**Abwägungen :**

Allgemeingültig: In unauthentifizierten Zustand können nur GET-Verben auf die Ressourcen (außer Account) angewendet werden.

Es wird die Annahme getroffen dass alle Nutzer des Kickers die App benutzen und das System nicht missbrauchen.

**Ressource Account**   
Die Ressource Account realisiert das Rechtesystem und beinhaltet lediglich eine eindeutige Email-Adresse sowie ein dazugehöriges Passwort. Es wird unterschieden zwischen Accounts für Spieler und Accounts für Standorte / Anbieter von Kickertischen.   
Je nach Typ des Accounts verweist eine Insanz der Ressource Account auf einen Benutzer oder einen Standort.

Die Ressource account enthält abhängig von ihrem Typ Verweise auf Benutzer oder Standort Ressourcen.

E-Mail und Passwort sind keine Sub-Ressourcen , die getrennt voneinander identifiziert werden.

GET: Validieren ob gültige E-Mail und Passwort, daher vom Client als XML übertragen (Content-Type). Server schickt Dokument in text/html als Content-Type zurück mit Statuscode 200. Wenn der Client im Accept Header einen nicht unterstützten Content-Type erfodert schickt der Server einen Statuscode 406 (Not Acceptable) zurück.

POST: Server schickt Status Code 201 zurück wenn die Eingaben erfolgreich validiert wurden. Server schickt Status Code 400 wenn die Daten nicht vollständig waren.

PUT: Server schickt Status Code 204 wenn die Accountdaten erfolgreich modifiziert wurden. Server schickt Status Code 403 wenn die Accountdaten nicht modifiziert werden dürfen.

DELETE: Kein Content-Type vom Client gesendet. Server antwortet mit Content Type 202 um zu zeigen dass die Anfrage verstanden wurde. Server antwortet mit Content Type 401 wenn man nicht eingeloggt ist.

**Ressource Benutzer:**   
Die Ressource Bentuzer beinhaltet seinen Name, Alter, bevorzugte Position , evtl Profilbild , Spielstatistik. Diese Entitäten werden nicht als Subressourcen modelliert, weil dies als zu feingranular empfunden wurde. Daraus folgt zwangläufig, dass bei einer Änderung einer dieser Information mittels PUT stets auch die nichtgeänderten Positionen übermittelt werden müssen.

GET: Im Request kein Content-Type angeben und Server schickt Content Type text/html zurück.

POST: Der Client schickt die Daten als Content-Type application/xml damit die validiert werden können.

PUT: Der Client schickt die Daten als Content-Type application/xml und Server schickt Content Type 200 wenn Daten erfolgreich geändert wurden (text/html).

DELETE: Kein Content-Type vom Client gesendet. Server antwortet mit Content Type 202 um zu zeigen dass die Anfrage verstanden wurde.

**Ressource Match:**

Jedes Match kennt seine Spieler , seinen Kickertisch , Datum und Uhrzeit der Austragung und seinen aktuellen Spielstand. Die Spieler eines Matches werden über Hypermedia Links auf ihre Benutzerressourcen identifiziert. Datum und Uhrzeit eines Matches werden nicht als Subressourcen modelliert, denn die Änderung dieser Information sollte mit dem Anlegen eines Matches übertragen werden. Es wird nicht davon ausgegangen ,dass diese Information frequent geändert wird. Datum und Uhrzeit werden also als Attribut der Ressource Match modelliert. Es wird sichergestellt , dass ein Match entweder aus 2 oder 4 Mitspielern besteht , damit eine Spielerstatistiken aussagekräftig sind.   
Der Spielstand eines Matches soll bei jedem Tor aktualisiert werden , daher wird die Spielstand Information als Subressource des Matches (mit eigenem URI Template) modelliert, sodass Änderungen mittels PUT ohne Übertragung der gesamten Matchressource stattfinden können.

Ein DELETE wird automatisch nach einem ausreichend großen Zeitraum ausgeführt wenn der Spielstand kein Ergebnis hervorbringen konnte.

Sonderfall :   
Ein Match wird vorzeitig abgebrochen : Beide Parteien bekommen die Möglichkeit zu bestätigen ,dass das Match nicht gewertet werden soll . Wenn nur eine Partei abbricht wird das Spiel als Niederlage für diese Partei gewertet.

**Ressource Tisch:**   
Jeder Kickertisch kennt seinen Standort, einen eindeutigen Identifier innerhalb dieses Standorts , seinen Hersteller / Typ sowie einen kurzen Kommentar zu dessen Zustand.

Jeder Tisch kennt seine Belegungssituation (generiert aus der Matchressource die dort stattfindet) kann gefordert werden, und verweist auf alle seine Forderungen.

Eine Forderung ist eine Subressource des Kickertischs und beinhaltet die Information wann die Forderung gestellt wurde sowie Links zu den fordernden Spielern. Es ist möglich als Einzelperson zu fordern , sowie als Team. Einzelpersonen werden dynamisch mit einem Mitspieler gematcht , ohne die Forderungsreihenfolge zu missachten.

Sonderfall 1: Am Tisch findet ein 1gg1 statt, eine Einzelperson fordert , der Verlierer des 1gg1 muss weichen.

Sonderfall 2 : Es findet ein 2gg2 statt , eine Einzelperson fordert , nach der Niederlage weichen 2 Spieler. Bei 3 Spielern am Tisch wird stets auf ein 1 gg 1 zurückgefallen. Die Entscheidung wer aus dem gewinnenden Team weichen muss wird nach dem anmeldenden Benutzer gefällt.

Sonderfall 3: Am Tisch findet ein 2gg2 statt , eine Einzelperson fordert, ein Spieler aus dem Verliererteam muss weichen , das Team wird mit der Einzelperson aufgefüllt.   
  
Sonderfall 4 : Am Tisch findet ein 2gg2 statt, eine Einzelperson fordert , es existieren weitere Forderungen von Zweierteams, die hinter der Einzelperson gefordert haben.

**Ressource Forderung:**

Man kann nur an einem speziellen Tisch fordern mit einer Instanz von Forderung. Die Ressource Forderung beinhaltet die Uhrzeit und den Tisch an dem gefordert wurde.

POST: Stellen einer Forderung an einen bestimmten Tisch. Bei Erfolg zeigt der Location Header auf die URI die angelegt wurde jedoch wird dies dem Client nicht im Payload übermittelt sodass er keine Zugriffsmöglichkeit darauf hat da ein GET auf die Ressource Forderung nicht erlaubt ist.

**Ressource Standort:**Ein Standort ist ein Ort, der mind. Einen Kickertisch im System zur Verfügung stellen will. Er ist identifiziert durch seine Adresse, einen Namen und evtl. eine Beschreibung der Lokalität. Ein Standort verfügt über Rechte zum Anlegen und Entfernen von Kickertischen, und verweist über Hyperlinks auf die von ihm bereitgestellten Tische.

Außerdem enthält jeder Standort links auf die von ihm geplanten Turniere

**Ressource Tunier :**

Ein Turnier ist eine zeitlich geregelte Abfolge von Matches (und seiner Teilnehmer). Die Attribute eines Turniers sind seine Teilnehmerzahl , sein Typ , sein Austragungszeitraum .  
Ein Turnier verweist auf seine Teilnehmer (Link auf Benutzerressource) und die Kickertische an dem es ausgetragen wird (Link auf Kickertischressource). Mittels QueryStrings der Form /Benutzer/{id}?select=name,profilbild werden Teilinformationen über einen Benutzer in die Repräsentation eines Turniers eingfügt. Für jedes Turnier werden über die einzelnen Benutzerstatistiken ein durchschn. Spielerniveau ermittelt.   
Wenn ein Benutzer ein Turnier anlegen will, spezifiziert den Turniertyp und die Spielerzahl , ein Turnier wird dann wahlweise für die Öffentlichkeit oder für einen ausgewählten Spielerkreis geöffnet.