


MongoDB support for Spring Data source code

- [Supported Libraries](#)
- [Supported Operations](#)
- [Objects](#)
- [Links](#)
- [What results can you expect](#)
 - [Creation of database object](#)
 - [With application.properties](#)
 - [With Java configuration file](#)
 - [Insert Operation](#)
 - [Update Operation](#)
 - [Delete Operation](#)
 - [Select Operation](#)
 - [Query Methods](#)
- [Evolution](#)
- [Limitations](#)

 CAST supports **MongoDB** via its **NoSQL for Java** extension. Details about how this support is provided for **Java with Spring Data source code** is discussed below.

Supported Libraries



Supported till version 3.2.x



Supported Operations

For Storing the data	<ul style="list-style-type: none">• <ul style="list-style-type: none">• <code>org.springframework.data.repository.CrudRepository.save</code>• <code>org.springframework.data.repository.CrudRepository.saveAll</code>• <code>org.springframework.data.repository.ReactiveCrudRepository.save</code>• <code>org.springframework.data.repository.ReactiveCrudRepository.saveAll</code>• <code>org.springframework.data.repository.MongoRepository.insert</code>• <code>org.springframework.data.repository.ReactiveMongoRepository.insert</code>• <code>org.springframework.data.mongodb.core.MongoTemplate.insert</code>• <code>org.springframework.data.mongodb.core.MongoTemplate.insertAll</code>• <code>org.springframework.data.mongodb.core.MongoTemplate.save</code>• <code>org.springframework.data.mongodb.core.MongoTemplate.upsert</code>• <code>org.springframework.data.mongodb.core.BulkOperations.upsert</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoTemplate.insert</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoTemplate.insertAll</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoTemplate.save</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoTemplate.upsert</code>• <code>org.springframework.data.mongodb.core.MongoOperations.insert</code>• <code>org.springframework.data.mongodb.core.MongoOperations.insertAll</code>• <code>org.springframework.data.mongodb.core.MongoOperations.save</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoOperations.insert</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoOperations.insertAll</code>• <code>org.springframework.data.mongodb.core.ReactiveMongoOperations.upsert</code>
For Fetching the data	<ul style="list-style-type: none">• <ul style="list-style-type: none">• <code>org.springframework.data.repository.CrudRepository.findAll</code>• <code>org.springframework.data.repository.CrudRepository.count</code>• <code>org.springframework.data.repository.CrudRepository.findById</code>• <code>org.springframework.data.repository.CrudRepository.findAllById</code>• <code>org.springframework.data.repository.CrudRepository.existsById</code>• <code>org.springframework.data.repository.PagingAndSortingRepository.findAll</code>• <code>org.springframework.data.repository.QueryByExampleExecutor.count</code>• <code>org.springframework.data.repository.QueryByExampleExecutor.exists</code>• <code>org.springframework.data.repository.QueryByExampleExecutor.findAll</code>• <code>org.springframework.data.repository.QueryByExampleExecutor.findOne</code>• <code>org.springframework.data.repository.ReactiveCrudRepository.findAll</code>• <code>org.springframework.data.repository.ReactiveCrudRepository.count</code>• <code>org.springframework.data.repository.ReactiveCrudRepository.findById</code>• <code>org.springframework.data.repository.ReactiveCrudRepository.findAllById</code>

	<ul style="list-style-type: none"> • org.springframework.data.mongodb.core.ReactiveMongoOperations.group
For Removing the data	<ul style="list-style-type: none"> • org.springframework.data.repository.CrudRepository.deleteAll • org.springframework.data.repository.CrudRepository.deleteById • org.springframework.data.repository.CrudRepository.delete • org.springframework.data.repository.ReactiveCrudRepository.deleteAll • org.springframework.data.repository.ReactiveCrudRepository.deleteById • org.springframework.data.repository.ReactiveCrudRepository.delete • org.springframework.data.mongodb.core.MongoTemplate.findAndRemove • org.springframework.data.mongodb.core.MongoTemplate.findAllAndRemove • org.springframework.data.mongodb.core.MongoTemplate.dropCollection • org.springframework.data.mongodb.core.BulkOperations.remove • org.springframework.data.mongodb.core.MongoTemplate.remove • org.springframework.data.mongodb.core.ReactiveMongoTemplate.remove • org.springframework.data.mongodb.core.ReactiveMongoTemplate.findAndRemove • org.springframework.data.mongodb.core.ReactiveMongoTemplate.findAllAndRemove • org.springframework.data.mongodb.core.ReactiveMongoTemplate.dropCollection • org.springframework.data.mongodb.core.ReactiveBulkOperations.remove • org.springframework.data.mongodb.core.MongoOperations.dropCollection • org.springframework.data.mongodb.core.MongoOperations.findAndRemove • org.springframework.data.mongodb.core.MongoOperations.findAllAndRemove • org.springframework.data.mongodb.core.MongoOperations.remove • org.springframework.data.mongodb.core.ReactiveMongoOperations.dropCollection • org.springframework.data.mongodb.core.ReactiveMongoOperations.findAndRemove • org.springframework.data.mongodb.core.ReactiveMongoOperations.findAllAndRemove • org.springframework.data.mongodb.core.ReactiveMongoOperations.remove
For Updating the data	<ul style="list-style-type: none"> • org.springframework.data.mongodb.core.MongoTemplate.upsert • org.springframework.data.mongodb.core.MongoTemplate.update • org.springframework.data.mongodb.core.MongoTemplate.updateFirst • org.springframework.data.mongodb.core.MongoTemplate.updateMulti • org.springframework.data.mongodb.core.MongoTemplate.findAndModify • org.springframework.data.mongodb.core.MongoTemplate.findAndReplace • org.springframework.data.mongodb.core.BulkOperations.updateOne • org.springframework.data.mongodb.core.BulkOperations.updateMulti • org.springframework.data.mongodb.core.BulkOperations.upsert • org.springframework.data.mongodb.core.ReactiveMongoTemplate.upsert • org.springframework.data.mongodb.core.ReactiveMongoTemplate.update • org.springframework.data.mongodb.core.ReactiveMongoTemplate.updateFirst • org.springframework.data.mongodb.core.ReactiveMongoTemplate.updateMulti • org.springframework.data.mongodb.core.ReactiveMongoTemplate.findAndModify • org.springframework.data.mongodb.core.ReactiveMongoTemplate.findAndReplace • org.springframework.data.mongodb.core.ReactiveBulkOperations.updateOne • org.springframework.data.mongodb.core.ReactiveBulkOperations.updateMulti • org.springframework.data.mongodb.core.ReactiveBulkOperations.upsert • org.springframework.data.mongodb.core.MongoOperations.findAndModify • org.springframework.data.mongodb.core.MongoOperations.findAndReplace • org.springframework.data.mongodb.core.MongoOperations.upsert • org.springframework.data.mongodb.core.MongoOperations.updateFirst • org.springframework.data.mongodb.core.MongoOperations.updateMulti • org.springframework.data.mongodb.core.ReactiveMongoOperations.findAndModify • org.springframework.data.mongodb.core.ReactiveMongoOperations.findAndReplace • org.springframework.data.mongodb.core.ReactiveMongoOperations.upsert • org.springframework.data.mongodb.core.ReactiveMongoOperations.updateFirst • org.springframework.data.mongodb.core.ReactiveMongoOperations.updateMulti

Objects

Icon	Description
	Java MongoDB database
	Java MongoDB collection

	Java unknown MongoDB database
	Java unknown MongoDB collection

Links

Link type	Source and destination of link	Methods supported
parentLink	Between Mongo Objects (Collection Database Project)	
useLink	Between the caller Spring Data Java Method objects and Mongo Collection Object	<ul style="list-style-type: none"> mapReduce
useSelectLink		<ul style="list-style-type: none"> find findAll findById findAllById findAndRemove findAllAndRemove findAndReplace findAndModify findOne stream group geoNear aggregate aggregateStream aggregateAndReturn executeQuery getCollection findDistinct
useUpdateLink		<ul style="list-style-type: none"> upsert update updateFirst updateOne updateMulti findAndModify findAndReplace
useDeleteLink		<ul style="list-style-type: none"> deleteAll deleteById delete findAndRemove dropCollection remove findAllAndRemove
useInsertLink		<ul style="list-style-type: none"> save saveAll insert insertAll upsert

What results can you expect

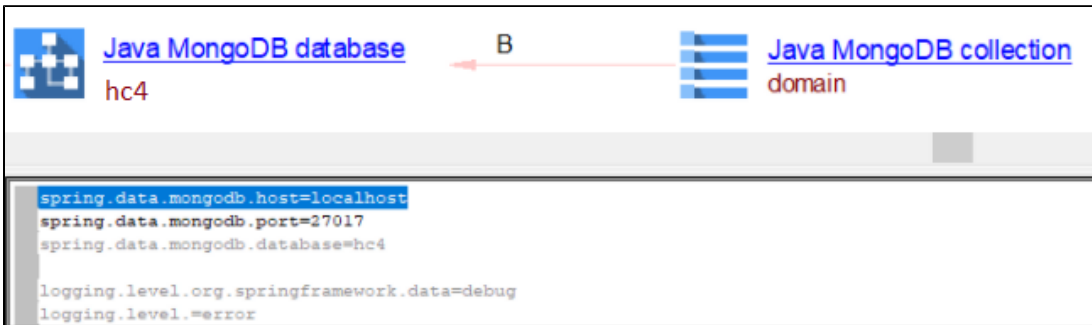
Once the analysis/snapshot generation has completed, you can view the results in the normal manner (for example via CAST Enlighten). Some examples are shown below.

Creation of database object

With application.properties

```
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=hc4

logging.level.org.springframework.data=debug
logging.level.=error
```



With Java configuration file

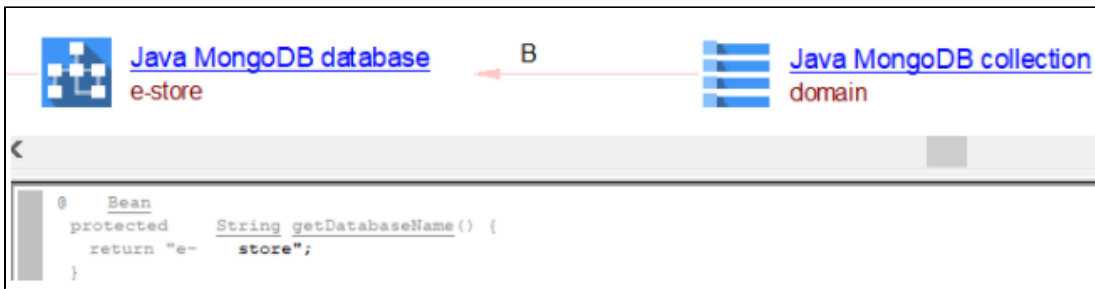
```
package com.mkyong;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.mongodb.config.AbstractMongoConfiguration;
import org.springframework.data.mongodb.repository.config.EnableMongoRepositories;

import com.mongodb.Mongo;
import com.mongodb.MongoClient;

@Configuration
@EnableMongoRepositories
public class ApplicationConfig extends AbstractMongoConfiguration{
    @Override
    @Bean
    protected String getDatabaseName() {
        return "e-store";
    }

    @Override
    protected String getMappingBasePackage() {
        return "com.oreilly.springdata.mongodb";
    }

    @Override
    @Bean
    public Mongo mongo() throws Exception {
        // TODO Auto-generated method stub
        return new MongoClient("localhost");
    }
}
```



Insert Operation

```

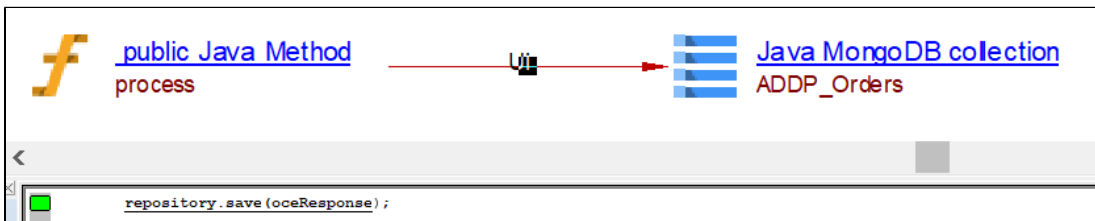
@Component("storeOrder")
public class StoreOCEOrder implements Processor {

    private static final Logger log = LoggerFactory.getLogger(StoreOCEOrder.class);

    @Autowired
    private OceOrderRepository repository;

    public void process(Exchange exg) throws Exception {
        repository.save(oceResponse);
    }
}

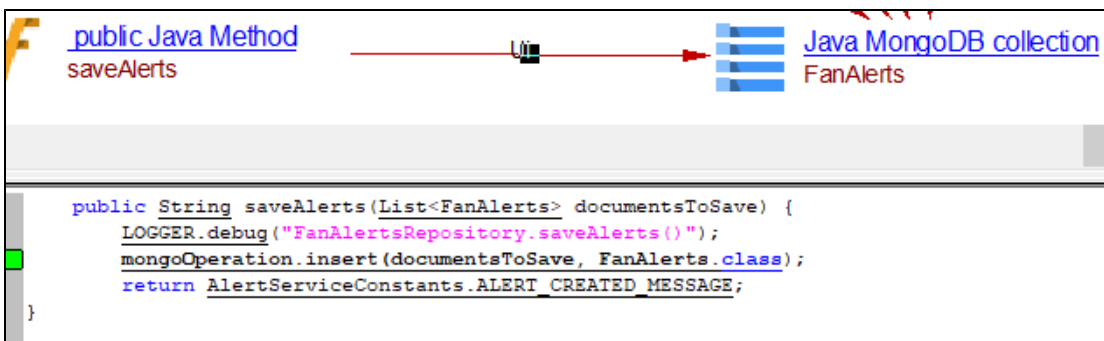
```



```

public String saveAlerts(List<FanAlerts> documentsToSave) {
    LOGGER.debug("FanAlertsRepository.saveAlerts()");
    mongoOperation.insert(documentsToSave, FanAlerts.class);
    return AlertServiceConstants.ALERT_CREATED_MESSAGE;
}

```

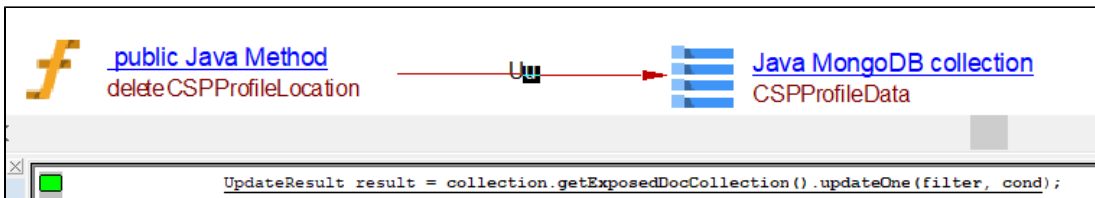


Update Operation

```

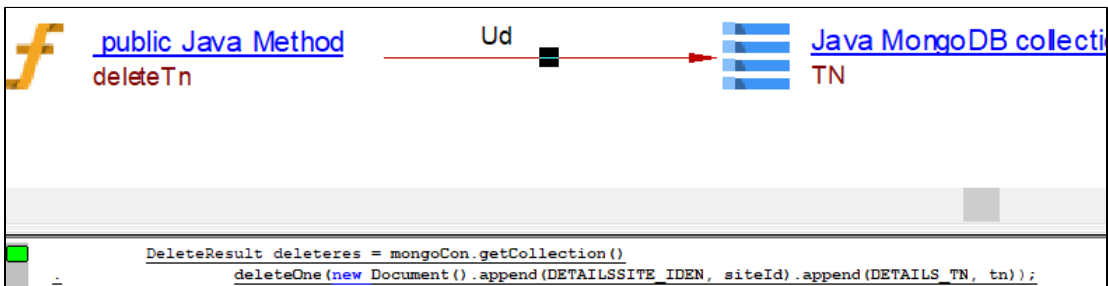
public Response deleteCSPPProfileLocation(CSPPProfileDeleteLocationRequest req) {
    UpdateResult result = collection.getExposedDocCollection().updateOne(filter, cond);
}

```



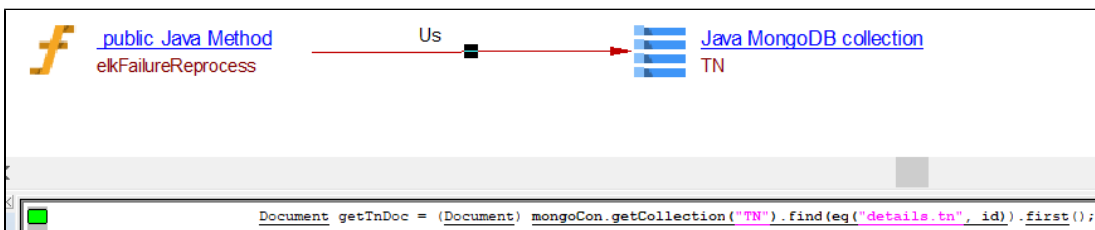
Delete Operation

```
public Response deleteTn(String siteId, String tn) {
    DeleteResult deleteres = mongoCon.getCollection()
        .deleteOne(new Document().append(DETAILSSITE_IDEN, siteId).append
(DETAILS_TN, tn));
}
```



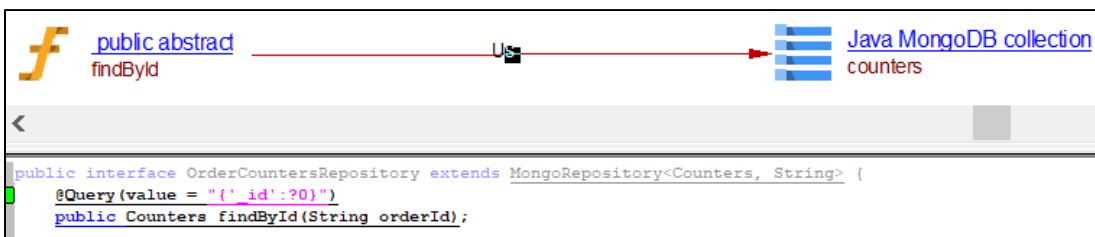
Select Operation

```
public void elkFailureReprocess() {
    Document getTnDoc = (Document) mongoCon.getCollection("TN").find(eq("details.tn", id)).first();
}
```



Query Methods

```
public interface OrderCountersRepository extends MongoRepository<Counters, String> {
    @Query(value = "{ '_id':?0}")
    public Counters findById(String orderId);
}
```



Evolution

- Query methods with and without @Query annotation are supported.
- Support for Reactive Mongo Repository
- Support for Reactive Mongo Template and Reactive Mongo Operations
- Support for APIs belonging to Mongo Collection of `com.mongodb.client.MongoCollection`
- No connection objects are created. Only databases and collection objects are created.

Limitations

- Resolution of Database and Collection is limited, "Unknown" is used when not resolved.
- Query method results in link between query method and repository collection (domain entity) and not between the actual caller method and repository collection