

How to Use NoSQL in Enterprise Java Applications

Patrick Baumgartner

Agenda

- Speaker Profile
- New Demands on Data Access
- New Types of Data Stores
- Integrating NoSQL Data Stores
- Spring Data Overview
- Example with MongoDB
- Example with Neo4j
- Q & A

Speaker Profile

Patrick Baumgartner

- Senior Software Consultant | Partner
- VMware/SpringSource Certified Instructor (Spring Trainer)
- Spring Framework, OSGi & agile engineering practices
- Co-author of „OSGi für Praktiker“ (Hanser)

Swiftmind GmbH <http://www.swiftmind.com>

- Enterprise Java, Spring & OSGi consulting
- Spring & OSGi workshops & trainings
- Agile engineering practices workshops

New Demands on Data Access



- Structured and unstructured data
- Massive amounts of data
- Inexpensive horizontal scaling
- Apps and data in the cloud
- Social network features
- ...

New Types of Data Stores

Object HTTP Blob
Key-Value
Relational
BigData Graph
Document Column

Integrating NoSQL Data Stores #1

We are not architects!



<http://www.flickr.com/photos/sakeeb/4087246274>

Integrating NoSQL Data Stores #2

Don't re-invent the wheel!



<http://www.flickr.com/photos/dmott9/5921728819>

Integrating NoSQL Data Stores #3

Let's use Spring!



<http://www.springsource.org/spring-data>

Spring Data

- Same goals as the Spring Framework
 - Productivity improvement
 - Programming model consistency
 - Portability
- Umbrella for subprojects, specific to a given database
- Mapping support for Java domain objects

- <http://www.springsource.org/spring-data>
- <http://github.com/SpringSource/spring-data-XXX>



Spring Data – Overview #1

Relational Databases

- JPA
- JDBC Extensions

Big Data

- Apache Hadoop

Data Grid

- GemFire

HTTP

- REST

Spring Data – Overview #2

Key Value Stores

- Redis

Document Stores

- Mongo DB

Graph Databases

- Neo4J

Column Stores

- HBase

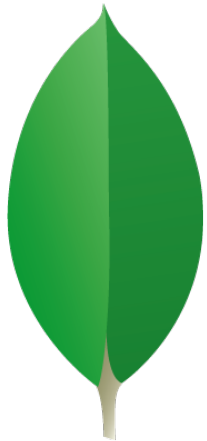
Common Infrastructure

- Commons

Spring Data – Key Features

- Low level data access API abstraction
 - MongoTemplate, RiakTemplate, Neo4jTemplate
 - Exception translation
 - Transaction management
- Object Mapping (Java to data store)
- Generic Repository Support
 - Basic CRUD, dynamic finders, pagination and sorting
- Spring namespaces
- Cross Store Persistence Programming Model
 - @Entity, @Document, @NodeEntity

Spring Data MongoDB – Example



mongoDB

Documents



Spring Data MongoDB – Document

@Document

```
public class Person {  
    @Id  
    private int id;  
    private String name;  
    private int age;  
    // getters/setters...  
}
```

Stored JSON:

```
{ "_id" : ObjectId("4f9886290364b533b3acd4ce"),  
  "_class" : "com.example.Person",  
  "name" : "Bob",  
  "age" : 33  
}
```

Spring Data MongoDB – Configuration

```
<bean id="mongoTemplate" class="org.springframework.data.document.mongodb.MongoTemplate">
  <constructor-arg name="mongo" ref="mongo"/>
  <constructor-arg name="databaseName" value="test"/>
  <constructor-arg name="defaultCollectionName" value="HelloMongo"/>
</bean>
```

```
<!-- Factory bean that creates the Mongo instance -->
<bean id="mongo" class="org.springframework.data.document.mongodb.MongoFactoryBean">
  <property name="host" value="localhost"/>
</bean>
```


Spring Data MongoDB – Repository

@Repository

```
public class MongoPersonRepository implements BaseRepository<Person> {
```

@Autowired

```
MongoOperations mongoTemplate;
```

```
Person createPerson(String name, int age){  
    if(!mongoTemplate.collectionExists(Person.class)){  
        mongoTemplate.createCollection(Person.class);  
    }  
    Person p = new Person(name, age);  
    mongoTemplate.insert(p)  
    return p;  
}
```

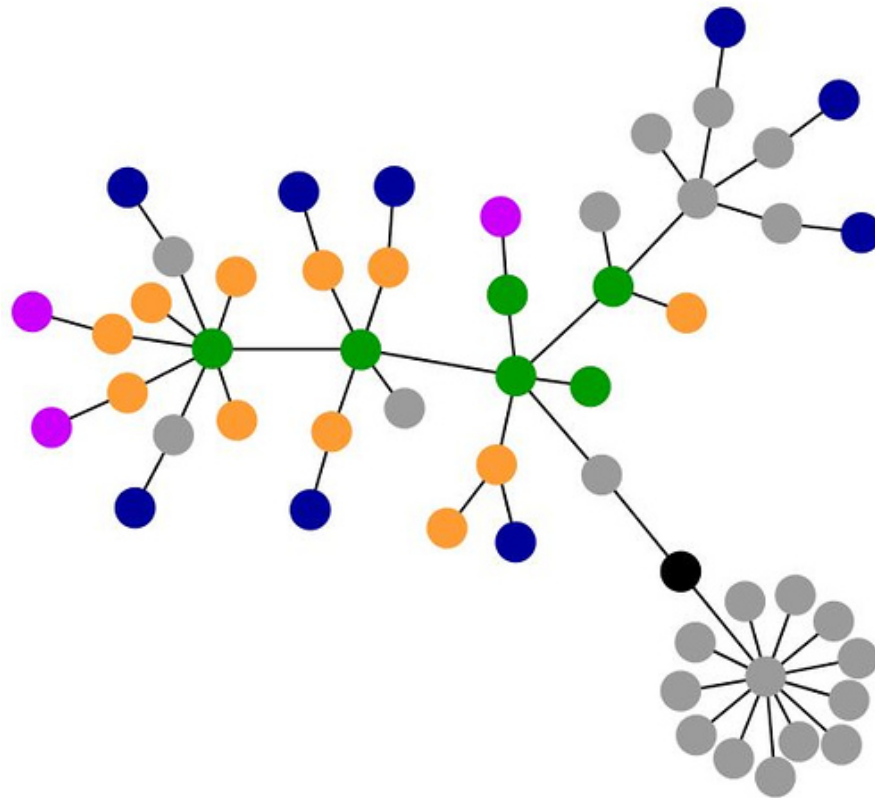
```
...
```

```
}
```

Spring Data Neo4j – Example



Graph



Spring Data Neo4j (SDN)

- POJOs mapped as nodes or relationships – type safe
- Works directly Database, typically embedded mode
- Data is fetched very fast and lazy
- Stores everything within a @Transaction
- Uses heavily AspectJ magic to enhance the POJOs
- ...

Spring Data Neo4j – Entity

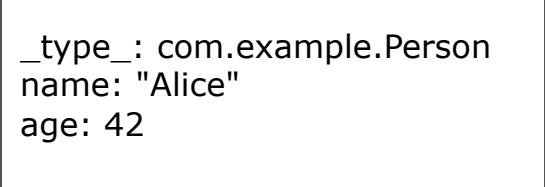
@NodeEntity

```
public class Person {  
    private String name;  
    private int age;  
    // getters/setters...  
}
```

```
Person alice = new Person();  
alice.setName("Alice");  
alice.setAge(42);  
alice.persist();
```



Node

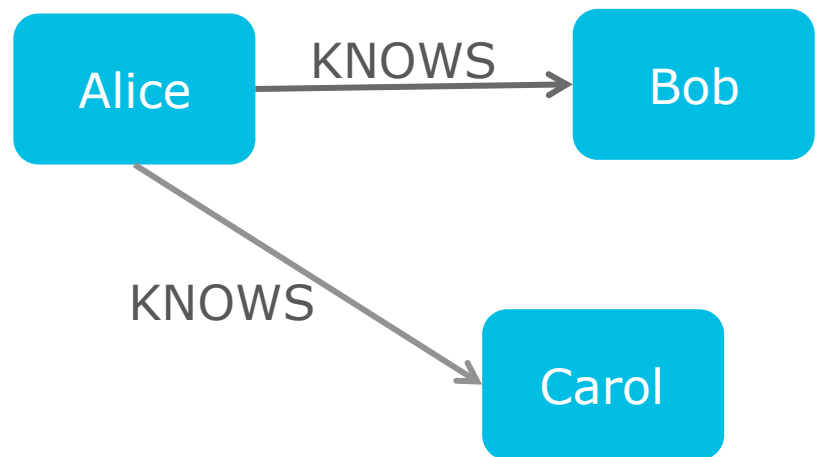


```
{  
  "_type_": com.example.Person  
  name: "Alice"  
  age: 42  
}
```

Spring Data Neo4j – NodeEntity

```
@NodeEntity
public class Person {
    private String name;
    private int yearOfBirth;
    @RelatedTo(type = "KNOWS", direction = Direction.OUTGOING)
    private Set<Person> knownPersons;

    public void knows(Person p) {
        knownPersons.add(p);
    }
    public Set<Person> getFriends() {
        return knownPersons;
    }
}
Person alice = ...;
alice.knows(bob);
alice.knows(carol);
```



Spring Data Neo4j – Relationship

@RelationshipEntity

```
public class Knows {  
    private int sinceYear;  
    public Knows since(int year) {  
        this.sinceYear = year;  
        return this;  
    }  
}
```

@NodeEntity

```
public class Person {  
    public Known knows(Person p) {  
        return this.relateTo(p, Knows.class, "KNOWS");  
    }  
}
```

```
Person alice = ...;
```

```
Person bob ...;
```

```
alice.knows(bob).since(2012);
```



Spring Data Neo4j – Repository

```
public interface PersonRepository extends
GraphRepository<Person> {
    @Query("start person = {0} match ... return ...")
    Iterable<Product> getOwnedServices(Person person);
    Iterable<Person> findByName(String name);
    Iterable<Person> findByNameLike(String name);
}
```

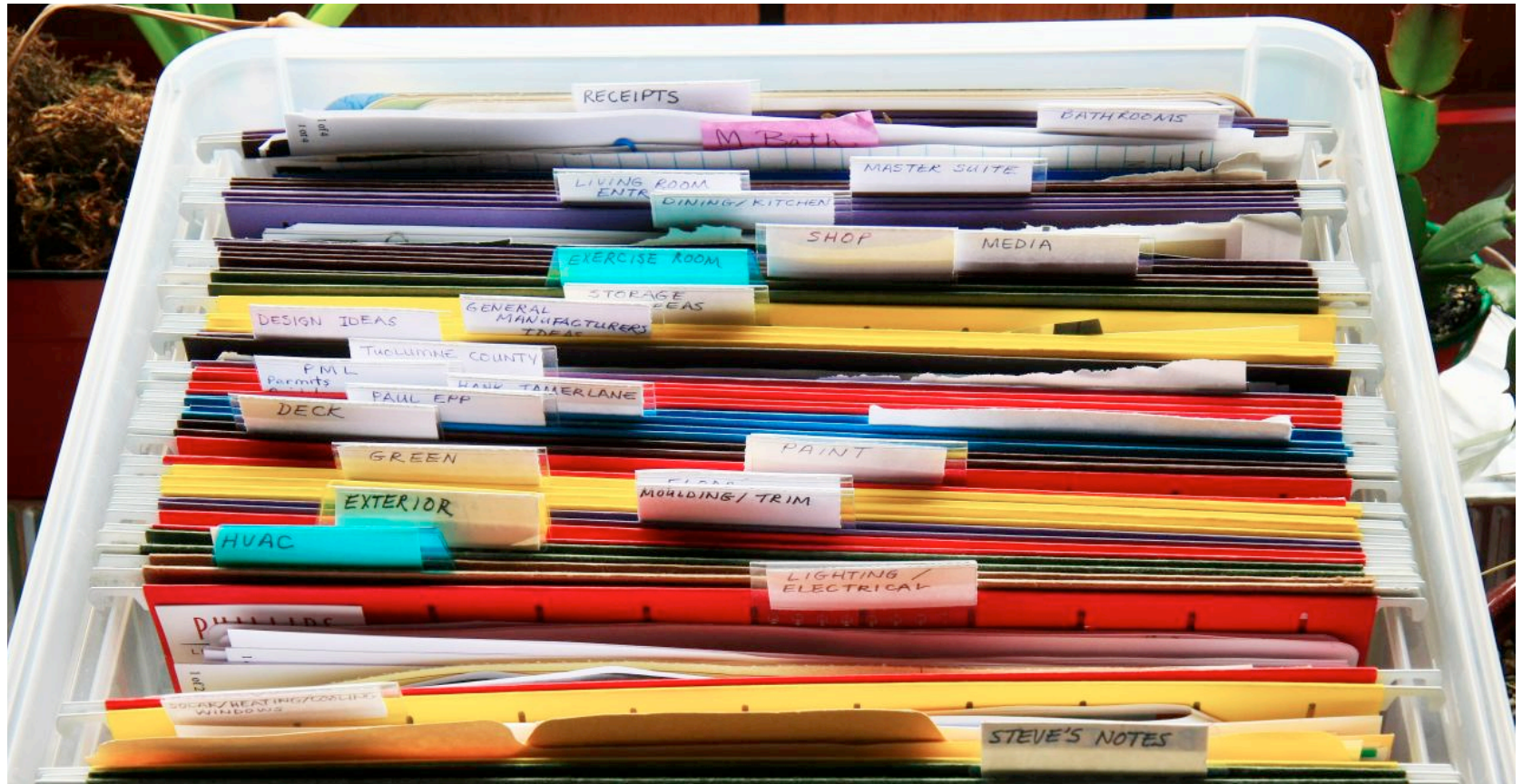
```
@Autowired
PersonRepository personRepository;
Person alice = personRepository.findByName("Alice");
```


Spring Data Neo4j – Querying

- Several possibilities implemented to query the graph
 - Neo4j API
 - Traversal descriptions
 - Graph algorithms
 - Index queries
 - Cypher queries

Give it a try!

Use a data model which really matches to your data ...



<http://www.flickr.com/photos/juniorvelo/3267647833>

05.09.2012 /ch/open Workshop-Tage

NoSQL für Java Enterprise Anwendungen mit Spring Data



<http://www.flickr.com/photos/4nitsirk/5211251578>

Q & A

Patrick Baumgartner

patrick.baumgartner [at] swiftmind [dot] com

<http://www.swiftmind.com> <http://www.twitter.com/patbaumgartner>