

## Datenstrukturen

Dieses Dokument soll die in JSON-Schema modellierten Datenstrukturen erläutern. Die modellierten Datenstrukturen überführen die im Abschnitt „Information und Attribute(WBA Modellierungen)“ modellierten Attribute in eine Schemasprache. Um Redundanz in diesem Artefakt zu Vermeiden wird nicht jedes modellierte Attribut in den JSON-Schemata noch einmal erläutert.

Bei der Ausgestaltung der JSON-Schemata wurde sich an bereits bestehenden Konventionen dieser noch jungen Schemasprache orientiert. [1] Es existieren auch hier bereits einige Werkzeuge zur Validation von Instanzdokumenten, deren Einsatz zur im Projektkontext geprüft wird. Die unter [1] modellierten Schemata richten sich nach bekannten Darstellungen, veröffentlicht von microformats.org und verschiedenen RFCs. Um ein gewisses Maß an Änderbarkeit dieser Schemata gewährleisten zu können wurden die Verweise auf diese externen Schemata in einem Schema zentralisiert (siehe „definitions.json“). Auf diese Weise können alle im Projekt-Namespace definierten Schemata auf das definitions-Schema verweisen. Zur Übersicht über die Verweisstruktur der spezifizierten Schemata wurde eine Schematische Abbildung angefertigt (siehe Abbildung 1).

Zur Darstellung von Zeitpunkten welche im System eine zentrale Rolle spielen wird das date-time Format ,welches im JSON-Schema Namespace definiert ist, verwendet. Diese Darstellung implementiert die Datetime-Darstellung aus RFC 3339 , Section 5.6. . Datetime-Strings in diesem Format verfügen über gute Kompatibilität mit dem Javascript-Date-Object ,was die Verarbeitung auf Seiten des Dienstgebers voraussichtlich erleichtert. Da dieses Datetime-Format (YYYY-MM-DDTHH:mm:ss:sssZ) In ISO 8601 standardisiert wurde existieren auch in der Implementierungssprache des Dienstgebers (Javascript) bereits einige Implementierungen zur Konvertierung mit diesem Format.

Auch zur Darstellung von Emails und Uris (siehe „Card“-Attribut in Benutzer, Lager, Tafelverein) wurde auf bestehende Darstellungsformen aus RFC 3986 (URI) , RFC Section 3.4.1 (Email) zurückgegriffen. Von der Verwendung des „Calendar“-Schemas aus dem JSON-Schema.org Namespace wurde abgesehen, da diese Repräsentation lediglich einen einzelnen Standort enthält. Zur Repräsentation der Ressourcen „Sammelaktion“ und „Transportaktion“ werden jedoch unterschiedliche Standorte zu unterschiedlichen Zeitpunkten benötigt.

Aus den modellierten JSON-Schemata lassen sich Repräsentationen der modellierten Ressourcen erzeugen. Zur persistenten Datenhaltung auf Seiten des Dienstgebers (unter MongoDB) wird kein gesondertes Schema angegeben. Auch von der Verwendung eines Object-Modelling Tools wie Mongoose wurde aufgrund unzureichender Erfahrung abgesehen.

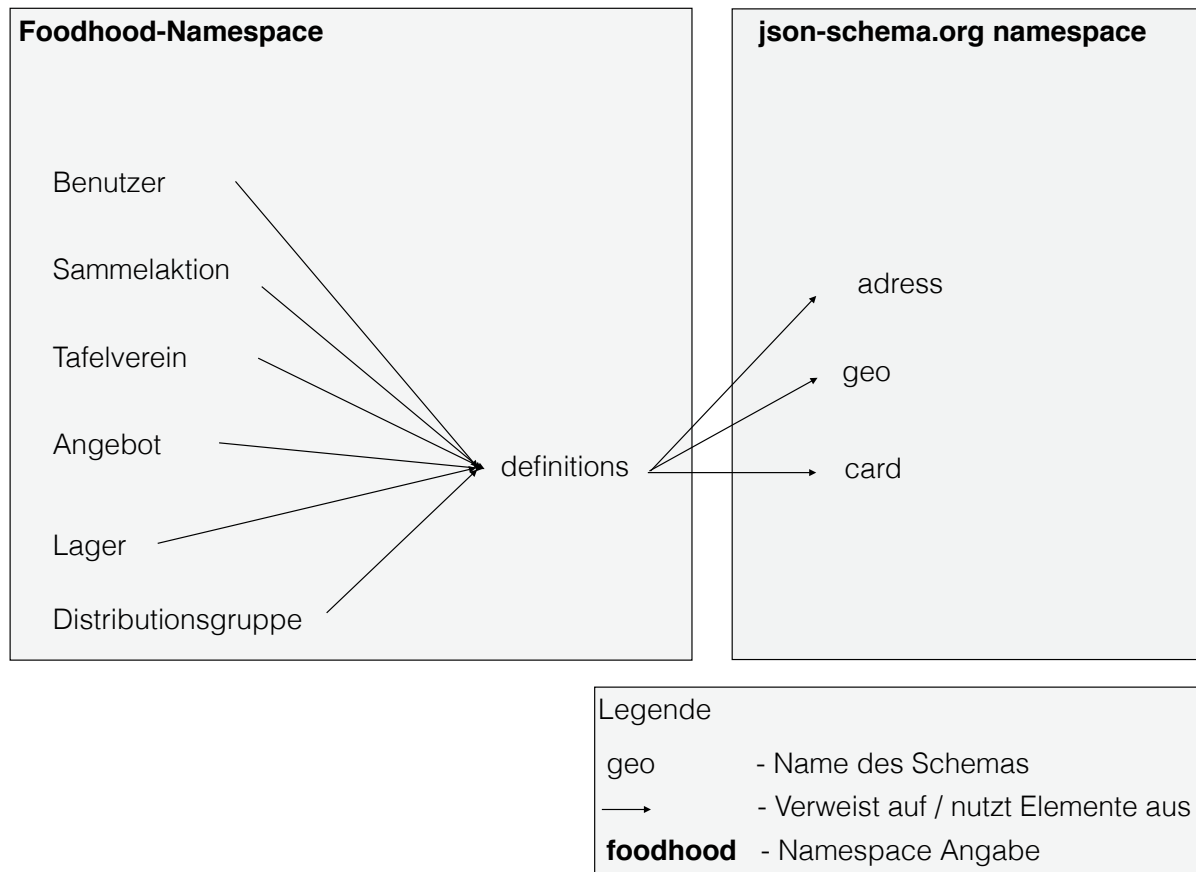


Abbildung 1 -Verweisstruktur der modellierten JSON-Schemata

## Quellen

[1] <http://json-schema.org/documentation.html>  
(Zuletzt eingesehen 20.12.2015 )(siehe Unterpunkt „standard schemas“)

JSON-Schema:

[2] <http://json-schema.org/latest/json-schema-core.html>  
(Zuletzt eingesehen 20.12.2015 )

[3] <http://json-schema.org/latest/json-schema-validation.html>  
(Zuletzt eingesehen 20.12.2015 )

[4] <http://json-schema.org/latest/json-schema-hypermedia.html>  
(Zuletzt eingesehen 20.12.2015 )

[5] <https://www.ietf.org/rfc/rfc3339.txt>  
(Zuletzt eingesehen 20.12.2015)

[6] <http://tools.ietf.org/html/rfc3986>  
(Zuletzt eingesehen 20.12.2015)