Intro to RavenDB

Oren Eini aka Ayende Rahien
ayende@ayende.com
http://ayende.com/blog
What?

- Document Database
- .NET
- Client API:
  - REST
  - JavaScript
  - .NET
- License
  - OSS – AGPL
  - Commercial
Why?

- Windows/.NET deserves some NoSQL goodness too 😊.
- Disagreement with some technical choices.
- Different requirement set from .NET community.
NoSQL from outside...

---

**Fault-tolerance**

So, how do I query the database?
It's not a database. It's a key-value store!

Ok, it's not a database.
How do I query it?

You write a distributed map reduce function in Erlang!

Did you just tell me to go fuck myself?
I believe I did, Bob.
Linq

- JavaScript isn’t a good language for queries
  - MongoDB
    - `db.users.find({joinedAt : {$lt: lastMonth}});`
  - CouchDB
    - `function(doc) {
        if (doc.Type == "User") {
            emit(null, {joinedAt: doc.joinedAt});
        }
    }
  - .NET has built-in query language
    - `from user in docs.Users
      where user.JoinedAt > lastMonth
      select user;`
Document & Metadata

- Document contains data
- Metadata describe document externally
- **Very useful** for extensions
Collections

- Document can belong to a collection (Posts, Orders, Users, etc).
- Mostly used for indexing.
Transactions

- Single document
- Multi document
- Multi node
Indexes

- Linq syntax
- Map / Reduce
- Stale reads
- Indexing happens on the background
- Indexing on top of Lucene
- Full text search!

```javascript
from post in docs.Posts
select category in post.Categories
select new {category}
```
Querying

- By index
- Dynamic queries
- Linear query
Client API (.NET)

- Unit of Work
- Based on the Hibernate API
- Linq support
Sharding

- Client driven
- Out of the box with the .NET client API
- Based on the Hibernate Shards design
Replication

- Master --> Slave
- Master <-> Master

- Async background replication
- Automatic failover
Replicate to SQL

- Reporting
- Data replication
Spatial queries

```csharp
from r in docs.Restaurants
select new {
    r.Rating,
    _ = SpatialIndex.Generate(r.Lat, r.Lng)
}

session.LuceneQuery<Restaurant>("Restaurants ")
    .WhereGreaterThanOrEquals("Rating", 4)
    .WithinRadiusOf(radius: 5, latitude: 38.91, longitude: -77.39)
```
Compiled indexes

- Complex logic
- Event sourcing
Set based operations

- DELETE 
  http://localhost:8080/bulk_docs/UsersByActivityStatus?query=IsActive:False

- PATCH 
  http://localhost:8080/bulk_docs/UsersByLastLoginDate?query=LastLoginDate:[NULL TO 20100527]
  
  [ { "Type": "Set", "Name": "IsActive", "Value": false } ]

- UPDATE 
  Users SET IsActive = 0 WHERE LastLoginDate < '2010-05-27'
Extensibility

- Core design principle
- RavenDB built on top of decoupled components
Customization

- One size does *NOT* fits all!
- Intended to be customized per application needs.
More...

- **Document authorization:**
  - Role based
  - Document based
  - Tag based

- **Versioning**
  - Snapshot on every change

- **Expiration**
  - Delete document after time $T$
One last cool feature...

Includes
Deployment options

- Embedded (desktop/mobile apps)
- Executable
- Service
- IIS
Questions?

- Where to find RavenDB?
- http://ravendb.net
- http://groups.google.com/group/ravendb/