Salesforce.com support

SalesforceDataContext

MetaModel's Salesforce.com connector, SalesforceDataContext, makes it possible to query and perform CRUD operations on Salesforce.com data.

Requirements

To connect to Salesforce.com with MetaModel you need to have:

- A Salesforce.com username
- The password that goes with the username
- A security token for that user to access Salesforce.com remotely

Obtaining a security token

The Salesforce.com security token can be retrieved by accessing the setup page in the upper right corner of the Salesforce layout (click [your name] -> Setup).

On the Setup page you'll find a menu item called 'Reset My Security Token'. Navigate to this menu item and have it send you the token in an email.

Usage

Let's assume your username is 'foo', your password is 'bar' and your security token is 'baz'. Here's then your Java code needed to connect to Salesforce (and a small example query):

```java
UpdateableDataContext dataContext = new SalesforceDataContext("foo", "bar", "baz");
Table accountTable = dataContext.getDefaultSchema().getTableByName("Account");

DataSet dataSet = dataContext.query().from(accountTable).select("Id", "Name").where("BillingCity").eq("New York").execute();

while (dataSet.next()) {
    Row row = dataSet.getRow();
    Object id = row.getValue(0);
    Object name = row.getValue(1);
    System.out.println("Account '" + name + "' from New York has id: " + row.getValue(0);
}
dataSet.close();
```

Connecting to Salesforce.com via MetaModel allows you to treat your Salesforce data just as if it was any other database or data file. Obviously you should be aware that you access remote web services, so there's network latency involved. Therefore we encourage you to spend time figuring out proper queries etc. that will fetch all the data you need in as few roundtrips as possible. That's obviously a rule to apply to any database, but it's even a bit more relevant in the case of remote web services.