REST Specfications
Projekt BierIdee

Danilo Bargen, Christian Fässler, Jonas Furrer

5. April 2012

Inhaltsverzeichnis

1 Einleitung **4**

2 REST Ressourcen **4**

2.1 Beer 5
2.2 Beers 5
2.3 Users 5
2.4 Recommendations 6
2.5 Ratings 6
2.6 Consumption 7
2.7 Consumptions 7
2.8 Brewery 7
2.9 Breweries 8
2.10 Timeline 8
2.11 Tag 9
2.12 Tags 9

Änderungshistorie

Version	**Datum**	**Änderung**	**Person**
v1.0	03.04.2012	Dokument erstellt	dbargen
v1.1	05.04.2012	Tags und Json Formate hinzugefügt	jfurrer

1 Einleitung

Die Definition der Ressourcen orientiert sich an den Regeln des Buches *REST API Design Rulebook* [Mas11] aus dem O'Reilly Verlag.

URI Definition

Bei der Bezeichnung der URIs[1] wurde folgende Terminologie gemäss RFC 3986 verwendet:

```
URI = scheme "://" authority "/" path [ "?" query ] [ "#" fragment ]
```

Ressource-Archetypen

Nachfolgend die Ressource-Archetypen gemäss [Mas11]. Die Erklärungstexte wurden direkt dem besagten Buch entnommen.

Document A document resource is a singular concept that is akin to an object instance or database record. A document's state representation typically includes both fields with values and links to other related resources.

Collection A collection resource is a server-managed directory of resources. Clients may propose new resources to be added to a collection. However, it is up to the collection to choose to create a new resource, or not.

Store A store is a client-managed resource repository. A store resource lets an API client put resources in, get them back out, and decide when to delete them. On their own, stores do not create new resources; therefore a store never generates new URIs. Instead, each stored resource has a URI that was chosen by a client when it was initially put into the store.

Controller A controller resource models a procedural concept. Controller resources are like executable functions, with parameters and return values; inputs and outputs. Like a traditional web application's use of HTML forms, a REST API relies on controller resources to perform application-specific actions that cannot be logically mapped to one of the standard methods (create, retrieve, update, and delete, also known as CRUD).

2 REST Ressourcen

Nachfolgend sind die verfügbaren REST Ressourcen definiert. Alle Ressourcen sind unter der URI Authority `http://brauerei.nusszipfel.com/` erreichbar.

[1]Uniform Resource Identifier

2.1 Beer

Ein spezifisches Bier, identifiziert durch die ID.

URI Path `/beers/{beer-id}`

Archetype Document

Methods GET, PUT, DELETE

Json Format

```
{
        type: "beer",
        name: "{beer-name}",
        image: "{image-path}",
        brand: "{brand-name}",
        beertype: "{resource-URI}",
        tags: "{resource-URI}?beer={beer-id}"
}
```

2.2 Beers

Der Bestand aller Biere.

URI Path `/beers`

Query Parameters `tag={tag-name}`

Archetype Collection

Methods GET, POST

Json Format

```
[{
        type: "beer",
        name: "{beer-name}",
        image: "{image-path}",
        resource: "{resource-URI}"
}]
```

2.3 Users

Ein Benutzer, identifiziert durch den Benutzernamen.

URI Path `/users/{username}`

Archetype Store

Methods GET, PUT, DELETE

Json Format

```
{
        type: "user",
        username: "{username}"
}
```

2.4 Recommendations

Bier-Empfehlungen für einen bestimmten Benutzer.

URI Path `/users/{username}/recommendations`

Archetype Controller

Methods GET

Json Format

```
[{
        "type": "recommendation",
        "name": "{beer-name}",
        "beer": "{resource-URI}"
}]
```

2.5 Ratings

Eine Bier-Bewertung durch einen bestimmten Benutzer.

URI Path `/beers/{beer-id}/ratings/{username}`

Archetype Store

Methods GET, PUT, DELETE

Json Format

```
{
        type: "rating",
        beer: "{resource-URI}",
        user: "{resource-URI)",
        value: {value}
}
```

2.6 Consumption

Ein Bierkonsum, identifiziert durch die ID.

URI Path `/consumption/{consumption-id}`

Archetype Document

Methods GET, PUT, DELETE

Json Format

```
{
        type: "consumtion",
        beer: "{resource-URI}",
        user: "{resource-URI)",
        timestamp: {value}

}
```

2.7 Consumptions

Der Bestand aller Bierkonsume.

URI Path `/consumption`

Query Parameters `user={username}, beer={beer-id}`

Archetype Collection

Methods GET, POST

Json Format

```
[{
        type: "consumtion",
        resource: "{recource-URI}"
        timestamp: {value}
}]
```

2.8 Brewery

Eine Brauerei, identifiziert durch die ID.

URI Path `/breweries/{brewery-id}`

Archetype Document

Methods GET, PUT, DELETE

Json Format

```
{
        type: "brewery",
        name: "{brewery-name}"
        size: "{value}",
        profile: "{value}"
}
```

2.9 Breweries

Der Bestand aller Brauereien.

URI Path `/breweries`

Query Parameters `brewerySize={size}`

Archetype Collection

Methods GET, POST

Json Format

```
[{
        type: "brewery",
        name: "{brewery-name}"
        resource: "{resource-URI}"
}]
```

2.10 Timeline

Die Aktivitäts-Timeline.

URI Path `/timeline`

Query Parameters `pageSize={size}, pageStartIndex={index}, user={username}`

Archetype Collection

Methods GET

Json Format

```
[{
        type: "{consumition | rating}",
        name: "{beer-name}",
        user: "{user-name}",
        resource: "{resource-URI}"
}]
```

2.11 Tag

Ein spezifischer Tag, identifiziert duch die ID.

URI Path `/tags/tag-id`

Archetype Document

Methods GET, POST

Json Format

```
{
        type: "tag",
        name: "{tag-name}"
}
```

2.12 Tags

Liste aller Tags.

URI Path `/tags`

Query Parameters `beer={beer-id},`

Archetype Collection

Methods GET

Json Format

```
[{
        type: "tag",
        name: "{tag-name}",
        resource: "{resource-URI}"
}]
```

Literatur

[Mas11] M. Masse. *REST API Design Rulebook*. O'Reilly Media, 2011.