**Beschreibung der angewandten REST-Prinzipien**

Die REST-Architektur besteht aus den folgenden Prinzipien:

* **Unterstützung der Client-Server-Architektur:**

🡪 Der Client sendet HTTP-Anfragen (GET, POST, PUT und DELETE) an die Java-Servlets des Servers. Dieser wiederrum kommuniziert, mit Unterstützung verschiedener Java-Klassen und über eine DatabaseFacade-Klasse, mit der Datenbank. An den Client wird eine JSON-Struktur zur Repräsentation der Ressourcen zurückgeliefert.

* **Stateless:**

🡪 Innerhalb unseres Projektes wird mit Zustandslosigkeit gearbeitet. Dies bedeutet, dass Zustandsinfos zwischen zwei Nachrichten nicht gespeichert werden. Die einzige Ausnahme hierbei bildet das Hinterlegen der Anmeldeinformation in Form eines Cookies. So kann der Server jederzeit nachvollziehen ob der Client ein bereits registrierter und angemeldeter User ist. Für diesen ergibt sich dadurch den Vorteil, dass er sich nicht bei jedem Besuch auf der Webseite "My2Hand" neu einloggen muss.

* **Einheitliche Schnittstellen:**

🡪 Adressierbarkeit von Ressourcen

* + Alle Ressourcen sind über ein eindeutiges URI erreichbar. So sind zum Beispiel über das URI „http://localhost:8080/Gruppe5/SecondHand/RestAPI/Artikel“ alle Artikel aus der Datenbank zu erreichen. Hier wird der Inhalt als JSON-Format zurückgegeben.

🡪 Repräsentationen zur Veränderung von Ressourcen

* + Ändern sich Werte in der Datenbank oder kommen welche hinzu (z.B. eine neue Filiale in Stuttgart) so sind keine Anpassungen im Rest-Service der Software nötig. Der Zugriff auf die Daten für den Kunden ist ohne weitere Anpassungen gewährleistet.

🡪Selbstbeschreibende Nachrichten

* + Über einheitliche Namensgebungen im Code, kann jeder aus der Gruppe den Code schnell und einfach lesen. Bekannte und etablierte Methodenaufrufe wie GET, POST, PUT und DELETE wurden verwendet.

**Unser Framework:**

**AJAX** bezeichnet die asynchrone Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, ohne die Seite komplett neu zu laden. Der HTML-Code wird schließlich, mithilfe von DOM-Manipulationen und der HTTP-Response des AJAX-Aufrufs, verändert.