

### 1 Einführung

Ressourcen finden wir in allen Bereichen unseres Lebens wieder.

In der Informatik wird etwas als Ressource definiert, wenn es „identifizierbar“ ist und über mindestens eine Repräsentation angesprochen werden kann.

Umgangssprachlich gesagt, wird etwas zu einer Ressource, sobald wir oder jemand anderer es „anspricht“.

### 2 Hauptteil

Ressourcen stellen in REST ein zentrales Konzept dar.

Es ist wichtig zu erwähnen, dass man nie direkt mit Ressourcen arbeiten kann, sondern nur mit deren Repräsentanten, von denen es für eine Ressource mehrere geben kann.

Unterschiedliche Betrachtungsweisen führen zu unterschiedlichen Repräsentationen. Aus diesem Grund ist es sinnvoll über ein generelles „Darstellungsdesign“ für Ressourcen nachzudenken.

Stefan Tilkow identifiziert in seinem Buch „REST und http“ neun verschiedene Designarten von Ressourcen, die für den Webbereich relevant sind.

#### Primärressourcen

Primärressourcen sind Ressourcen, die man als „Individuen“ identifizieren und immer referenzieren kann. Granularität, Implementation usw. spielen dabei keine Rolle.

#### Subressourcen

Subressourcen sind Teilressourcen von anderen Ressourcen.

#### Listenressourcen

Listenressourcen sind eine (vollständige) Menge

von Ressourcen.

#### Filterressourcen

Filterressourcen sind eine Menge von Ressourcen, die aus einer Liste stammen, auf die mindestens ein Filterkriterium angewendet wurde.

#### Pagierung

Eine Pagierung beschränkt die Anzahl der Ressourcen, die auf einmal angezeigt werden.

#### Projektion

Bei einer Projektion werden nur bestimmte Eigenschaften von Ressourcen angezeigt.

#### Aggregationen

Aggregationen sind Mengen von Projektionen verschiedener Ressourcen.

#### Aktivitäten

Aktivitäten sind Ressourcen, die aus Methoden und Aktionen entstehen.

#### Konzeptressourcen

Konzeptressourcen sind Ressourcen, die selbst nicht referenzierbar sind, sondern nur ein Konzept darstellen.

Manche Ressourcen tauchen in vielen Kategorien auf, andere scheinen dort wo sie auftauchen vollkommen unpassend zu sein. Es kommt letzten Endes immer auf die Perspektive an. Korrekterweise sollte man immer von Repräsentationen sprechen und sich vor Beginn eines Projektes auf eine gemeinsame Sicht der Dinge einigen.

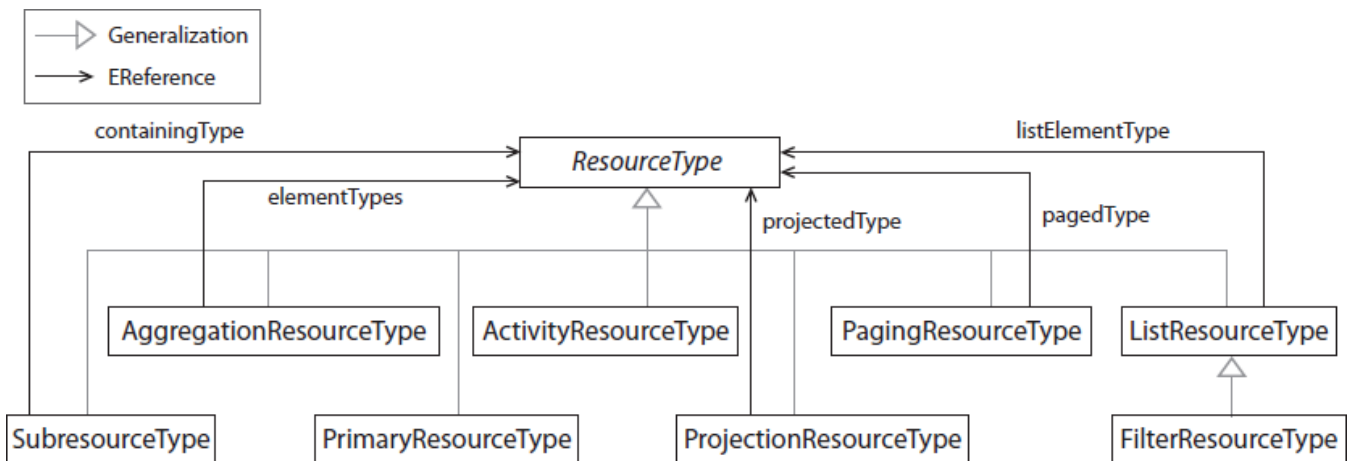


Abbildung 1: Hierarchien und Relationen von Ressourcentypen, Silvia Schreier

## URI

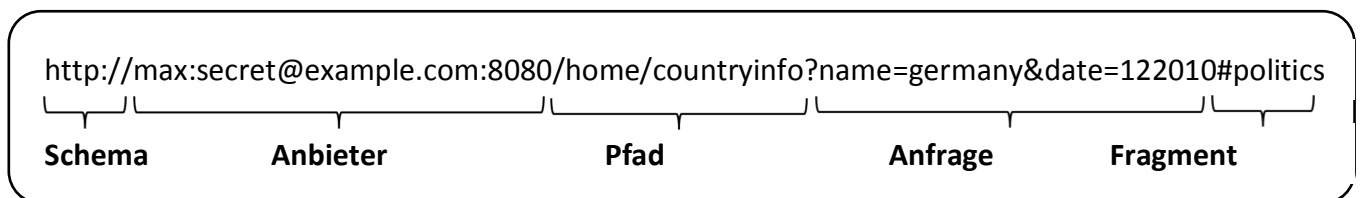


Abbildung 2: Beispielhafter Aufbau einer URI, nach RFC 3986

Uniform Ressourcen Identifier (URI) sind Identifikatoren einzelner Ressourcen. Eine URI ist (sollte) weltweit einmalig sein und immer nur auf eine Ressource verweisen.

URIs bestehen generell immer aus einem Schema und einem Pfad. Das jeweilige Schema kann jedoch noch weitere Angaben fordern.

### Schema

Das Schema (z.B. http, ftp) gibt an, wie die weiteren Angaben zu interpretieren sind.

### Anbieter

Die Angabe des Anbieters dient meist zu Festlegung von Namensräumen (z.B. DNS) und kann auch eine Portangabe sowie Informationen zur Autorisierung beinhalten.

### Pfad

Der Pfad ist in der Regel eine hierarchisch Angabe, welche zur Identifizierung einer Ressource innerhalb des Schemas und Anbieters

### Anfrage

Die Anfrage ist eine nicht hierarchische Angabe und unterstützt den Pfad in der Identifizierung einer Ressource.

### Fragment

Das Fragment erlaubt die indirekte Identifikation einer weiteren (Sub-)Ressource, innerhalb der durch die bisherigen Angaben identifizierten (Primär-)Ressource.

### Literatur:

- [1] Stefan Tilkov, „REST und HTTP“, 2. Auflage, Heidelberg, Dpunkt Verlag, 2011
- [2] "Uniform Resource Identifier", Online; Stand 23. Oktober 2011 [http://de.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://de.wikipedia.org/wiki/Uniform_Resource_Identifier)
- [3] Silvia Schreier, „Modeling RESTful applications“, WS-REST '11, New York, ACM, 2011
- [4] RFC 3986, „Uniform Resource Identifier (URI): Generic Syntax“, Online; Stand 23. Oktober 2011 <http://tools.ietf.org/html/rfc3986>