Representational State Transfer, kurz REST wird erstmals in der Dissertation von Roy Fielding im Jahr 2000 vorgestellt [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\_arch\_style.htm]. REST basiert auf den Prinzipien des World Wide Web und fast sie zu einem Architekturstil zusammen. Dabei wird unter andrem beschrieben wie Ressourcen im WEB identifiziert und adressiert werden können.

Roy Fielding abstrahierte sich von konkreten Architekturen wie HTTP und URIs. Er legte nur Kernprinzipien fest, die mit unterschiedlichen Protokollen umgesetzt werden könnten. Diese Kernprinzipien lauten [http://www.edv-buchversand.de/productinfo.php?replace=false&cnt=productinfo&mode=2&type=2&id=dp-120&index=2&nr=0&art=Anleitung}]:

1. Ressourcen mit eindeutiger Identifikation

URIs stellen im Web ein einheitliches Konzept für die Vergabe von IDs dar. Durch einen global definierten Namensraum wird sichergestellt, dass Ressourcen weltweit eindeutig identifiziert werden.

1. Hypermedia

Mithilfe dieses Konzeptes ist es möglich andere Ressource zu referenzieren, um beispielsweise an zusätzliche Informationen zu gelangen. Ein weiterer wichtiger Aspekt ist die Möglichkeit die Steuerung der Applikation durch Links. Ein Server kann dem Client über Hypermedia-Elemente mitteilen, welche Aktion er als nächstes auszuführen hat.

1. Standardmethoden

Jede Ressource unterstützt den gleichen Satz an Methoden, bei HTTP zählen dazu folgende:

GET, für die Dartstellung von Ressourcen

POST, für das Erstellen einer Ressource

PUT, für das Aktualisieren einer Ressource

DELETE, für das Löschen einer Ressource

HEAD, Methadaten einer Ressource

OPTIONS, es warden Methadaten geliefert welche methoden eine Ressource unterstützt

1. Unterschiedliche Repräsentationen

HTTP verfolgte einen Ansatz zur Trennung der Verantwortlichkeiten für Daten und Operationen. Ein Client der ein bestimmtes Dateiformat verarbeiten kann, ist in der Lage jede Ressource mit diesem Format zu verarbeiten, da die Operationen dafür dieselben sind.

1. Statuslose Kommunikation

Serverseitig wir der Zustand des Clients nicht gespeichert. Der aktuelle Zustand muss vollständig auf Seiten des Clients abgespeichert werden und bei Reqeuests die nötigen Informationen an den Server übermittelt werden.

<http://www.se.uni-hannover.de/priv/lehre_2010sommer_wwwseminar/05-REST-Grundlagen-Ausarbeitung.pdf>

<http://searchsoa.techtarget.com/definition/REST>

State oft he Art

REST, Archtiektur Beschreibung der Methoden

Auswahl der Frameworks

Beschreibung der Frameworks

Evaluierung

Conclusio

<http://www.theseus.fi/bitstream/handle/10024/91409/Rest%20Client%20For%20Android.pdf?sequence=1>

<http://sdsu-dspace.calstate.edu/bitstream/handle/10211.3/135688/Ranebennur_sdsu_0220N_10598.pdf?sequence=1>

<http://www.blogwerk.com/2014/08/18/entscheidungshilfe-app-responsive-website-mobile-website/>

[http://download.springer.com/static/pdf/436/bok%253A978-3-642-22259-7.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Fbook%2F10.1007%2F978-3-642-22259-7&token2=exp=1442993017~acl=%2Fstatic%2Fpdf%2F436%2Fbok%25253A978-3-642-22259-7.pdf%3ForiginUrl%3Dhttp%253A%252F%252Flink.springer.com%252Fbook%252F10.1007%252F978-3-642-22259-7\*~hmac=042e5734ad9ecaf02a1f19445daed02700f88cd3ce76c48ed79834705ba9dbaa](http://download.springer.com/static/pdf/436/bok%253A978-3-642-22259-7.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Fbook%2F10.1007%2F978-3-642-22259-7&token2=exp=1442993017~acl=%2Fstatic%2Fpdf%2F436%2Fbok%25253A978-3-642-22259-7.pdf%3ForiginUrl%3Dhttp%253A%252F%252Flink.springer.com%252Fbook%252F10.1007%252F978-3-642-22259-7*~hmac=042e5734ad9ecaf02a1f19445daed02700f88cd3ce76c48ed79834705ba9dbaa)

**Pros and Cons of some Android frameworks**

http://www.jayway.com/2015/03/04/pros-and-cons-of-some-android-frameworks/

**Retrofit**

Retrofit ist ein typsicherer HTTP Client für Android und wurde von Square entwickelt. Retrofit sagt über sich selbst aus:“Retrofit turns your HTTP API into a Java interface.”

Durch Annotationen bei Interface Methoden und den Parametern ist es möglich anzugegeben wie ein Request zu verarbeiten ist. Jede Methode muss eine HTTP Annotation haben, die angibt welche Request Methode zu verwenden ist und die relative URL.

@Path("id")

@Body

|  |  |  |
| --- | --- | --- |
| Aktive Community |  |  |
| Tutorial |  |  |
| HATEOS unterstützung |  |  |
| Diskussionen bei stackflow |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |