

Features

Das Backend basiert auf REST. Wie oben erwähnt, nutzt es das Konzept von Entitäten und Links, wobei die Entitäten als Knoten und die Links als gerichtete Kanten zu sehen sind. Jede Entität ist von einem bestimmten Typ, der vom Entwickler festgelegt werden kann. Eine Entität kann beliebig viele Properties in Form von Key-Value-Paaren besitzen, ebenso verhält sich das Ganze mit Links. Ein Link zeigt immer auf eine andere Entität und ist kein eigenständiges Objekt, sondern immer fest mit seinem Subjekt- und Objekt-Knoten verbunden. RiP kann mit den Datentypen long, double, boolean und String umgehen. Zusätzlich dazu bieten wir noch die Möglichkeiten, Dates und Locations zu speichern.

Da im Bereich der App-Entwicklung Entitäten mit Eigenschaften nicht ausreichen, sondern oft noch Dinge wie Profilbilder etc. benötigt werden, bietet RiP ebenfalls die Möglichkeit, Bilder abzuspeichern. Diese können dann wieder vom Backend angefragt werden. Gibt man bei der Anfrage zusätzliche Parameter an, ermöglicht dies Skalierungen, Hintergrundfarben, Transparenz etc. anzuwenden.

Da wir mit einem starren Backend nicht alle möglichen Varianten abdecken können, gibt es eine Möglichkeit, über eine Administrationsoberfläche Hooks in den gewöhnlichen Ablauf im Backend einzubetten. Diese werden über Groovy-Skripte ermöglicht. Es gibt eine Vielzahl von Hooks im Backend, für die sich solche Skripte hochladen lassen, um den starren Ablauf von RiP zu modifizieren, falls Funktionen wie Email-Benachrichtigungen etc. erweitert werden sollen.

Weitere Kapitel der Dokumentation

In den folgenden Kapiteln werden wir einen tieferen Einblick in unser Backend gewähren. Zunächst werden wir dabei im [zweiten Kapitel](#) auf die REST-Schnittstelle mit all ihren Endpunkten eingehen. Es gibt Endpunkte für [Entitäten](#), [Links](#) und [Bilder](#), sowie für die Administrationsoberfläche – also Accounts, Apps, ApiKeys, Groovy-Skripte. Im vierten Kapitel gehen wir noch eine Abstraktionsebene tiefer und geben Einblicke in unsere Implementierung, damit verständlich wird, wie RiP aufgebaut ist. Hierbei soll auch gezeigt werden, wie die JSON-Objekte der http-Requests in Entitäten umgewandelt, die Bilder sowie die Integration der Hooks gehandhabt werden. Abschließend wird in diesem Kapitel dann auch die Implementierung der Administrationsoberfläche erklärt. Im letzten Kapitel wollen wir dann noch einen beispielhaften Ablauf einer App-Entwicklung durchführen. Dort werden wir zunächst im administrativen Teil des Backends einen neuen Account anlegen. Anschließend dann eine konkrete Anwendung unseres Backends vorführen.

Im Anhang finden sich dann noch zusätzliche Dokumentationen zu den möglichen [Fehlercodes](#) des Backends, den Ablauf der Autorisierung und des Pagings.