

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 23-25 SEPTEMBER 2013

“RESTful transactions” with Apache Sling
Şenol Taş

- My Name is Şenol Taş
- Senior Computer Scientist
- Adobe in Basel, Switzerland



@thelabertasch

Why such a boring Topic?



adaptTo()



adaptTo

Because websites are...

- Serious business
- Make \$\$\$
- Complex
- MUST have
- “Real” Applications

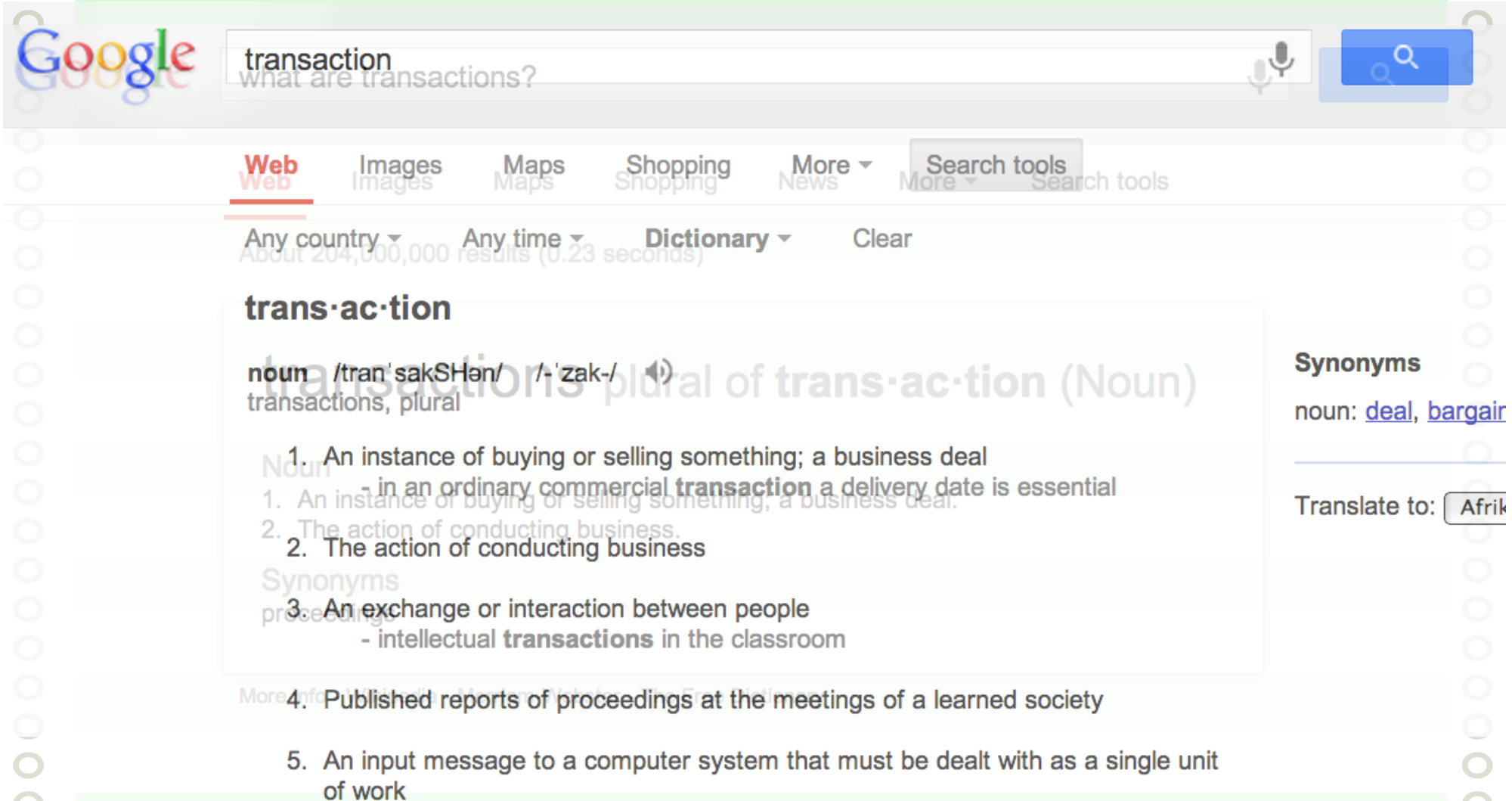
Your Website IS THE Application

Let's....

- Find out “what is a transaction?”
- Quick reminder about REST
- How do these two concepts work together?

What are Transactions?

What are Transactions?




Google transaction what are transactions?

Web Images Maps Shopping More Search tools

Any country Any time Dictionary Clear

About 204,000,000 results (0.23 seconds)

trans·ac·tion

noun /tran'sakSHən/ /-'zak-/ 

transactions, plural

trans·ac·tion (Noun)

1. An instance of buying or selling something; a business deal
- in an ordinary commercial **transaction** a delivery date is essential
2. The action of conducting business.
3. An exchange or interaction between people
- intellectual **transactions** in the classroom
4. Published reports of proceedings at the meetings of a learned society
5. An input message to a computer system that must be dealt with as a single unit of work

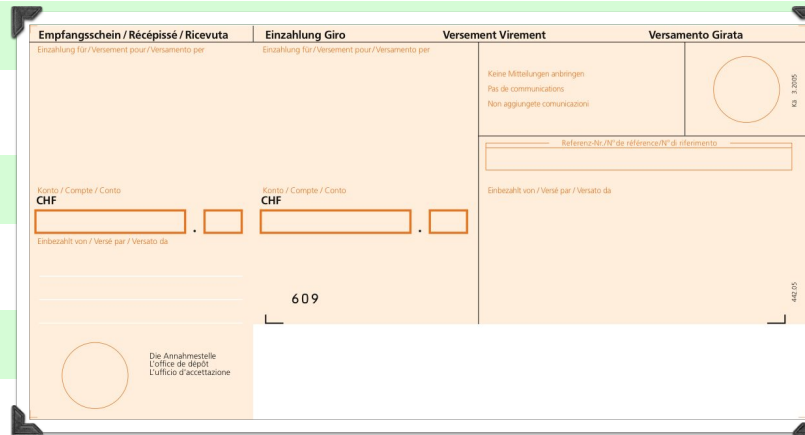
Synonyms

noun: [deal](#), [bargain](#)

Translate to:

“An input message to a computer system
that must be dealt with
as a single unit of work”

What are Transactions?



The image shows a sample Swiss bank check (Einzahlung Giro) with the following fields and text:

- Empfangsschein / Récépissé / Ricevuta** (top left)
- Einzahlung Giro** (top center)
- Versement Virement** (top right)
- Versamento Girata** (top right)
- CHF** (currency symbol)
- 609** (amount)
- Die Annahmestelle / L'office de dépôt / L'ufficio d'accettazione** (bottom left)
- 442 05** (bottom right)

- begin transaction
- debit checking account
- credit savings account
- update history log
- commit transaction

What are Transactions?

- Late 70s: Jim Gray defined the semantic of reliable transactions
- Early 80s: Andreas Reuter and Theo Härder defined the acronym “ACID” in their paper “Principles of Transaction-Oriented Database Recovery”

ACID

- Atomicity
- Consistency
- Isolation
- Durability

How do tools like hybernate help?

```
Session sess = factory.openSession();
Transaction tx = null;
try {
    tx = sess.beginTransaction();

    // do some work
    ...

    tx.commit();
}
catch (RuntimeException e) {
    if (tx != null) tx.rollback();
    throw e; // or display error message
}
finally {
    sess.close();
}
```

What is REST?

What is REST?

- Representational State Transfer
- **Stateless**
- Client-server
- **Cachable**
- Scalable
-

What is REST?

- So? How does this work together with transactions?

REST and transaction is a oxymoron

REST and transactions

- Atomicity
- Consistency
- Isolation
- Durability

What are the challenges?

- HTTP REQUEST – RESPONSE PATTERN
- STATELESS
-?

Change your mindset...

- Not everything has to be done in one request
- If you are using POST.jsp or similar in Sling, most likely something is wrong
- Events and Observations are your friends
- Use the full power of the Sling Default Post and GET Servlet

Remember the cheat sheet from Lars?

Using the SlingPostServlet
this is the default handler for your
POST requests. It can do nearly
anything.

```
<form action="/mynode" method="POST">
  <input type="text" name="title">
  <textarea name="body">
</form>
```

Create or update /mynode, set title and body. Set lastModified and lastModifiedBy automatically

```
<form action="/mynode/" method="POST">
  <input type="text" name="dummy">
  <input type="hidden" name=":order"
    value="first">
</form>
```

Create new node below /mynode and make it the first child (also valid: last, before x, after x, 3, 7, 9).

```
<form action="/node" method="POST">
  <input name=":operation"
    type="hidden" value="delete">
</form>
```

Delete /node

```
<form action="/node" method="POST">
  <input type="hidden"
    name=":operation" value="delete">
  <input type="hidden"
    name=":applyTo" value="/node/one">
  <input type="hidden"
    name=":applyTo" value="/node/two">
</form>
```

Delete /node/one and /node/two

```
<form action="/mynode/" method="POST">
  <input type="hidden"
    name=":name" value="new_node">
  <input type="hidden"
    name=":nameHint" value="new node">
</form>
```

Create new node below /mynode, use name or name hint. Set created and createdBy automatically

```
<input type="text" name="customer">
<input type="hidden" value="John Doe"
  name="customer@DefaultValue" >
<input type="hidden" name="title@Delete">
```

Take default value for customer property, remove the title property

```
<form action="/old/node" method="POST">
  <input type="hidden"
    name=":operation" value="move">
  <input type="hidden"
    name=":dest" value="/new/place">
</form>
```

Move /old/node to /new/place

```
<input type="text" name="date1"
  value="2008-06-13T18:55:00">
<input type="text" name="date2">
<input type="hidden"
  name="date2@TypeHint" value="Date">
<input type="hidden" value="nt:file"
  name="./uploaded/jcr:primaryType">
```

Guess property type from date pattern, set property type explicitly and set node type explicitly

```
<form action="/old/node" method="POST">
  <input type="hidden" name=":operation"
    value="copy">
  <input type="hidden" name=":dest"
    value="/new/place">
  <input type="hidden" name=":replace"
    value="true">
</form>
```

Copy /old/node to /new/place and replace the existing node there.

```
<input type="text" name="oldtitle">
<input type="hidden" value="oldtitle"
  name="newtitle@ValueFrom">
```

Get value for property title from field oldtitle

```
<input type="hidden" value="/node/prop"
  name="title@CopyFrom">
```

Copy property title from other node's property

Example

- Resource: /home/senol/profile
- Begin transaction
- change interests
- Commit transaction

Begin Transaction

- Copy the node /home/senol/profile to /home/senol/transactions/profile/*

```
<form action="/home/senol/transactions/profile/*" method="POST">
  <input type="hidden" name="@CopyFrom" value="/home/senol/profile">
  <input type="hidden" name="status" value="begin">
  <input type="hidden" name=":replace" value="true">
  <input type="hidden" name=":redirect" value="/home/senol/transactions/profile/*.html">
  <input type="submit">
</form>
```

- Update with regular post request

```
<form action="/home/senol/transactions/profile/1_1380000477631" method="POST">  
  <input type="hidden" name="status" value="inprogress">  
  <input type="text" name="interests" value="my interests">  
  <input type="submit">  
</form>
```

Commit transaction

- With observation or events

```
<form action="/home/senol/transactions/profile/1_1380000477631" method="POST">  
  <input type="hidden" name="status" value="commit">  
  <input type="submit">  
</form>
```

- Directly

```
<form action="/home/senol/transactions/profile/1_1380000477631" method="POST">  
  <input type="hidden" name="@CopyFrom" value="/home/senol/profile">  
  <input type="hidden" name="status" value="begin">  
  <input type="hidden" name=":replace" value="true">  
  <input type="hidden" name=":redirect" value="/home/senol/transactions/profile/*.html">  
  <input type="submit" value="save / commit">  
</form>
```

Things of interest

- Communicating with services layer
- Distributed transactions
- Nested transactions
- Locking...
- Using Adobe CQ?
 - Have a look at the Workflow Launcher!

Thank you and have a RESTful Day!