Entwurf REST-Webservice

# Einleitung

Interaktion mit dem Webservice erfolgt nur per JSON!

# Ressourcen

Der Webservice soll folgende Ressourcen repräsentieren.

Für den Bau der URI ist es wichtig, sich auf eine Bezeichnung der Ressource zu einigen (Singular / Plural). Abstimmung!

Eine einheitliche URI ist ein Maß für die Qualität des Rest-Services!

|  |  |
| --- | --- |
| Singular | Plural |
| Freundesliste | Freundeslisten |
| Freundschaftsanfrage | Freundschaftsanfragen |
| Benutzer | Benutzer |
| Sitzung | Sitzungen |
| Veranstaltung | Veranstaltungen |
| Raum | Räume |
| Tag | Tags |
| Konfiguration | Konfigurationen |

TODO: Einigen, welche Bezeichnung wir nehmen!

# Was muss der REST-Service potenziell bereitstellen können?

Der REST-Service muss potenziell alle Instanzen von Ressourcen und Listen dieser bereitstellen können

# Interaktion mit Ressourcen

|  |  |  |  |
| --- | --- | --- | --- |
| HTTP-Methode | CRUD-Operation | Beschreibung | Idem potent |
| POST | CREATE |  | (NEIN) |
| GET | READ |  | JA |
| PUT | UPDATE |  | JA |
| DELETE | DELETE |  | JA |

# Entwurf der URIs

|  |  |  |
| --- | --- | --- |
| URI | HTTP-Methode | Ergebnis |
| /freundesliste | GET | Freundesliste |
|  |  |  |
| /freundschaftsanfrage | GET | Liste Freundschaftsanfragen |
| /freundschaftsanfrage | POST | Neue Freundschaftsanfrage |
| /freundschaftsanfrage/{id} | GET | Freundschaftsanfrage |
| /freundschaftsanfrage/{id} | PUT | Freundschaftsanfrage ändern |
| /freundschaftsanfrage/{id} | DELETE | Freundschaftsanfrage löschen |
|  |  |  |
| /benutzer | GET | Liste von Benutzern |
|  |  |  |
| /sitzung | GET | Aktive Sitzung des Fragenden |
| /sitzung | POST | Neue Sitzung |
| /sitzung/{id} | PUT | Sitzung ändern |
| /sitzung/{id} | DELETE | Sitzung löschen |
|  |  |  |
| /veranstaltung | GET | Liste der Veranstaltungen |
| /veranstaltung | POST | Neue Veranstaltung |
| /veranstaltung/{id} | GET | Veranstaltungsdetails |
| /veranstaltung/{id} | PUT | Veranstaltung ändern |
| /veranstaltung/{id} | DELETE | Veranstaltung löschen |
|  |  |  |
| /raum | GET | Liste der Räume |
| /raum/{id} | GET | Raumdetails |
| /raum/{id} | PUT | TAG SETZEN ???? |
|  |  |  |
| /tag | GET | Liste der Tags |
| /tag | POST | Neuer Tag |
| /tag/{id} | DELETE | Tag löschen |
|  |  |  |
| /konfiguration | GET |  |
| /konfiguration | PUT | Ändern der Konfiguration |

# Spezifikation der Antworten im OK Fall:

# Spezifikation der Antworten im Fehler Fall:

1. {
2. "id":"54887487875584",
3. "id":"54887487875584",
4. "id":"54887487875584",
5. "id":"54887487875584",
6. "id":"54887487875584",
7. "id":"54887487875584",
8. "id":"54887487875584",
9. }

# Fehlerbehandlung

Neben einem spezifischen Fehlercode, soll der http-Status als Fehlercode genutzt werden!

# Notizen für Später / Nützliches

* Für das Sortieren der Informationen / Listen kann URI-Parameter **sort** genutzt werden, dem eine kommaseparierte Liste von Attributen der Ressource mit jeweils führendem + (aufsteigende Sortierung), oder – (absteigende Sortierung) zugewiesen wird
* Wie lösen wir das mit der Sicherheit, Jeder kann nur seine Freunde sehen etc.

# Offene