

JBoss Community

Not Just ORM: Powerful Hibernate ORM Features and Capabilities

Brett Meyer

Senior Software Engineer

Hibernate ORM, Red Hat



HIBERNATE

Brett Meyer

- Hibernate ORM
 - ORM 4 & 5 development
 - Hibernate OSGi
 - Developer community engagement
 - Red Hat support, Hibernate engineering lead
- Other contributions
 - Apache Camel
 - Infinispan
- Contact me
 - @brettemeyer or +brettmeyer
 - Freenode #hibernate or #hibernate-dev (brmeyer)

[github.com/brmeyer
/HibernateDemos](https://github.com/brmeyer/HibernateDemos)

slideshare.net/brmeyer

ORM? JPA?

- ORM: Object/Relational Mapping
 - Persistence: Data objects outlive the JVM app
 - Maps Java POJOs to relational databases
 - Supports OO concepts: inheritance, object identity, etc.
 - Navigate data by walking the object graph, not the explicit relational model
- JPA: Java Persistence API
- Hibernate ORM provides its own native API, in addition to full JPA support
- Annotations and XML

Overview

- Multi-Tenancy
- Value Generation
- Hibernate Spatial
- Hibernate Envers
- Hibernate OSGi
- Hibernate Search
- Hibernate OGM & Validator
- Hibernate Shards
- **Ask questions after each section!**

Multi-Tenancy

Multi-Tenancy

- 1 ORM instance
- Multiple concurrent clients
- Data specific to each tenant
- Legacy
 - Separate SessionFactories for each tenant
 - Application-provided Connections (given when opening a Session)
 - Big schemas or many tenants = huge footprint

Hibernate ORM Multi-Tenancy

- Strategy selection is global
- Tenant identifier provided when opening a Session
- Works with 2LC: identifier used in cache data key
- Custom impls:
 - MultiTenantConnectionProvider: required for physical and schema separated (required)
 - CurrentTenantIdentifierResolver: required for opening a Session without providing the tenant id (explicitly or Session#getCurrentSession)

Hibernate ORM

Multi-Tenancy Strategies

- Physically separated databases
 - 1 JDBC connection pool per tenant
 - Pool selected based on tenant identifier
- Single database, separate schemas
 - Option #1
 - Similar to physically separated
 - 1 JDBC connection pool per tenant
 - Provides schema through the connection URL or pool
 - Option #2:
 - 1 JDBC connection pool using a default schema
 - Each Connection altered with SQL “SET SCHEMA” prior to use

Hibernate ORM

Multi-Tenancy Strategies

- Single database and schema
 - Data partitioned by discriminator value
 - Discriminator complexity varies
 - Each SQL statement altered to include the discriminator
 - Not yet implemented (planned for ORM 5)
 - Alternative: @FilterDef/@Filter w/ a tenantId as a param

Hibernate ORM Multi-Tenancy

- DEMO
- Questions?

Value Generation

Value Generation (Legacy)

- Limited to in-database value generation on insert/update
- Properties marked with @Generated immediately selected

Hibernate ORM Value Generation

- New, expanded support in ORM 4.3
- Supports legacy in-DB generation, but also in-mem
- Create custom annotations!
- ORM provides built in generators
 - @Generated: legacy, providing in-database generation
 - @CreationTimestamp: set only once when owning entity is saved for the first time
 - @UpdateTimestamp: set any time owning entity is saved
 - @GeneratorType: provide custom in-memory generator and define “when”
- DEMO
- Questions?

Hibernate Spatial

Hibernate Spatial

- Currently 3rd party, but pulling in as a new ORM module in 5
- Originally created by Karel Maesen (geovise.com)
- Java Topology Suite (JTS)
 - OpenGIS Simple Feature Spec (SFS) & SQL/MM Spatial (extends SFS)
 - Oracle, Postgres, MySQL, MS SQL, H2, etc. all implement the specs
 - Attempts to provide abstract, cross-platform geo data, but diffs between them
- geolatte-geom
 - Developed/maintained by Karel
 - Improvement over JTS
 - Fully interoperable with JTS
 - Support for lat/lon geographies
 - Additional Dialect differences

Hibernate Spatial (cont'd)

- Properties use JTS Geometry types
 - Point
 - LineString
 - Polygon
 - etc.
- Adds HQL functions for Dialect-supported methods
 - Same functions also implemented as Criteria API Criteria
 - ex: "select e from Event e where within(e.location, :filter) = true" (:filter is a Geometry)
 - <http://www.hibernate.org/hibernatespatial/documentation/03-dialects/01-overview/>

Hibernate Spatial (cont'd)

- Translates between “Well-known text” (WKT) and Geometry types
 - Spatial object markup language
 - 2D and 3D
 - ex: “POINT (30 10)”, “POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))”
- Coordinate system transformations possible in queries
- Does not require JDBC extension drivers (ex: Oracle SDOAPI.jar or Postgres postgis.jar)
- DEMO
- Questions?

Hibernate Envers

Hibernate Envers

- Provides historical versioning and auditing (“SCM for data”)
- Each transaction == a “revision”
- Revision #s are global --> querying for a historical snapshot of the entire database is possible
- Requires
 - @Audited on the entity or individual properties
 - [entity]_AUD tables to store the historical data (DDL created automated if using hbm2ddl, otherwise it's exportable)
- Duplicate data? Yes.
- Powerful queries/capabilities > cheap mem
- Highly configurable

Hibernate Envers (cont'd)

- Revision info
 - Default: simple internal entity stored in REVINFO table
 - Revision # and revision timestamp
 - Override with custom @RevisionEntity & RevisionListener (add other useful fields: IP address, “blame”, etc.)
- Revision entity types
 - Track the entity types that were changed in each revision
 - Disabled by default: requires querying tables for changed data
 - Can be overridden by implementing EntityTrackingRevisionListener on your RevisionEntity

Hibernate Envers (cont'd)

- Revision properties
 - Track the entity properties that were changed in each revision
 - Disabled by default: requires querying tables for changed data
 - Adds boolean columns to audit tables
- Queries
 - Snapshot of entity states at a given revision (horizontal)
 - Revisions at which entities changed (vertical)
- DEMO
- Questions?
- Conditional auditing
 - Default: Envers auditing reacts to Hibernate ORM events
 - Disable `hibernate.listeners.envers.autoRegister`, create your own listeners, wire them in using Integrator

Hibernate OSGi

Hibernate OSGi

- OSGi?
- Provide only the OSGi manifest and hacky workarounds <-- NO!
- Emphasize doing things “the OSGi way”
 - Dynamic environment
 - Scoping
 - Limited visibility into the container
 - No scanning **all** bundles
 - Reduced conflicts, allows concurrent instances, etc.
 - OSGi services
- Isolated in hibernate-osgi module (no pervasive dependencies or OSGi code)

Hibernate OSGi (cont'd)

- 3 supported environments
 - Enterprise OSGi Managed JPA
 - Container discovers and manages persistence units and EntityManagerFactories
 - Similar to many app servers, Spring, etc.
 - Ex: Apache Aries JPA
 - Un-managed JPA
 - Direct use of hibernate-entitymanager
 - EntityManagerFactories created through OSGi services
 - Native
 - Direct use of hibernate-core
 - SessionFactories created through OSGi services
- Quickstarts
- Questions?

Hibernate Search

Hibernate Search

- Full-text search on the DB
 - Bad performance
 - CPU/IO overhead
- Offload full-text queries to Hibernate Search engine
 - Fully indexed
 - Horizontally scalable
- Based on Apache Lucene
- “Google for your entities”

Hibernate Search (cont'd)

- Annotate entities with `@Indexed`
- Annotate properties with `@Field`
 - Index the text: `index=Index.YES`
 - “Analyze” the text: `analyze=Analyze.YES`
 - Lucene analyzer
 - Chunks sentences into words
 - Lowercase all of them
 - Exclude common words (“a”, “the”)
 - Stemming

Hibernate OGM & Hibernate Validator

Hibernate OGM & Validator

- Hibernate OGM:
 - ORM/JPA support for NoSQL
 - Infinispan, EHCache, MongoDB, Neo4j
- Hibernate Validator
 - Bean Validation impl, but extended
 - Both annotation and XML based
 - @NotNull, @Size(min = 2, max = 14), @Min(2), etc.

Hibernate Shards

Hibernate Shards

- Started as a Google team's 20% project
- Horizontal partitioning across multiple databases
- Flexible sharding strategies, both provided and custom
- Supports virtual shards: simplifies re-sharding
- Typical Hibernate ORM usage: HQL, Criteria, etc.
- **Last supported ORM version: 3.6.x**
- **Currently has upgrade momentum – contact me if interested!**

How to Help:

hibernate.org
/orm/contribute

Hibernate ORM: Tips, Tricks, and Performance Techniques

**Tomorrow, 1pm,
Ballroom D**

QUESTIONS?

- Q&A
- #hibernate or #hibernate-dev (brmeyer)
- @brettemeyer
- +brettmeyer