

Technische Universität Dortmund
Fakultät Informatik
Lehrstuhl 5, Prof. Dr. Bernhard Steffen
Wintersemester 2012/13

Aktuelle Themen der Dienstleistungsinformatik

Projekt: Kontakte

Markus Marzotko, Thorben Seeland & Dominic Wirkner
17. Januar 2013

Inhaltsverzeichnis

1	Einleitung	3
1.1	Projektbeschreibung	3
1.2	Erste Überlegungen	3
2	Unsere Projektbausteine	3
2.1	Google-Connector	3
2.1.1	Die Bibliothek <i>gdata</i>	3
2.1.2	Authentifizieren und Verbinden mit <i>gdata</i>	3
2.1.3	Kontakte suchen	5
2.1.4	Kontakte einfügen	6
2.2	SAP-Connector	7
2.3	GUI-Elemente	7
2.4	SIB-Klassen	7
3	Zusammenfassung / Eigene Meinung / Weiterführendes	7
4	Literaturverweise	8

1 Einleitung

Hinführung auf Beschreibung und erste Überlegungen

1.1 Projektbeschreibung

Beschreibung Ähnlich der auf den Folien

1.2 Erste Überlegungen

Aufteilung des Projektes (siehe Grafik). Wahl der Contact-Attribute. Management über SAP-/Google-IDs.

2 Unsere Projektbausteine

Im Folgenden Beschreibung, besondere Überlegungen und Probleme(!)/Entscheidungen der einzelnen Elemente

2.1 Google-Connector

2.1.1 Die Bibliothek *gdata*

Die Google-Bibliothek *gdata* ist eine frei verfügbare Bibliothek zum erstellen von Clientapplications für die Services der Google-Cloud. *gdata* kapselt die Webservices komplett in Java-Klassen, so dass ein importieren (z. B. mit *wsimport*) nicht mehr notwendig ist.

In der Beschreibung der *gdata*-Bibliothek wird beschrieben, wie man aus der zip-Datei ein JAR compilieren kann, da dies bei mir in mehreren Versuchen nicht geklappt hat, haben wir die pragmatische Lösung gewählt und die in der zip-Datei enthaltenen JARs von Hand in das Projekt eingefügt.

2.1.2 Authentifizieren und Verbinden mit *gdata*

Google bietet zwei Authentifizierungsverfahren an

1. *OAuth*
2. Username und Passwort

OAuth ist ein Service, der bei erfolgreicher Anmeldung ein Token erstellt, mit dem der Client von Google bereitgestellte Services aufrufen und sich authentifizieren kann. So muss der Client die Anmelde-Daten des Nutzers nicht speichern, sondern nur den Token. In Abbildung 1 wird ein Beispiel für die Nutzung von *OAuth* dargestellt.

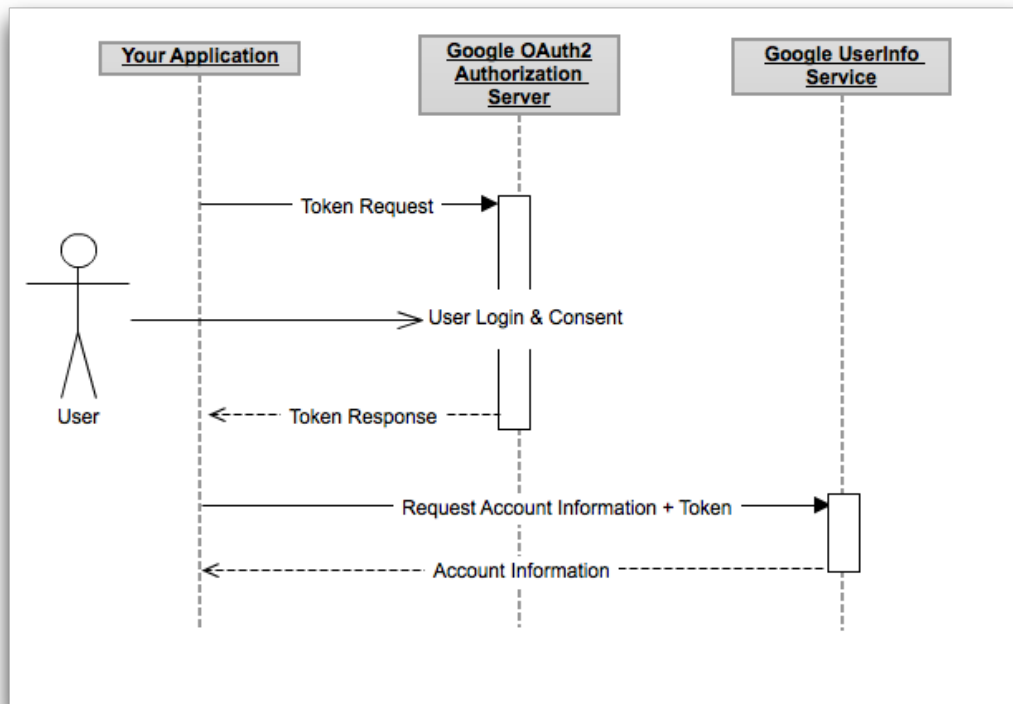


Abbildung 1: Nutzungsbeispiel für *OAuth*[5]

Da wir unseren eigenen Account nutzen und die Daten nicht sicherheitskritisch sind, haben wir die zweite Variante gewählt und authentifizieren uns bei jedem Service-Aufruf mit Username und Passwort.

Die Authentifizierung wird für ein *ContactsService*-Objekt, wie in Listing 1 abgebildet, einmal durchgeführt, danach wird sie vom Framework automatisch durchgeführt.

Listing 1: Beispiel für die Authentifizierung ohne *OAuth*

```
ContactsService myService;  
myService = new ContactsService(servicename);  
try {
```

```
        myService.setUserCredentials(username, password);
    } catch (AuthenticationException e) {
        e.printStackTrace();
    }
}
```

2.1.3 Kontakte suchen

Die *gdata*-Bibliothek bietet die Möglichkeit, Kontakte wie in Listing 2 mit Angabe eines *Query*s herunterzuladen. Das *Query* kann jedoch nur zwischen Gruppen unterscheiden, jedoch nicht nach anderen Kriterien wie z. B. dem Vornamen oder dem Nachnamen eines Kontakts filtern.

Listing 2: Kontaktsuche per Query

```
URL feedUrl = new URL(contactsURL);
Query myQuery = new Query(feedUrl);
ContactFeed resultFeed = null;
// Gruppe
String groupId = null;
// Parameter Contact filter
switch (filter.getType()) {
case CUSTOMER:
    groupId = customerGroupURL;
    break;
case SUPPLIER:
    groupId = supplierGroupURL;
    break;
case EMPLOYEE:
    groupId = employeeGroupURL;
    break;
default:
    break;
}
myQuery.setStringCustomParameter("group", groupId);
// submit request
resultFeed = myService.getFeed(feedUrl, ContactFeed.class);
```

Das Suchen von Kontakten geschieht in unserem Projekt durch das Herunterladen aller Kontakte einer Gruppe und anschließend sortieren „von Hand“.

2.1.4 Kontakte einfügen

Das Einfügen von Kontakten ist über ein erstelltes Service-Objekt

Listing 3: Kontakt-Objekt erstellen

```
// Create the entry to insert
ContactEntry contact = new ContactEntry();
contact.setTitle(new
    PlainTextConstruct(contactInfoCopy.getFirstname()
        + contactInfoCopy.getLastname()));
```

Listing 4: Namen in ein Kontakt-Objekt einfügen

```
// Name
Name name = new Name();
name.setFamilyName(new
    FamilyName(contactInfoCopy.getLastname(), null));
name.setGivenName(new
    GivenName(contactInfoCopy.getFirstname(), null));
contact.setName(name);
```

Listing 5: Benutzerdefinierte Einträge zu einem Kontakt-Objekt hinzufügen

```
// Firma
if (contactInfoCopy.getCompany() != null) {
    ExtendedProperty company = new ExtendedProperty();
    company.setName(DLI_GoogleContactsConnector.company);
    company.setValue(contactInfoCopy.getCompany());
    contact.addExtendedProperty(company);
}
```

Listing 6: Den Kontakt einer Gruppe hinzufügen

```
// Gruppe setzen
String groupURL = null;
switch (contactInfoCopy.getType()) {
```

```
case CUSTOMER:
groupURL = customerGroupURL;
contact.addGroupMembershipInfo(new
    GroupMembershipInfo(false, groupURL));
```

Listing 7: Das Kontakt-Objekt senden

```
// Kontakt senden
URL postUrl = new URL(contactsURL);
return myService.insert(postUrl, contact);
```

2.2 SAP-Connector

- Was ist der ES-Workplace?
- Welche WSDLs wurden verwendet? Warum?
- Wie wurden die WSDLs ins Java Format übertragen? (WsImport..)
- Wie lief das Programm ungefähr ab?
- Welche Probleme gab es bezüglich IDs und “normalen Daten”?
- Was könnte diesbezüglich auf eigener Seite verbessert werden?

2.3 GUI-Elemente

2 GUI-Elemente. Gemeinsamkeiten/Konzeption. Validierung in EDIT. Details in CHOOSE.

2.4 SIB-Klassen

Generelles zum Aufbau von SIBs. Besonderheiten

3 Zusammenfassung / Eigene Meinung / Weiterführendes

...

4 Literaturverweise

Literatur

- [1] Wasserman, Stanley und Katherine Faust; *Social Network Analysis: Methods and Applications*; Cambridge University Press, New York; 1995
- [2] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest und Clifford Stein; *Introduction to Algorithms, Second Edition*; The MIT Press and McGraw-Hill Book Company; 2001
- [3] Jacob, Riko, Dirk Koschitzki, Katharina Anna Lehmann, Leon Peeters und Dagmar Tenfelde-Podehl; *Algorithms for Centrality Indices*;
In: Brandes, Ulrik und Thomas Erlebach
- [4] Brandes, Ulrik und Thomas Erlebach; *Network Analysis: Methodological Foundations*; Band 3418 der Reihe *Lecture Notes in Computer Science*; Springer; 2005
- [5] Google Developers; *Using OAuth 2.0 to Access Google APIs*; <https://developers.google.com/accounts/docs/OAuth2>; 17. Januar 2013