

SLOVENSKÁ TECHNICKÁ UNIVERZITA BRATISLAVA
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

TÚTORIÁL KU KNIŽNICI

Tímový projekt

Názov projektu: Objektové úložisko dát
Vedúci tímového projektu: Ing. Lubomír Varga
Akademický rok: 2010/11



Obsah

1 Rozbehanie knižnice	1-1
2 Používanie knižnice	2-1
2.1 Vytváranie DataObjectov a MetadataObjectov	2-1
2.2 Práca s objektami (fotky, videá, . . .)	2-1
2.2.1 Pridanie objektu	2-1
2.2.2 Získanie objektu	2-2
2.2.3 Vymazanie objektu	2-2
2.2.4 Získanie zoznamu všetkých objektov, ktoré má používateľ uložené	2-2
2.3 Práca s metadátami	2-2
2.3.1 Pridanie metadát k objektu	2-2
2.3.2 Vymazanie metadát	2-2
2.3.3 Získanie zoznamu všetkých metadát, ktoré má používateľ uložené	2-2
2.4 Vyhľadanie objektu (fotky, videá) podľa metadát	2-3



Kapitola 1

Rozbehanie knižnice

Knižnica ja dostupná na:

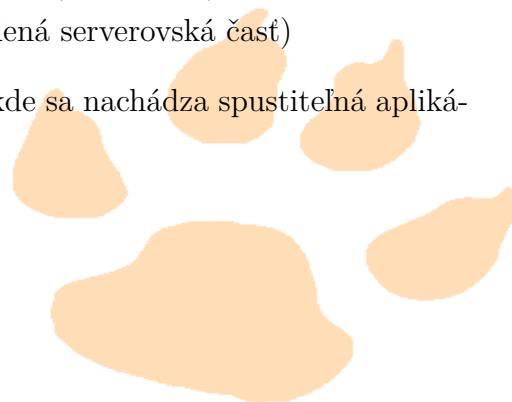
<http://labss2.fiit.stuba.sk/TeamProject/2010/team17is-si/ine.html>

1. stiahnuť knižnicu zo stránky uvedenej vyššie (.zip súbor obsahujúci aj ostatné knižnice, ktoré je potrebné importovať)
2. importovať knižnicu a ostatné knižnice, ktoré sú potrebné do projektu vo vývojovom prostredí
3. v .zip súbore sa nachádza konfiguračný súbor (classes/host-configuration.xml), kde je potrebné nastaviť adresu, kam sa knižnica pripája. Štruktúra konfiguračného súboru (host-configuration.xml) je nasledovná:

```
<server>  
  <hostname>http://localhost:8080/provider/ulozisko/</hostname>  
</server>
```

Podľa servera, na ktorý sa má knižnica pripojiť, je potrebné zmeniť hostname nasledovne: <http://localhost:8080/provider/ulozisko/> nahradiť červenú časť adresy aktuálnou adresou (kde je nasadená serverovská časť)

4. Konfiguračný súbor treba vložiť na miesto, kde sa nachádza spustiteľná aplikácia (napr. jar), ktorá využíva knižnicu.



Kapitola 2

Používanie knižnice

Knižnica obsahuje triedy : `DataObjectRestClient`, `MetadataObjectRestClient`, `RestClientManager`, `ObjectController` a `SecurityRestClient`.

Štruktúra balíkov je nasledovná:

```
sk.fiiit.team17cf.stfu.client.controller
    ObjectController.java
sk.fiiit.team17cf.stfu.client.impl
    DataObjectRestClient.java
    MetadataObjectRestClient.java
    RestClientManager.java
sk.fiiit.team17cf.stfu.client.security
    SecurityRestClient.java
```

2.1 Vytváranie `DataObject`ov a `MetadataObject`ov

```
DataObject dataObject = ObjectController.createDataObject(fileName, objectName);
MetadataObject metaObject = ObjectController.createMetadataObject(object, String predicate);
MetadataObject metaObject = ObjectController.createMetadataObject(String predicate, String subject);
MetadataObject metaObject = ObjectController.createMetadataObject(String subject);
```

2.2 Práca s objektami (fotky, videá, ...)

Na prácu s objektami slúži trieda `DataObjectRestClient`, ktorá implementuje rozhranie `DataObjectManager`.

2.2.1 Pridanie objektu

```
DataObject dataObject = ObjectController.createDataObject(fileName, objectName);
DataObjectRestClient dataRestClient = new DataObjectRestClient();
String result = dataRestClient.add(dataObject);
```

Funkcia `add` vráti id objektu (kde je objekt fyzicky uložený).

2.2.2 Získanie objektu

```
DataObjectRestClient dataRestClient = new DataObjectRestClient();
DataObject dataObject = dataRestClient.get(idObjektu);
```

2.2.3 Vymazanie objektu

```
DataObjectRestClient dataRestClient = new DataObjectRestClient();
dataRestClient.delete(objectId);
```

Vymaže objekt a všetky jeho metadáta.

2.2.4 Získanie zoznamu všetkých objektov, ktoré má používateľ uložené

```
DataObjectRestClient dataRestClient = new DataObjectRestClient();
Collection<DataObject> resultList = dataRestClient.list();
```

Vráti zoznam Idčiek objektu.

2.3 Práca s metadátami

Na prácu s metadátami slúži trieda MetadataObjectRestClient, ktorá implementuje rozhranie MetadataObjectManager.

2.3.1 Pridanie metadát k objektu

```
MetadataObject metaObject = ObjectController.createMetadataObject(IDobject, predicate, subject);
MetadataObjectRestClient metaRestClient = new MetadataRestClient();
String result = metaRestClient.add(metaObj);
```

Funkcia vráti Idobjektu, ku ktorému boli pridané metadáta.

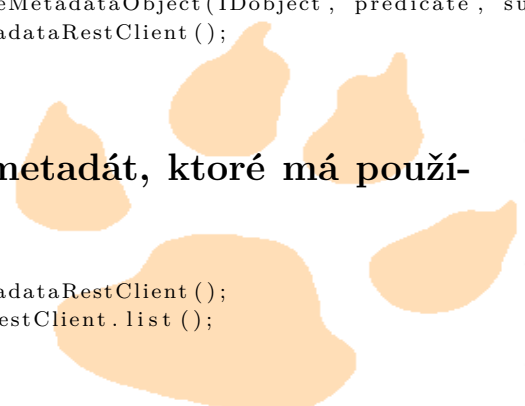
2.3.2 Vymazanie metadát

```
MetadataObject metaObject = ObjectController.createMetadataObject(IDobject, predicate, subject);
MetadataObjectRestClient metaRestClient = new MetadataRestClient();
metaRestClient.delete(metadataObject);
```

Vymažeme zadaný triplet.

2.3.3 Získanie zoznamu všetkých metadát, ktoré má používateľ uložené

```
MetadataObjectRestClient metaRestClient = new MetadataRestClient();
Collection<MetadataObject> metaCollection = metaRestClient.list();
```



2.4 Vyhľadanie objektu (fotky, videá) podľa meta-dát

- je **zadaný iba subject** (tag, ktorým je fotka otagovaná). Napríklad triplet vyzerá nasledovne:

- object: <http://localhost:8080/katka/fotka1.jpg>
- predikát: obsahuje
- subject: slnko

Chceme vyhľadať všetky fotky, ktoré obsahujú slnko. Na to slúži nasledujúci príklad:

```
MetadataObject metaObj = ObjectController.createMetadataObject(subject);
List<MetadataObject> metaObjList = new ArrayList<MetadataObject>();
metaObjList.add(metaObj);
Collection<PreviewObject> previewCollection = metaRestClient.search(metaObjList);
```

Funkcia vráti zoznam, ktorý obsahuje PreviewObject-y (náhlady).

- je **zadaný predikát a subject** (“vzťah” a “tag”). Napríklad triplet vyzerá nasledovne:

- object: <http://localhost:8080/katka/fotka1.jpg>
- predikát: je
- subject: fotka

Chceme vyhľadať všetky objekty, ktoré sú fotky. Na to slúži nasledujúci príklad:

```
MetadataObject metaObj = ObjectController.createMetadataObject(subject);
List<MetadataObject> metaObjList = new ArrayList<MetadataObject>();
metaObjList.add(metaObj);
Collection<PreviewObject> previewCollection = metaRestClient.search(metaObjList);
```

Funkcia vráti zoznam, ktorý obsahuje PreviewObject-y (náhlady).

