

Level 1:

Visual Paradigm Online Free Edition

Level 2:



Die für Equaly gewählte Softwarearchitektur sollte Einfachheit und gleichzeitig Modularität und Wartbarkeit in der Software etablieren. Auf Basis dieser Anforderungen wurde sich für eine Strukturierung nach dem Architekturmuster MVC entschieden.

So ist eine Instanz des EqualyControllers für die Verarbeitung von Anfragen seitens einer Weboberfläche zuständig. Diese nimmt die Rolle der View ein und ist für den Nutzer der einzige Interaktionspunkt mit Equaly. Aus der View werden Daten zu einem zu gendernden Text entgegengenommen. Neben dem Text selbst kann das zusätzlich auch die Spezifizierung des Gendering-Stils sein.

Auf Knopfdruck im HTML-Frontend werden die Daten mittels Thymeleaf und JavaScript an den EqualyController im Java-Backend übermittelt. Von dort werden sie an einen Language-Tagger zur Sprachidentifikation weitergeleitet. Dieser ist mit dem Framework Lingua realisiert. Basierend auf der identifizierten Sprache findet dann durch einen Tokenizer die Auftrennung des Textes in Wörter statt. Die Token werden einem POS-Tagger (Parts of Speech Tagger) übergeben, der u.A. Wortarten und Satzzeichen identifiziert und somit für Equaly interpretierbar macht und mittels dem Framework OpenNLP realisiert wird. Der EqualyController erhält diesen Token-Datensatz und reicht ihn weiter an einen Substituter. Hier werden die identifizierten Substantive und darauffolgend die Artikel und Präpositionen herausgelöst und an eine Datenbank weitergeschickt. Die Interaktion mit der Datenbank findet dabei über einen DatabaseController statt. Er ist die Schnittstelle zur SQLite-Datenbank, die wiederum das Model darstellt. Aus der Datenbank erhaltene Substitute werden dann durch den Substituter eingesetzt und vermerkt. Der gegenderte Text wird an den EqualyController weitergeleitet und durch ihn mittels Thymeleaf im Frontend visualisiert.

Wortanzahl: 246