



LoungeLog Core

A RESTful web-service for social media tracking



Christian Breil
Medieninformatik Oberseminar SoSe 2010
Abschlussvortrag Bachelorarbeit
Betreuer: Dominikus Baur
Verantw. Hochschullehrer: Prof. Dr. Andreas Butz

“a RESTful web-service for social media-tracking”

- Mediatracking-Service:
 - Dienst um den eigenen Medienkonsum aufzuzeichnen
- Social
 - Freundesnetzwerk
 - Anbindung an weitere Netzwerke wie Facebook, last.fm
- REST
 - Zustandslose Architektur um Zugang über HTTP zu Ressourcen zu ermöglichen.
 - Datenaustausch z.B. über XML

- “Self-knowledge”
- Medientypen übergreifende Statistiken
- Ein soziales Netzwerk um sich über Medieninhalte auszutauschen
- Aggregieren von anderen Trackingdiensten
- Empfehlungen aufgrund aller konsumierten Medien möglich

Medientypen

book	Romane, Fachliteratur, eBooks, audio books...
music	Songs und Alben
movie	Kinofilme, große TV-Produktionen...
tv	Serien, Nachrichten, Dokumentarfilme...
game	Brettspiele, Kartenspiele, Würfelspiele...
videogame	Computer- und Konsolenspiele
web	Webseiten, Blogbeiträge, YouTube Videos...
paper	Zeitschriften, Zeitungen, wiss. Publikationen...

- Welchen Medieninhalt (Medialtem)
- Wer
- Wann
 - Start und Ende des Konsums
- Wo (optional)
- Wie nennt der User das Ereignis? (optional)
- Tags (optional)

- LoungeLog Web
 - Plattform und „Community Website“
 - Volle Funktionalität von LoungeLog
 - Integration von Facebook
- LoungeLog Mobile
 - Mobile App zum schnellen Tracken unterwegs
 - Als Zugang zu Previews und anderen Metadaten
 - Augenmerk auf Usability

“social catalogs”

- Last.fm für Musik und Konzerte



- Personalisierte Streams
- Empfehlungssystem
- Freundesnetzwerk, aber eher untergeordnet
- Umfangreiche API

- Goodreads für Bücher



- „shelves“ um eigene Bücher zu organisieren
- Empfehlungssystem basiert auf Freundesnetzwerk
- Umfangreiche API

[Bilder | <http://www.last.fm/resources>; <http://www.goodreads.com/about/press>]

“social catalogs”

- DAYTUM für alles
- Stark angelehnt an N. Feltron’s „Annual Reports“
- Hauptsächlich Durchschnittswerte und einfache Charts
- Tracking z.B. via Twitter
- API in Planung
- 4\$ im Monat, falls Privacy erwünscht



- REpresentational State Transfer (REST)
 - Jede Ressource
 - hat eine eindeutige URI
 - verschiedene mögliche Repräsentationen (XML, JSON, pdf...)
 - Der Zugriff
 - erfolgt über die HTTP Methoden GET, POST, PUT & DELETE analog zu SQL select, update, insert, delete
 - ist Zustandslos – jede Operation ist in sich abgeschlossen
- ➔ REST ist weitgehend plattformunabhängig
 - HTTP Protokoll kann jeder
 - XML kann jeder

- Jede Ressource hat eine URI?

Beispiele:

- User:

loungeLog.de:8080/mediaTrack/rest/user/hansiH

loungeLog.de:8080/mediaTrack/rest/user/1/friends

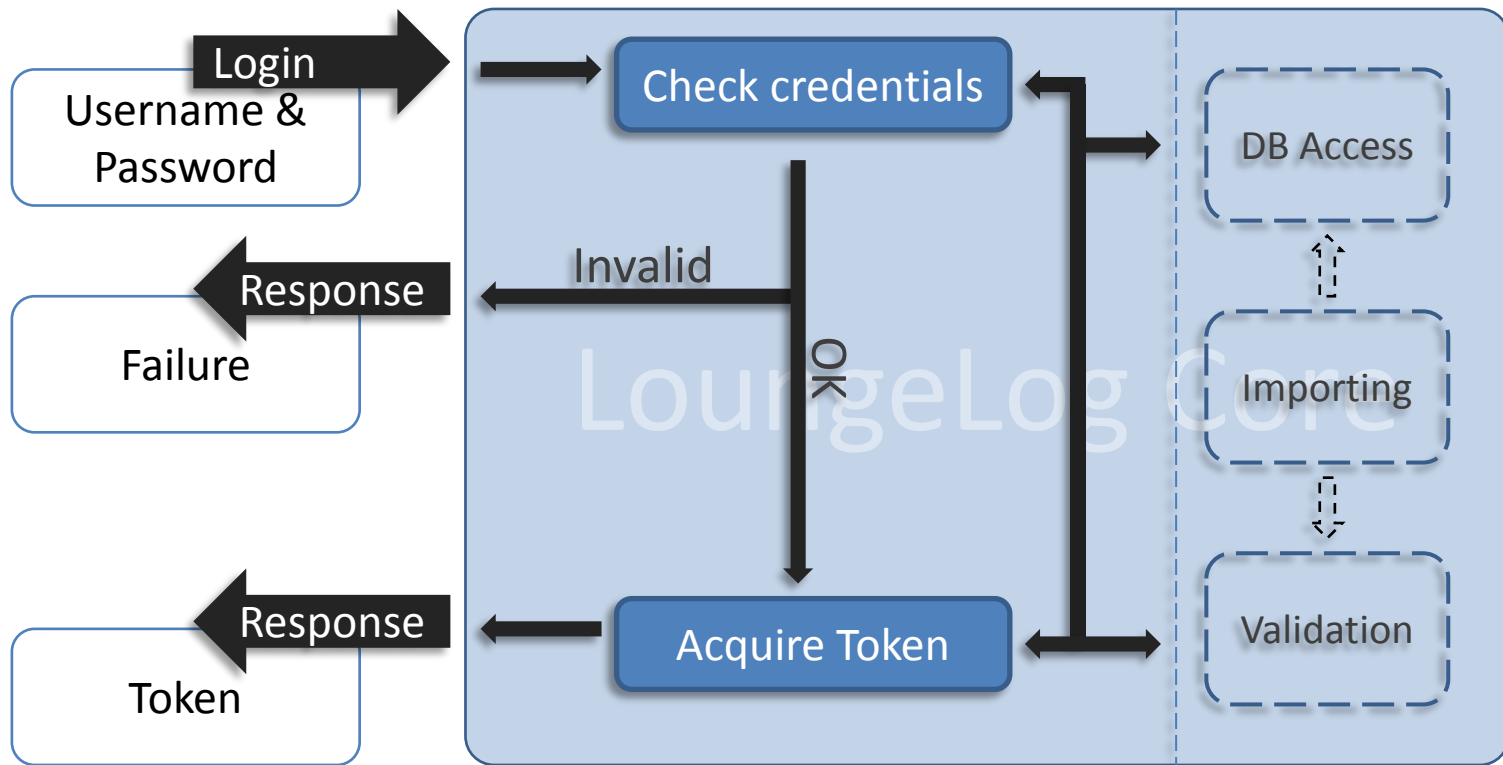
- Medialtems:

loungeLog.de:8080/mediaTrack/rest/medialtem/142715

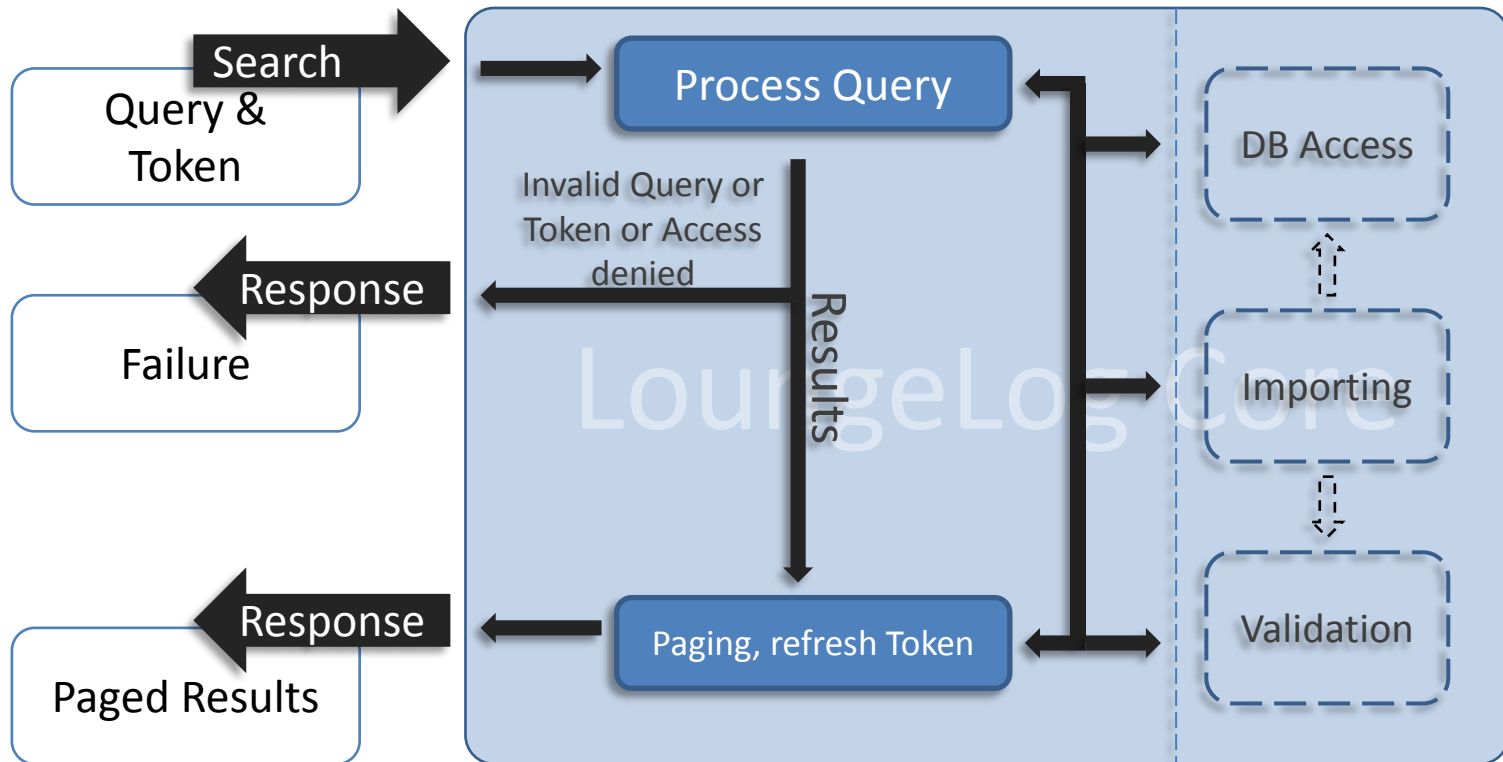
- Suchen:

loungeLog.de:8080/mediaTrack/rest/search/medialtem;type=book;
name=Harry%20Potter%20and

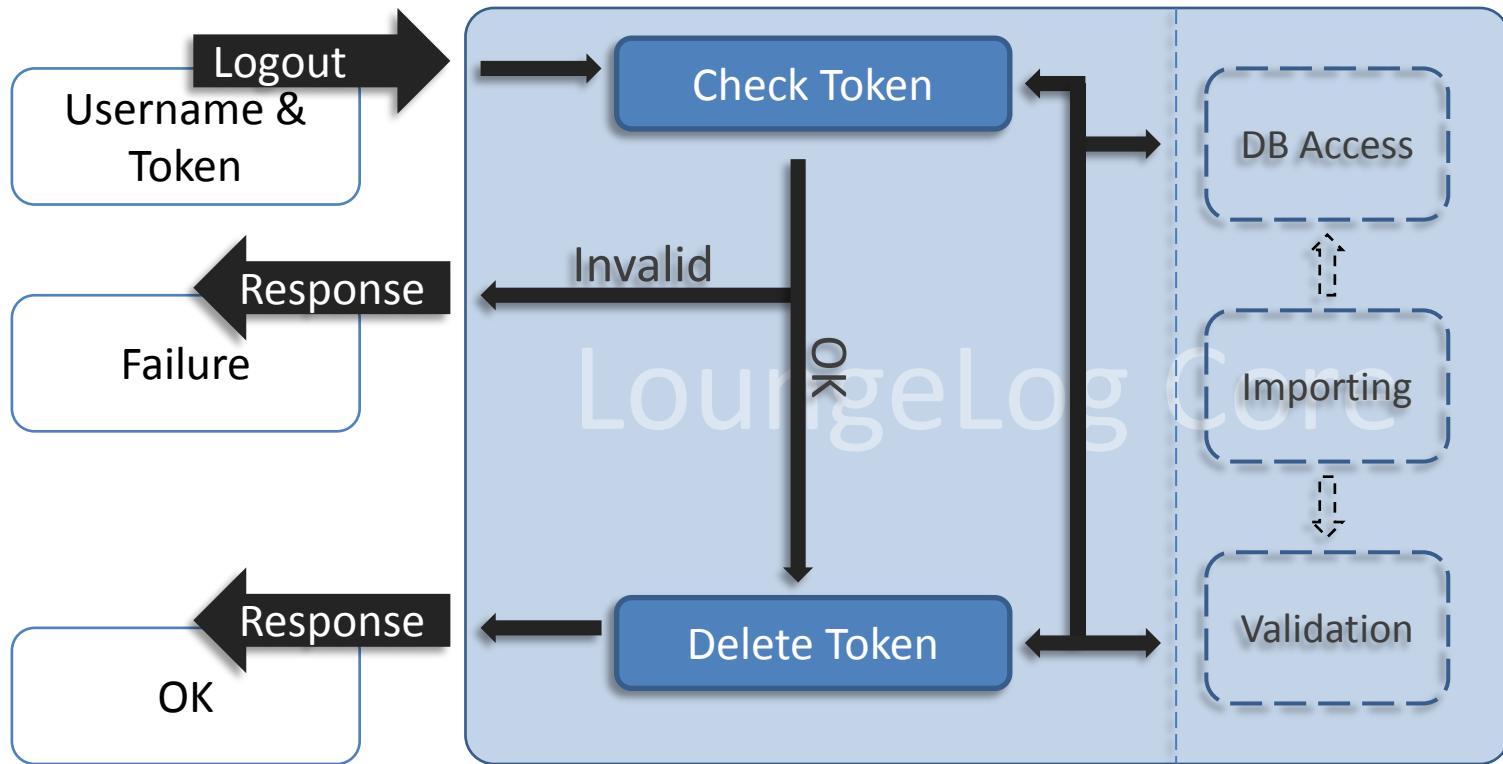
Login



Suchen & automatischer Import



Logout



- Improvements
 - Volltextsuche
 - Tiefere Integration anderer Netzwerke
 - Änderungshistorien
 - Umstieg auf Java EE und Application Servers
- Features
 - Pinnwand
 - Empfehlungssystem
 - Events (Konzerte, Treffen..)
 - (Weitere) Statistiken

LoungeLog

Vielen Dank.



Referenzen:

- R.T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. Dissertation, University of California, Irvine, 2000
- C. Pautasso, O. Zimmermann, F. Leymann. *Restful web services vs. "big" web services: making the right architectural decision*. In Proceedings of the 17th international conference on World Wide Web(WWW '08), Beijing, China, 2008

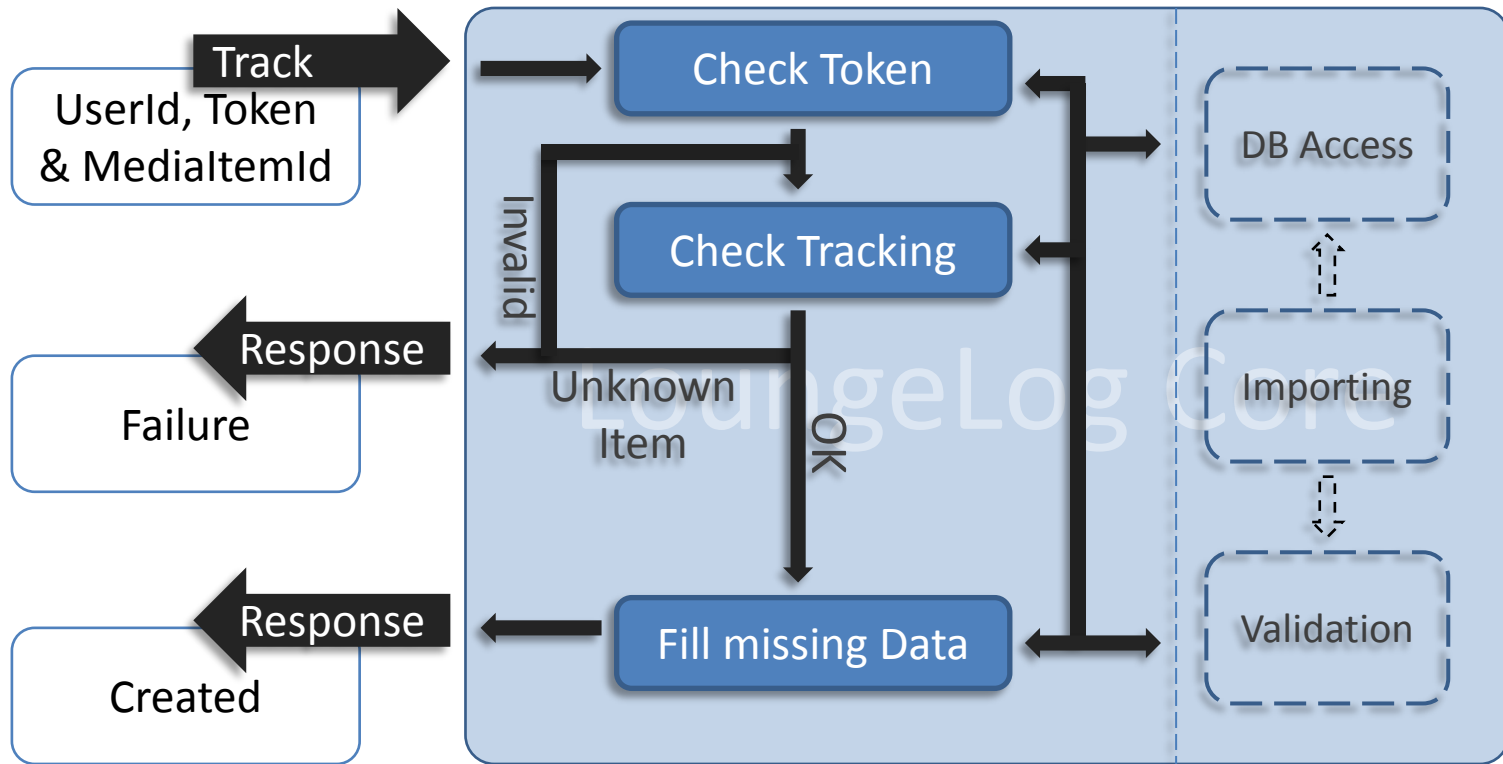
Web-Referenzen:

- <http://hibernate.org>,
<http://hibernate.org/subprojects/validator>,
<http://jboss.org/resteasy>,
<http://www.mysql.com>,
<http://tomcat.apache.org>,
<http://musicbrainz.org>,
<http://themoviedb.org>,
<http://giantBomb.com>
- <http://jcp.org/en/jsr/summary?id=317>,
<http://jcp.org/en/jsr/summary?id=303>,
<http://jcp.org/en/jsr/summary?id=311>,
<http://www.mchange.com/projects/c3p0>,
- <http://thetvdb.com>,
<http://books.google.com>

Bilder von:

- <http://loungeLog.de>
<http://www.last.fm>
<http://www.goodreads.com>
<http://www.daytum.de>

Tracking



```
<response totalResultCount="11" statusCode="OK" page="0" maxPage="1">
  <content>
    <item xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xmlBook">
      <artists>
        <artist>
          <id>294298</id>
          <name>J. K. Rowling</name>
        </artist>
      </artists>
      <childCount>0</childCount>
      <id>296289</id>
      <name>Harry Potter and the Chamber of Secrets</name>
      <picUrl>[...]</picUrl>
      <published>1037574000000</published>
      <tags>
        <tag><name>Education</name></tag>
      </tags>
      <type>book</type>
    </item>
    <item xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xmlMovie">
      [...]
    </item>
    [...]
  </content>
</response>
```

```
<response totalResultCount="1" statusCode="OK" page="0" maxPage="0">
  <content>
    <item xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xmlUser">
      <firstName>Hansi</firstName>
      <id>121417</id>
      <lastName>Hinterseer</lastName>
      <lang>de</lang>
      <nick>hansiH</nick>
    </item>
  </content>
</response>
```

```
{ "response": {"@totalResultCount": "1", "@statusCode": "OK", "@page": "0", "@maxPage": "0",
  "content": {
    "item": {"@type": "xmlUser",
      "firstName": "Hansi",
      "id": 121417,
      "lastName": "Hinterseer",
      "lang": "de",
      "nick": "hansiH"
    }
  }
}
```

```
{ "response": {"@totalResultCount": "0", "@statusCode": "NO_SUCH_USER", "@page": "0", "@maxPage": "0",
  "content": "Unknown user: '654'"
}
```

- Umgebung

- Java SE 6.0.15
 - RESTEasy 2.0.0
(JSR 311: JAX-RS)
 - Hibernate Core 3.5.3
(JSR 317: Java Persistence 2.0)
 - Hibernate Validator 4.1.0
(JSR 303: Bean Validation)
- Apache Tomcat 6.0.26
- MySql 5.0.45
 - C3P0 0.9.1
(Connection Pool)

- Datenquellen

- musicbrainz.org
(music)
- themoviedb.org
(movie)
- thetvdb.com
(tv)
- books.google.com
(books)
- giantbomb.com
(videogame)

- User

- /new/user
- /user/{ *nickOrId* }
- /user/{ *id* }/update
- /user/{ *id* }/uploadProfilePic
- /user/{ *id* }/history
- /user/{ *id* }/lists
- /user/{ *id* }/list/{ *listname* }
- /user/{ *id* }/friends
- /user/{ *id* }/ownPendingRequests
- /user/{ *id* }/requestFriendship
- /user/{ *id* }/acceptFriendship
- /user/{ *id* }/declineFriendship
- /user/{ *id* }/cancelFriendship
- /user/{ *id* }/withdrawFriendshipRequest

Artist

- /new/artist
- /artist/{ *id* }
- /artist/{ *id* }/update

- HistoryItem

- /new/historyItem
- /historyItem/{ *id* }
- /delete/historyItem/{ *id* }

- MediaItem

- /mediaItem/{ *id* }
- /mediaItem/{ *id* }/detachChild
- /mediaItem/{ *id* }/update
- /mediaItem/{ *id* }/doTag
- /mediaItem/{ *id* }/addChild
- /new/mediaItem

- Sonstiges

- /search/tag
- /login
- /logout/{ *id* }
- /new/list
- /delete/list
- /search/user
- /search/mediaItem