



stORM preview

David Chandler
Android Developer Advocate
blog: turbomanage.com

simple template-based Object Relational Mapping



why



- hand-coded SQL, ugh
- SQLite already has obj-like wrappers
 - ContentValues (insert, update)
 - Cursor (query)
- many apps just require a place to stuff some objects
- existing ORMs use reflection or require up-front modeling

goals

- easy
- convention over configuration
- annotation-driven
 - @Database, @Entity
 - seamless code gen with JDT
- easy to debug
- minimal performance overhead



non-goals

- kitchen sink
- model relations
- absolute max performance



setup

- add storm-api.jar to build path
 - and export! (Order and Export tab)
- add storm-apt.jar to annotation factory classpath



use



- Extend DatabaseHelper, annotate with @Database
- POJOs implement Persistable, annotate with @Entity
- Generates
 - DbFactory
 - EntityTable
 - EntityDao
- `new EntityDao(ctx).insert/get/query...`

use



- `dao.insert(T obj)`
- `dao.insertMany(Iterable<T> objs)`
- `T dao.get(long id)`
- `dao.update(T obj)`
- `dao.delete(long id) / dao.deleteAll()`
- `List<T> dao.listAll()`
- `dao.listAllByExample(T exampleObj)`
- `dao.filter().eq(COL, val).eq(...).exec()`

conventions

- `COLNAME == fieldName.toUpperCase()`
- `id`
 - long id (required)
 - column name is `_ID`



supported types



- all primitives & wrappers
 - boolean, byte, byte[], char, double, float, int, long, short, String
 - byte[] have affinity BLOB
 - boolean, byte, char have affinity INTEGER
- roll-your-own
 - extend `TypeConverter<J,S>`
 - annotate with `@Converter`

nice



- `List<T> dao.asList(Cursor c)`
- `T dao.asObject(Cursor c)`
 - throws `TooManyResultsException`
- `insertMany` uses 1 transaction
- column name enum in `Table` class
 - `filter().eq(Columns.FIRSTNAME, "David")...`
- red squiggles from APT

CSV



- `dao.getDatabaseHelper(ctx)`
- `dbHelper.backupAllTablesToCsv()`
- `dbHelper.restore...FromCsv()`
- exact type conversions
 - blobs are Base64 encoded
 - doubles saved as raw hex values
- file named `dbName.vn.TableName`

UpgradeStrategy



- DROP_CREATE
- BACKUP_RESTORE (csv)
 - dropped cols disappear
 - new cols get default values
 - renaming not yet supported
- UPGRADE
 - override DatabaseHelper.upgrade() and/or
 - override TableHelper.onUpgrade(...) for each

limits

- no relations (yet)
- can't compare doubles or blobs with `FilterBuilder.eq()`
 - likely add `.eq(COL, val, delta)`
 - or `dao.query(String where, String[] args)`



dbFactory

- static db name, version
- singleton instance of DatabaseHelper
- getTableHelpers()



dao

- extends SQLiteDatabase<T>
 - most of the code lives in the base class
- points to DatabaseFactory
- points to TableHelper



table

- all SQL
- all getX() / bindX()
- Cursor --> obj --> ContentValues
- obj --> String[] (for csv)



impl



- how to support incremental compilation?
 - anno processing happens in rounds
 - full src not available in every round
 - stormEnv file under .apt_generated
 - TypeConverters aren't there yet
 - if in doubt, Project | Clean
- Freemarker templates in impl/src/res
- watch the Error Log view

future

- limited relations
- more filter methods `gt()`, `lt()`, etc.
- `@Id` for custom id col
- `@Column(name="custom")`
- part of ADT?



src



drfibonacci.googlecode.com

native http

- Apache HttpClient
- HttpURLConnection
- google-api-java-client
- google-http-java-client
- basic-http-client



basic-http-client



- just the basics
- sync api w/ smart error handling
- async api
 - automatically wraps in AsyncTask
 - auto retry / backoff
- extensible
 - RequestLogger
 - RequestHandler

src



basic-http-client.googlecode.com

R

- developer.android.com
- [Common Tasks](#)
- Google I/O sessions
- [+Android Developers](#)
- turbomanage.com (my blog)

