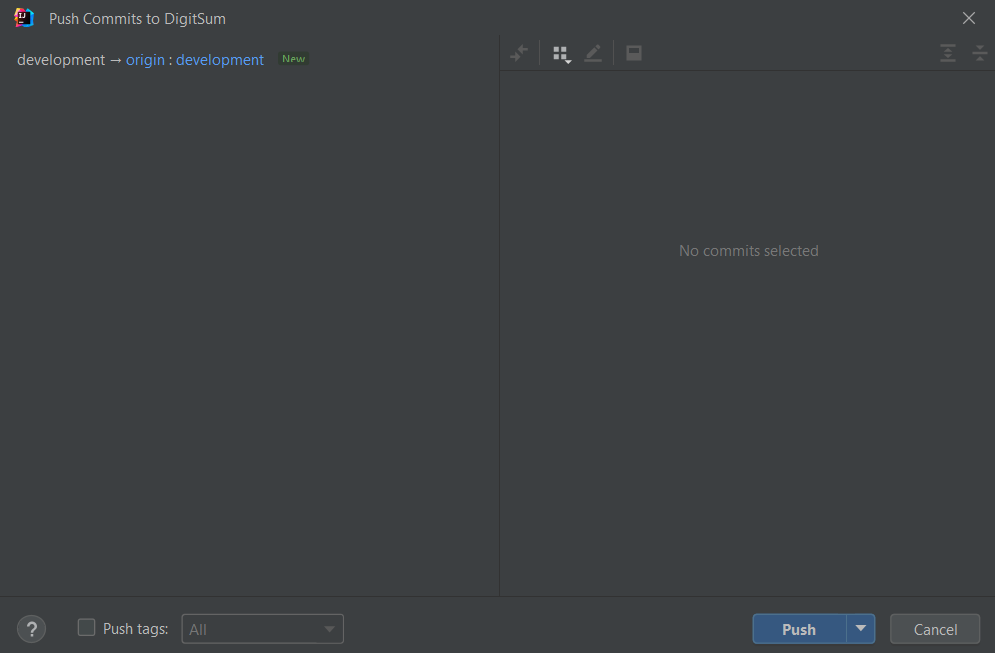
Automatisch generierte Beschreibung

Das Repo wurde erfolgreich erstellt und verknüpft. Der erste Commit ist auch zu sehen

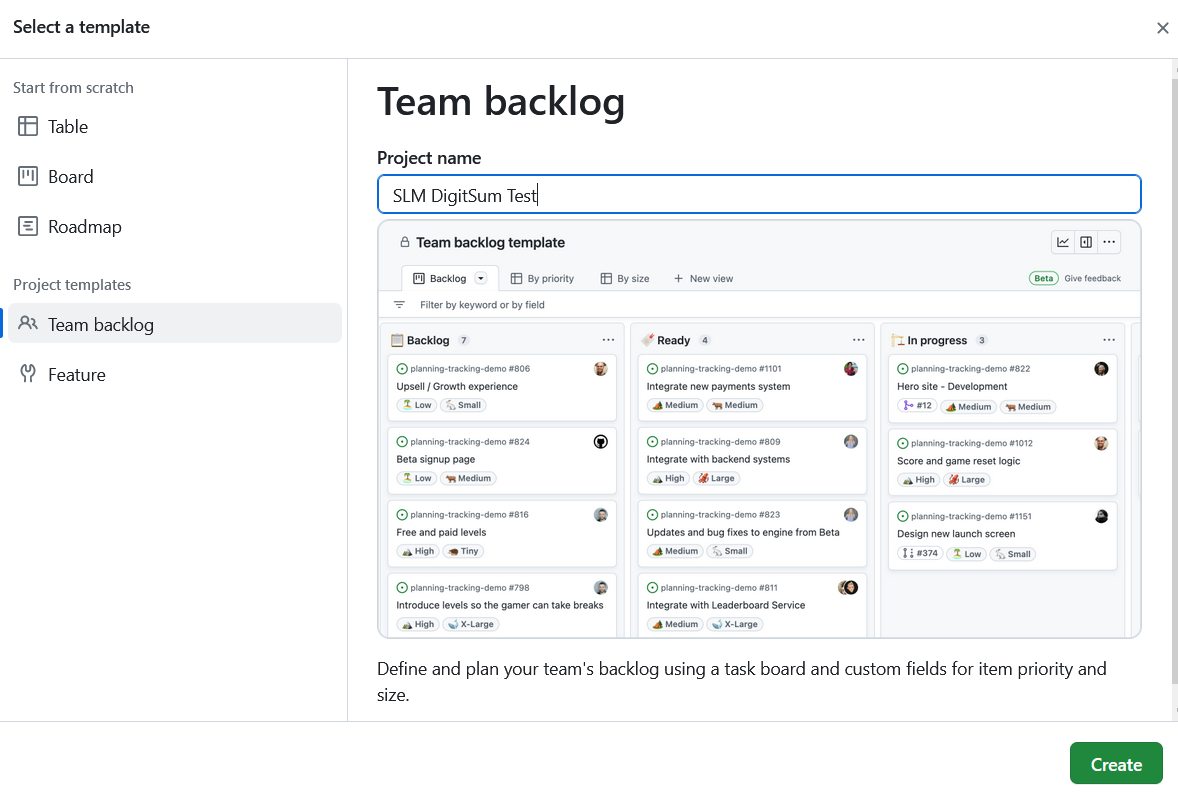
Ein Bild, das Text, Screenshot, Zahl, Software enthält.

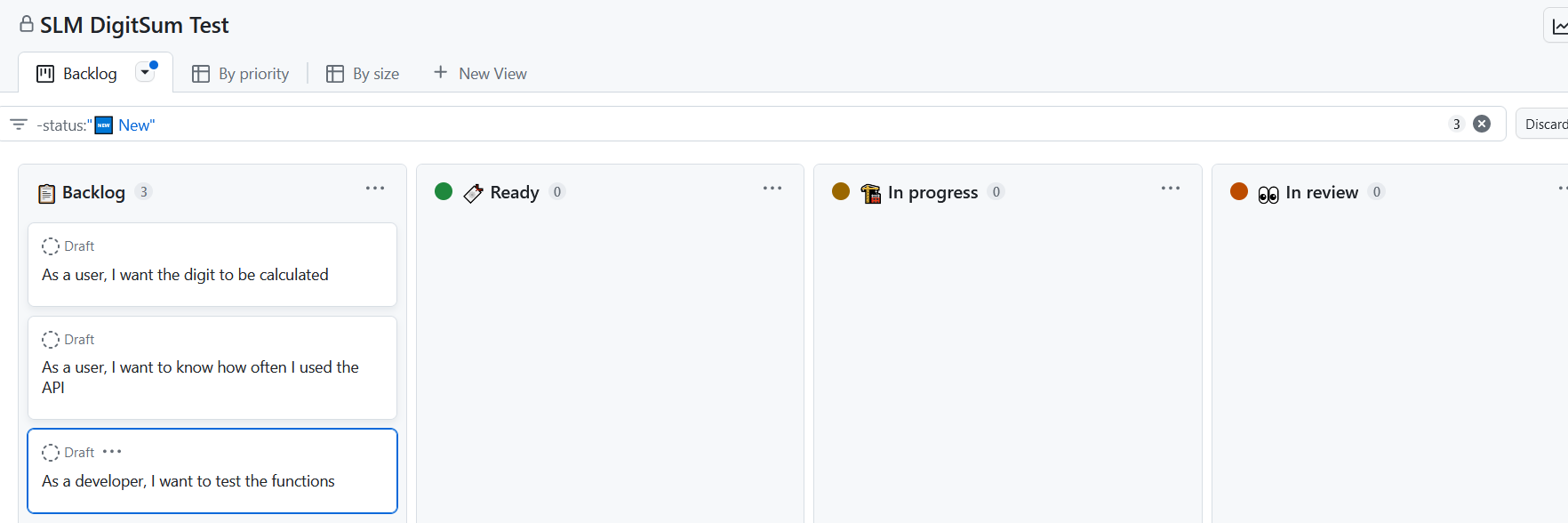
Automatisch generierte Beschreibung

Ein neuer Branch „development“, der vom Main Branch ausgeht, wurde erstellt (in IntelliJ: Git -> New Branch). Nachdem er erzeugt wurde, wurde er gepusht.



Auf GitHub gehe ich jetzt auf Projects, um ein neues Projekt zu erstellen, wo ich die User Stories (im Team) beschreiben kann.

Die User Stories wurden beschrieben…

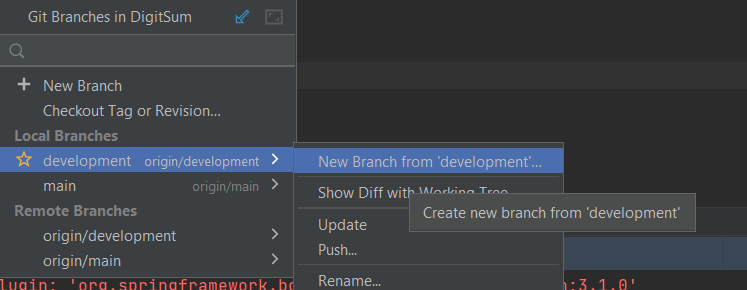


Da ich jetzt beginne an der ersten User Story zu arbeiten, verschiebe ich sie zu „in progress“ und die anderen zu „ready“.

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Ich erstelle einen neuen Branch „feature\_digitsum“, um die erste User Story zu implementieren.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

…und pushe das wieder

Ein Bild, das Screenshot, Text, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ich erstelle zum Implementieren eine neue Java Klasse „DigitSumController“

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

…und adde sie gleich ins Git

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

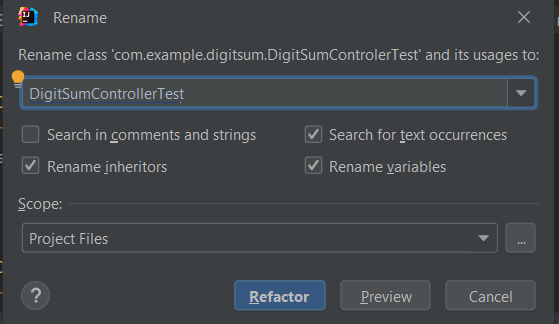
Automatisch generierte Beschreibung

Nachdem ich das erste Feature implementiert habe, schreibe ich die dazugehörigen Unit-Tests. Dafür erstelle ich erstmal eine Klasse und füge sie ins Git hin zu

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Da ein kleiner Tippfehler im Klassennamen ist, refactore ich den Namen



Code des Unit-Tests

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Die Unit-Tests waren erfolgreich

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Jetzt habe ich die neuen Changes committed und gepusht.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Der Push ist auch auf GitHub zu sehen

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Im nächsten Schritt wird ein Pull Request erzeugt und die User Story in „in Review“ gezogen.

Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Schrift, Zahl, Reihe enthält.

Automatisch generierte Beschreibung

Der Branch „feature\_digitsum“ wird mit dem „development“ gemerged.

Ein Bild, das Text, Reihe, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Die User Story ist abgeschlossen.

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Für das nächste Feature sollten wir wieder einen neuen Branch „feature\_countPageVisits“ erzeugen und eine Klasse zum Implementieren.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Aus Zeitgründen habe ich diese zwei Schritte aber wieder revertiert und es schnell in dasselbe Branch und dieselbe Klasse gemacht wie das erste Feature.



In Kalkulier-Methode

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Um eine Pipeline zu konfigurieren, gehen wir in GitHub auf „Actions“ und dort auf „Java with Maven“

Ein Bild, das Text, Schrift, Zahl, Screenshot enthält.

Automatisch generierte Beschreibung

Leider hat die Zeit gefehlt, um die Pipeline fertig zu konfigurieren. Was wir jedoch dabei beachten müssten, ist, dass wir sie für die richtigen Branches, als auch die richtige Java Version konfigurieren.

Wenn die Pipeline erfolgreich durchläuft, kann das Artefakt heruntergeladen werden.