DynamoDB support for Spring Data source code

CAST supports **DynamoDB** via its NoSQL for Java extension. Details about the support provided for **Java with Spring Data source code** is explained below.

Supported Client Libraries

DynamoDBCrudRepository	✓
DynamoDBPagingAndSortingRepository	✓

Supported Operations

Operations	Method Supported
Insert	• save • saveAll
Select	• existsById • findById • findAll
Delete	• delete • deleteById • deleteAllById • deleteAll

Objects

Icon	Description
	Java_DynamoDB_Client
	Java_DynamoDB_Table
	Java_Unknown_DynamoDB_Client
	Java_Unknown_DynamoDB_Table

Links

Links are created for transaction and function point needs:

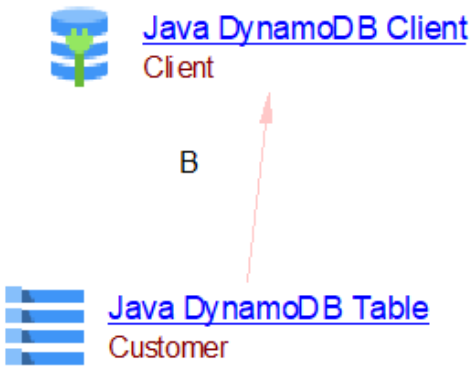
Link type	Source and destination of link	Methods Supported
parentLink	Between DynamoDB client object and DynamoDB table	
useInsertLink	Between the caller Java Method objects and DynamoDB client	• save • saveAll
useSelectLink		• existsById • findById • findAll
useDeleteLink		• delete • deleteById • deleteAll • deleteAllById

What results can you expect?

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

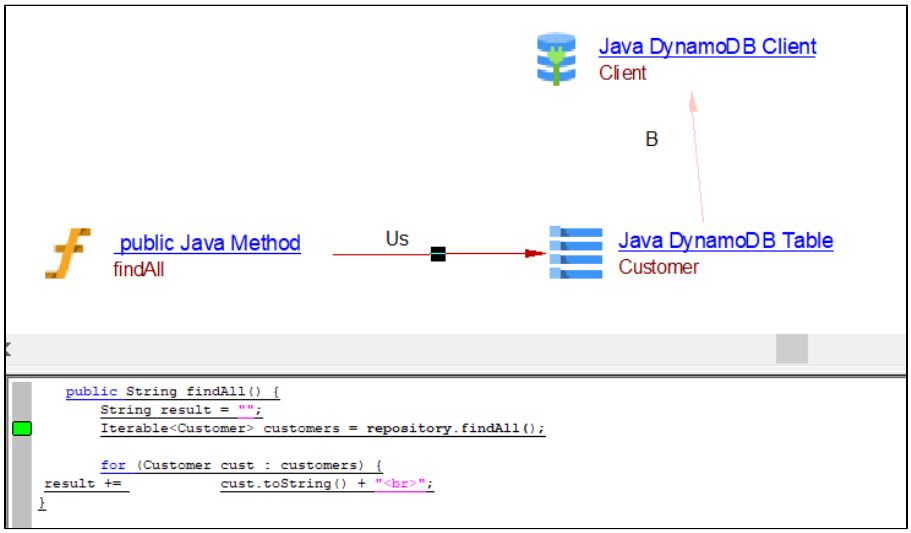
DynamoDB Client with Java Configuration

```
@Configuration
@EnableDynamoDBRepositories(basePackages = "com.javasampleapproach.dynamodb.repo")
public class DynamoDBConfig {

        @Value("${amazon.dynamodb.endpoint}")
        private String dBEndpoint;

        @Value("${amazon.aws.accesskey}")
        private String accessKey;

        @Value("${amazon.aws.secretkey}")
        private String secretKey;

        @Bean
        public AmazonDynamoDB amazonDynamoDB() {
                AmazonDynamoDB dynamoDB = new AmazonDynamoDBClient(amazonAWSCredentials());

                if (!StringUtils.isNullOrEmpty(dBEndpoint)) {
                        dynamoDB.setEndpoint(dBEndpoint);
                }

                return dynamoDB;
        }

        @Bean
        public AWSCredentials amazonAWSCredentials() {
                return new BasicAWSCredentials(accessKey, secretKey);
        }
}
```



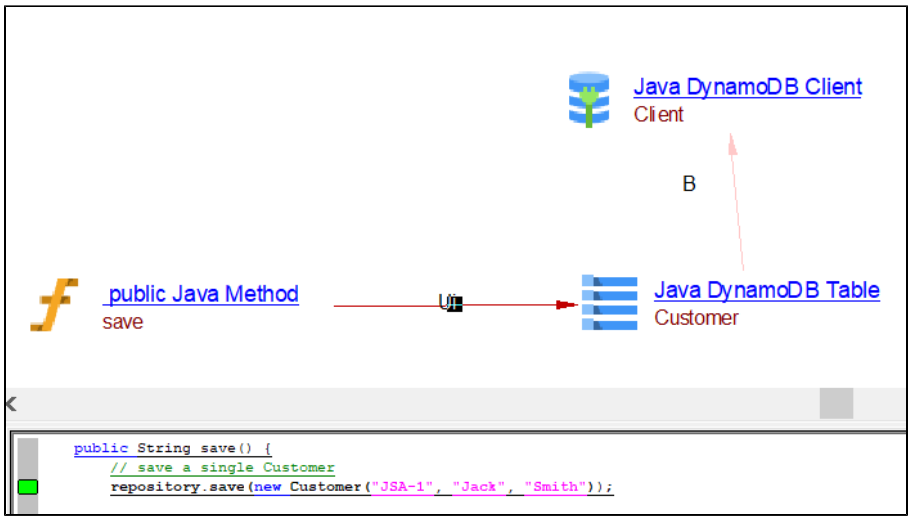
Select Operation

```
public String findAll() {
                String result = "";
                Iterable<Customer> customers = repository.findAll();

                for (Customer cust : customers) {
                        result += cust.toString() + "<br>";
                }

                return result;
        }
```



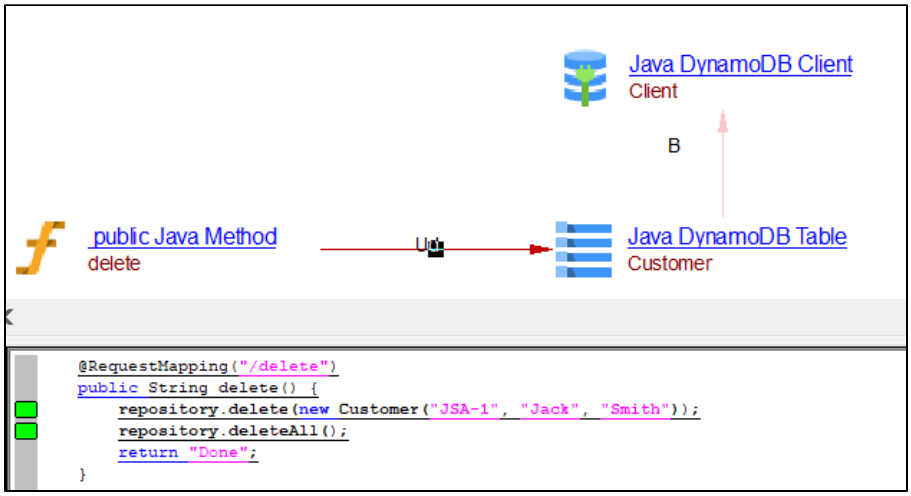
Insert Operation

```
public String save() {
                // save a single Customer
                repository.save(new Customer("JSA-1", "Jack", "Smith"));

}
```



Delete Operation

```
public String delete() {
                repository.delete(new Customer("JSA-1", "Jack", "Smith"));
                repository.deleteAll();
                return "Done";
        }
```



Evolution

- Better resolution for tables
- Query Methods are supported
- Query methods with @Query annotation are supported

Limitations

- Client is created as unknown, if the name is not retrieved from the properties file or if the name could not be resolved.