

Building Microservices with **Spring Data Cassandra** and **Spring Boot**

Special GUEST: Mark Paluch

- Spring Data Project Lead at Pivotal
- Lead of the Lettuce Redis driver
- Reactive data integrations and R2DBC
- Spring R2DBC
- Spring Data Redis
- Spring Data Cassandra
- Trance Producer **#RFLKTD**



@mp911de

Developers

 Subscribe



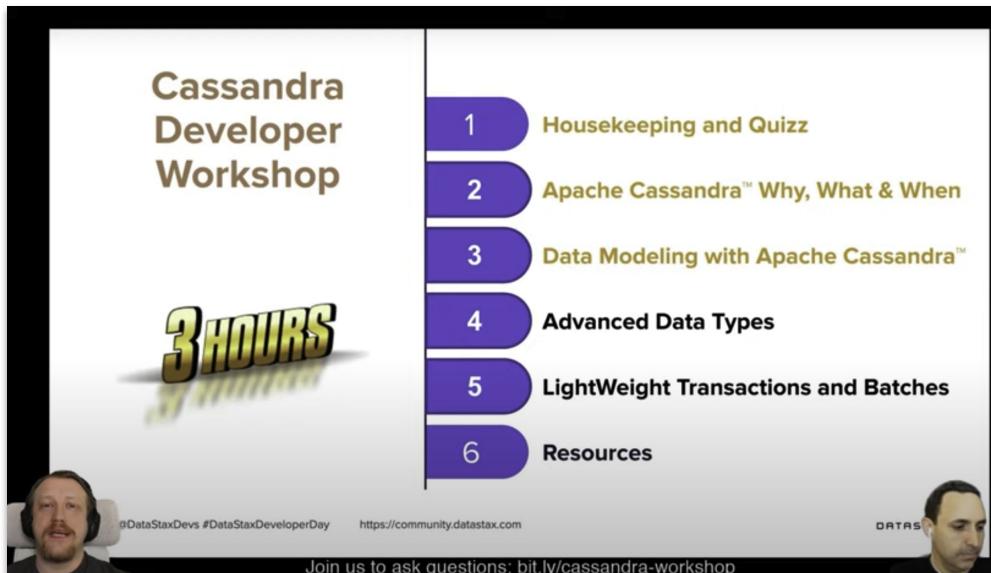
LiveStreams youtube.com/DataStaxDevs



YouTube



Twitch



Links in video description

Questions: bit.ly/cassandra-workshop



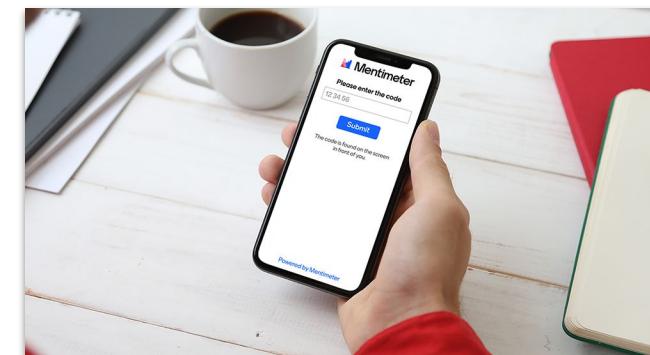
Discord



YouTube



Available on the iPhone
App Store



Quizz: menti.com



GET IT ON
Google play

Mentimeter

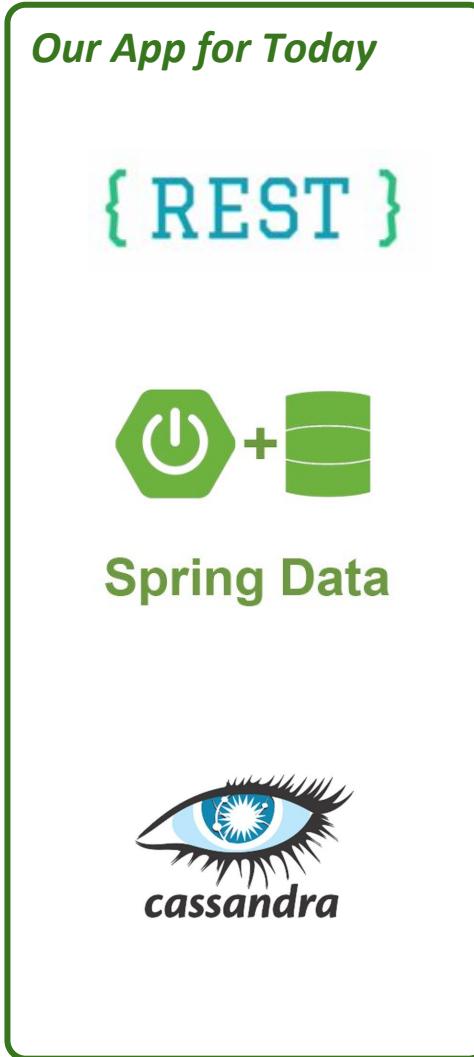
menti.com

67 24 20 4



Available on the iPhone
App Store

GET IT ON
Google play





What we will cover:

TodoApp with **Spring Data Cassandra**



Create Todo Application

- ◆ Run existing Todo Application
- ◆ Create your empty application
- ◆ REST API and mock Repository

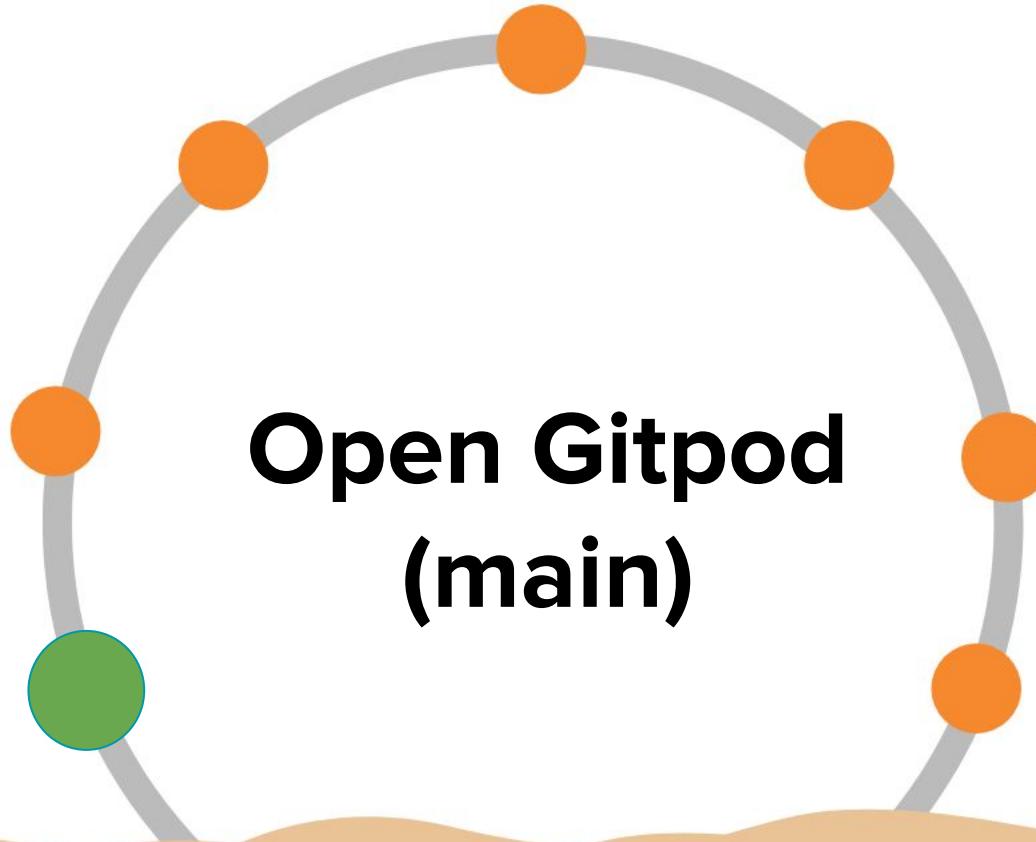


Connect Todo Application

- ◆ Create your Astra Instance
- ◆ Connectivity to Cassandra
- ◆ TodoRepository with Spring Data



<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



**Open Gitpod
(main)**



What we will cover:

TodoApp with **Spring Data Cassandra**



Create Todo Application

- ◆ Run existing Todo Application
- ◆ Create your empty application
- ◆ REST API and mock Repository



Connect Todo Application

- ◆ Create your Astra Instance
- ◆ Connectivity to Cassandra
- ◆ TodoRepository with Spring Data



Helping you **select** an MV* framework

[Download](#)

[View on GitHub](#)

[Blog](#)

Introduction

Developers these days are spoiled with choice when it comes to **selecting** an **MV* framework** for structuring and organizing their JavaScript web apps.

Backbone, Ember, AngularJS... the list of new and stable solutions continues to grow, but just how do you decide on which to use in a sea of so many options?

To help solve this problem, we created **TodoMVC** - a project which offers the same Todo application implemented using MV* concepts in most of the popular JavaScript MV* frameworks of today.

Examples

JavaScript

These are examples w...

Backbone.js®

Dojo®

React®

These are applications

Kotlin + React®

Created and curated by Pete Hodgson.

Featuring HTTP APIs built with:



aiohttp



Akka



API Platform



Axon Framework



Azure Functions



CakePHP



Catalyst



Ceylon



Clojure



CoffeeScript



Compojure



CouchDB



Crystal



C#



Django



.NET



Dropwizard



Elixir



ES6



express



Finatra



Finch



Fintrospect



Flask



Flyway



F#



Dalli



Redis



MongoDB



Redis



MongoDB



Redis



MongoDB



Redis



Redis



Redis



Redis



Redis



Redis



Todo-Backend

a shared example to showcase backend tech stacks

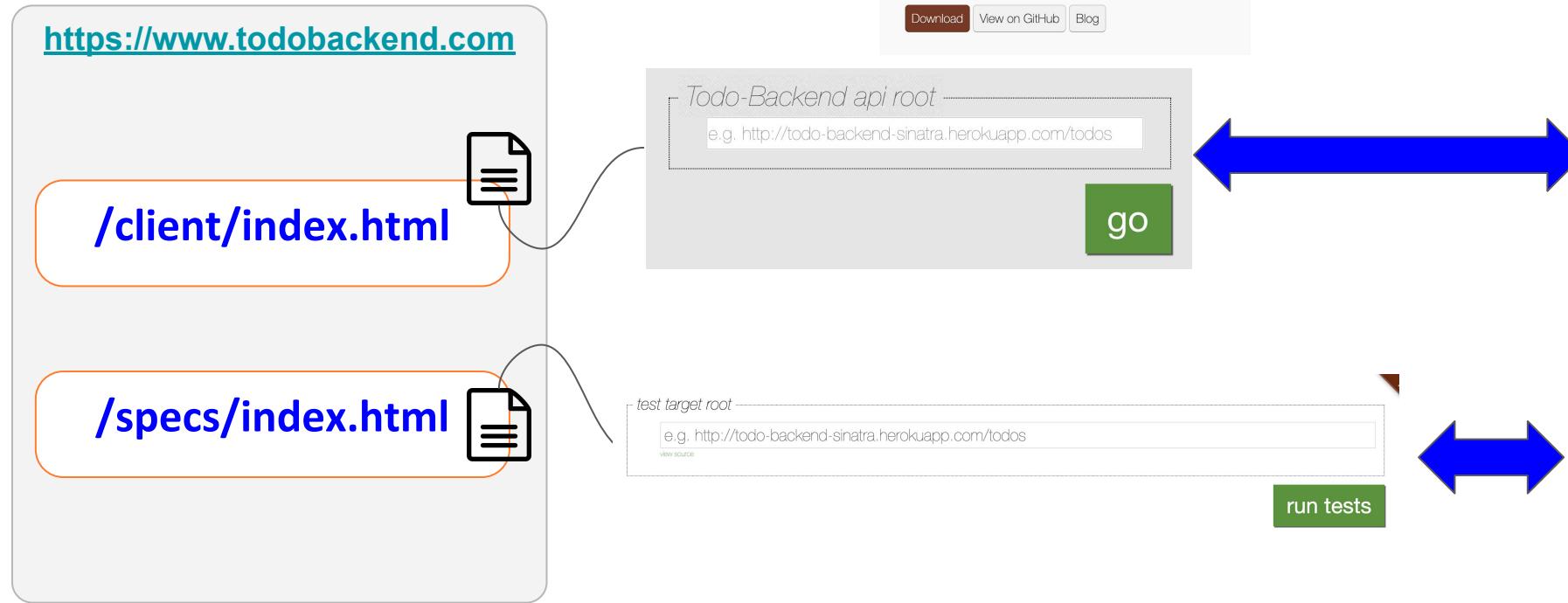
The Todo-Backend project defines a simple web API spec - for managing a todo list. Contributors [implement](#) that spec using various tech stacks. Those implementations are cataloged below. A [spec runner](#) verifies that each contribution implements the exact same API, by running an automated test suite which [defines](#) the API.

The Todo-Backend project was inspired by the TodoMVC project, and some code (specifically the todo client app) was borrowed directly from TodoMVC.



UI Components

<https://www.todobackend.com>



{ REST }



Spring Data



cassandra



<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>





Inversion of control : Work with interfaces

```
public interface GreetingServices {  
    String sayHello();  
}
```

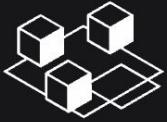
```
@Component  
public class GreetingServicesImpl implements GreetingServices {  
    /** {@inheritDoc} */  
    public String sayHello() {  
        return "hello";  
    }  
}
```

```
@RestController  
public class MyApi {  
  
    private GreetingServices greetingServices;  
  
    // Inversion of control  
    public MyApi(GreetingServices gs) {  
        this.greetingServices = gs;  
    }  
  
    @GetMapping  
    public String hello() {  
        return greetingServices.sayHello();  
    }  
}
```

What Spring can do



spring.io



Microservices

Quickly deliver production-grade features with independently evolvable microservices.



Reactive

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.



Cloud

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.



Web apps

Frameworks for fast, secure, and responsive web applications connected to any data store.



Serverless

The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.



Event Driven

Integrate with your enterprise. React to business events. Act on your streaming data in realtime.

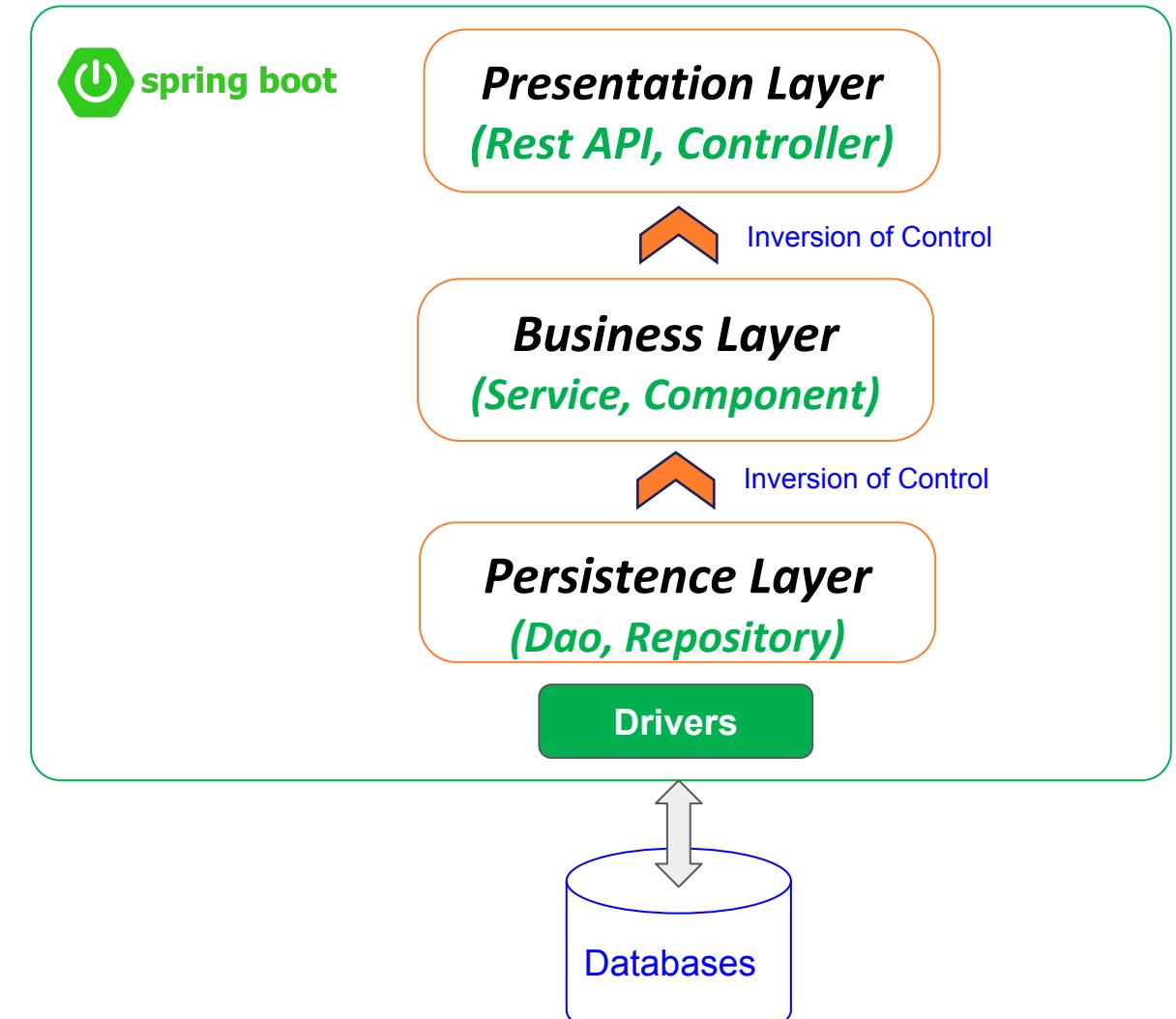


Batch

Automated tasks. Offline processing of data at a time to suit you.

Spring Boot

- Create **stand-alone** Spring applications (no need to deploy WAR files)
- **Simplify** your build configuration
- **Automatically** configure Spring and 3rd party libraries
- Production-ready features such as
 - metrics
 - health checks
 - externalized configuration





Reactive Stack

Spring WebFlux is a non-blocking web framework built from the ground up to take advantage of multi-core, next-generation processors and handle massive numbers of concurrent connections.

Netty, Servlet 3.1+ Containers

Reactive Streams Adapters

Spring Security Reactive

Spring WebFlux

Spring Data Reactive Repositories

Mongo, Cassandra, Redis, Couchbase, R2DBC

Servlet Stack

Spring MVC is built on the Servlet API and uses a synchronous blocking I/O architecture with a one-request-per-thread model.

Servlet Containers

Servlet API

Spring Security

Spring MVC

Spring Data Repositories

JDBC, JPA, NoSQL

CONTROLLERS

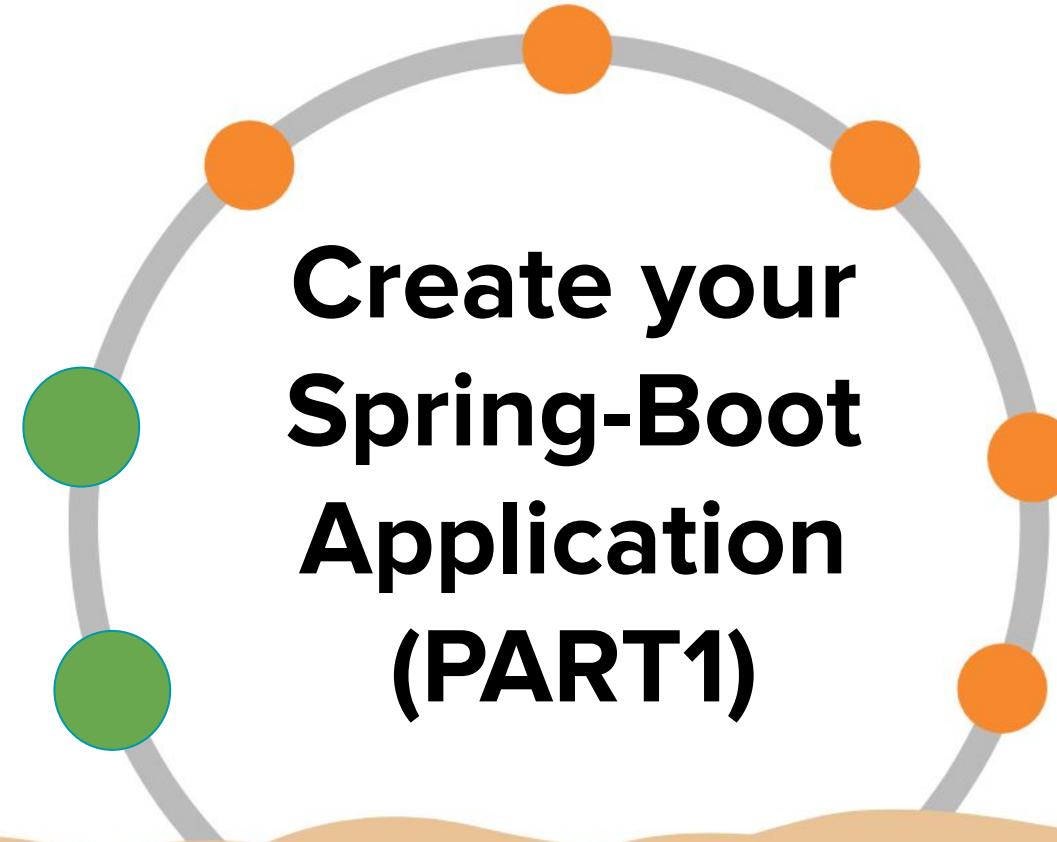
SERVICES

REPOSITORIES

CONNECTIVITY

DATABASE

<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



CONTROLLERS

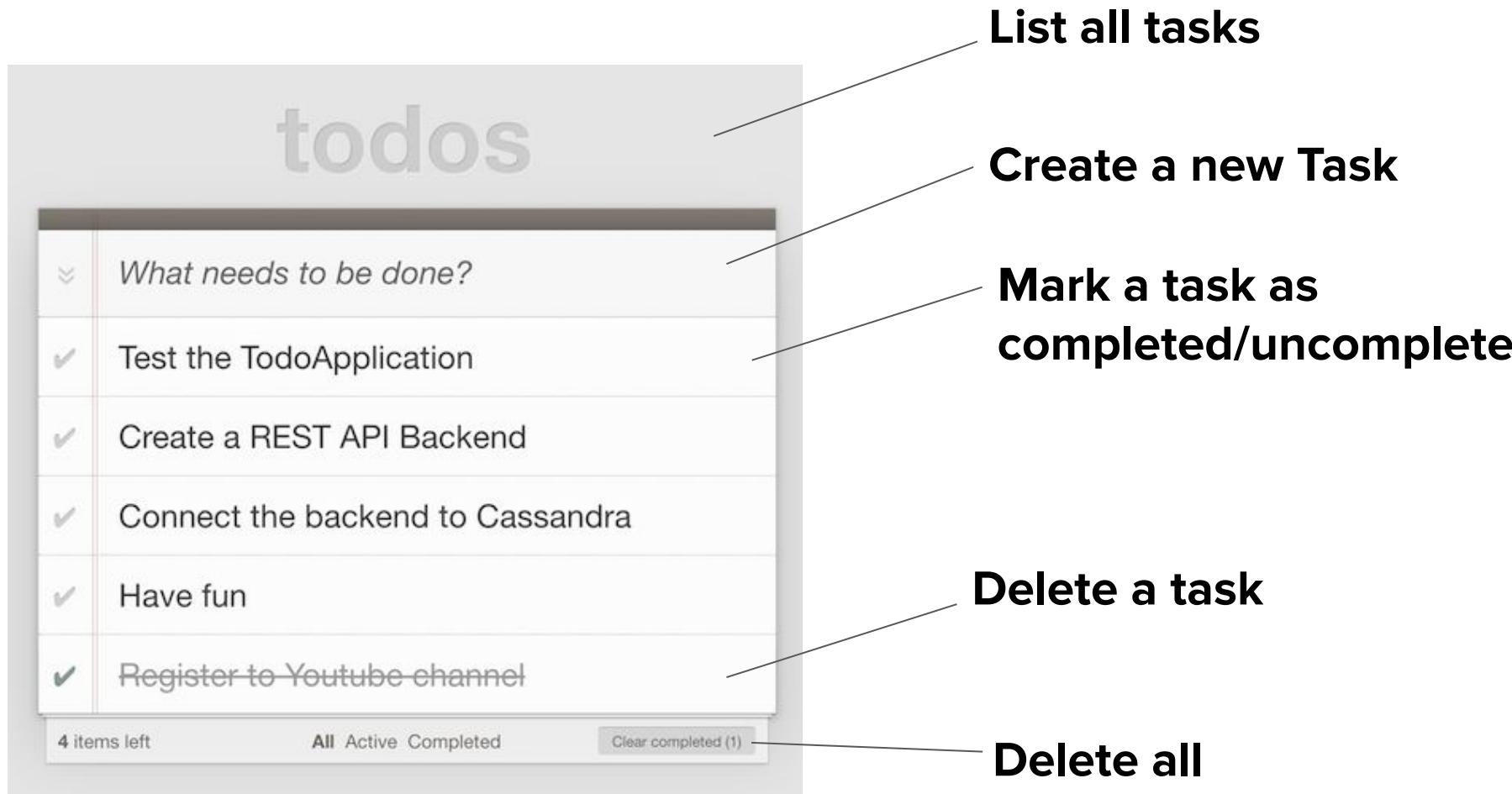
SERVICES

REPOSITORIES

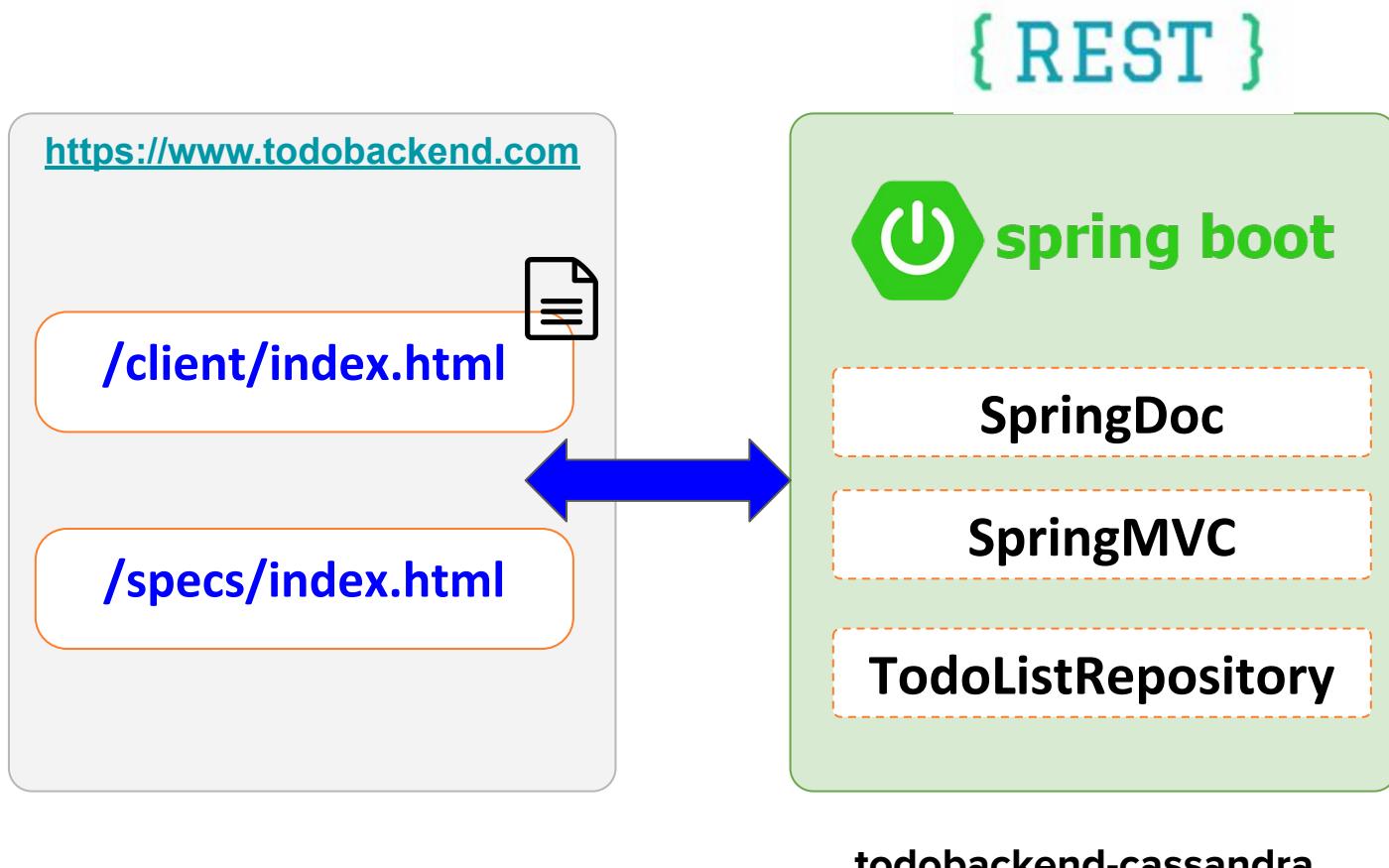
CONNECTIVITY

DATABASE

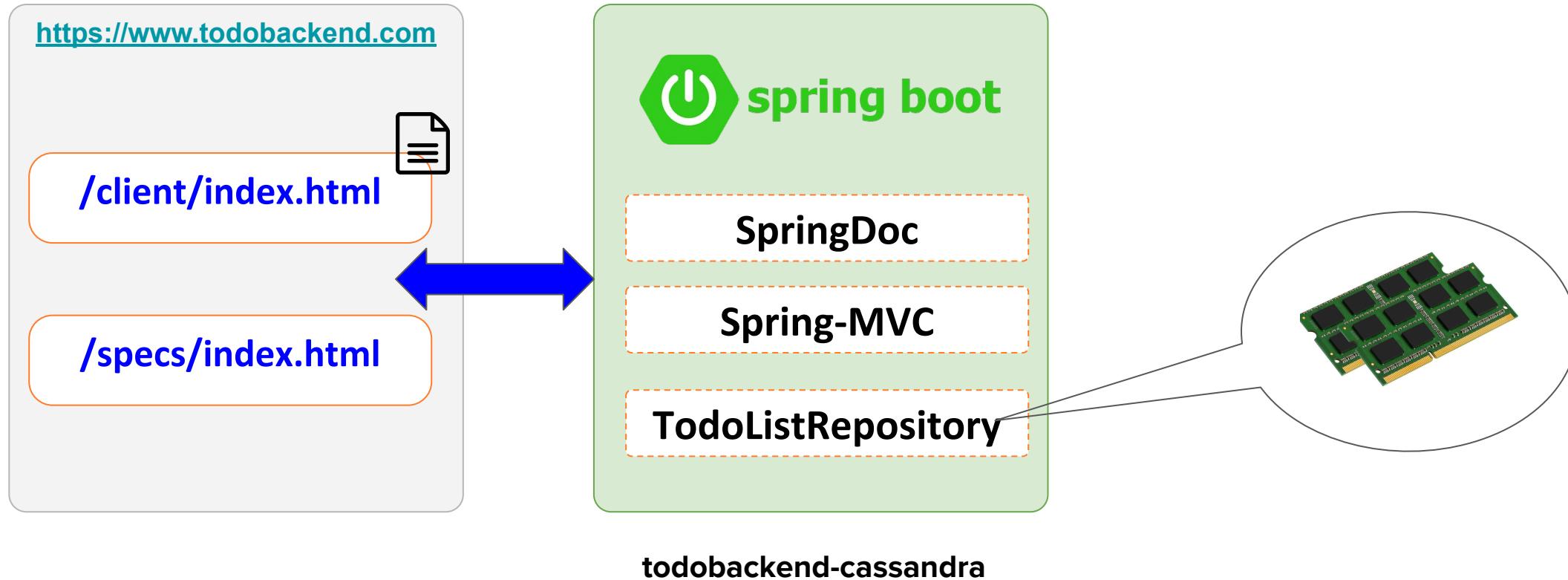
What are the queries? CRUD for ToDos



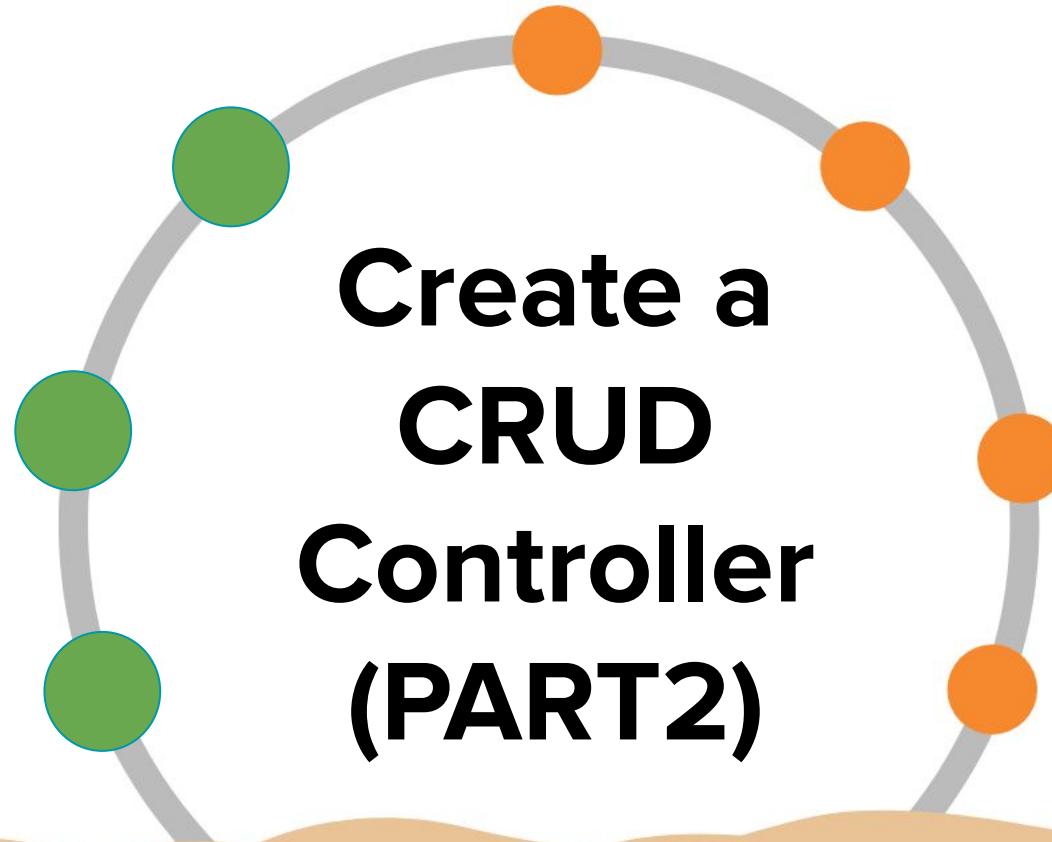
REST API = todobackend-cassandra



REST API = todobackend-cassandra



<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



CONTROLLERS

SERVICES

REPOSITORIES

CONNECTIVITY

DATABASE



What we will cover:

TodoApp with **Spring Data Cassandra**



Create Todo Application

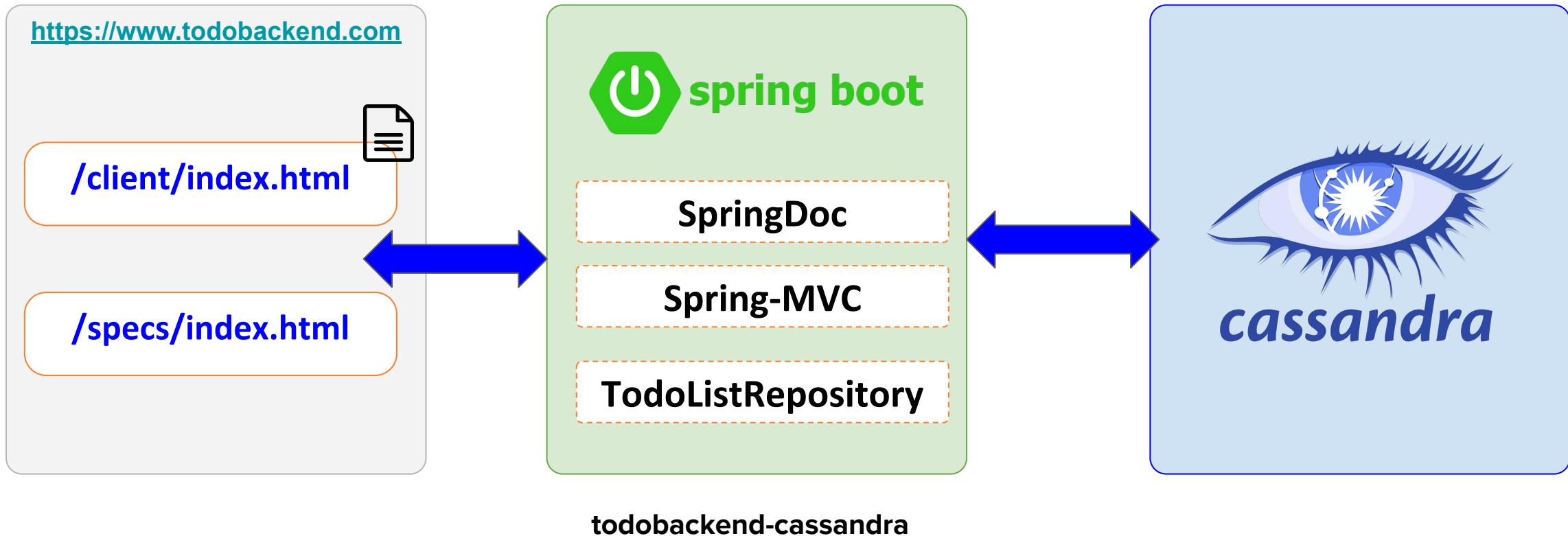
- ◆ Run existing Todo Application
- ◆ Create your empty application
- ◆ REST API and mock Repository



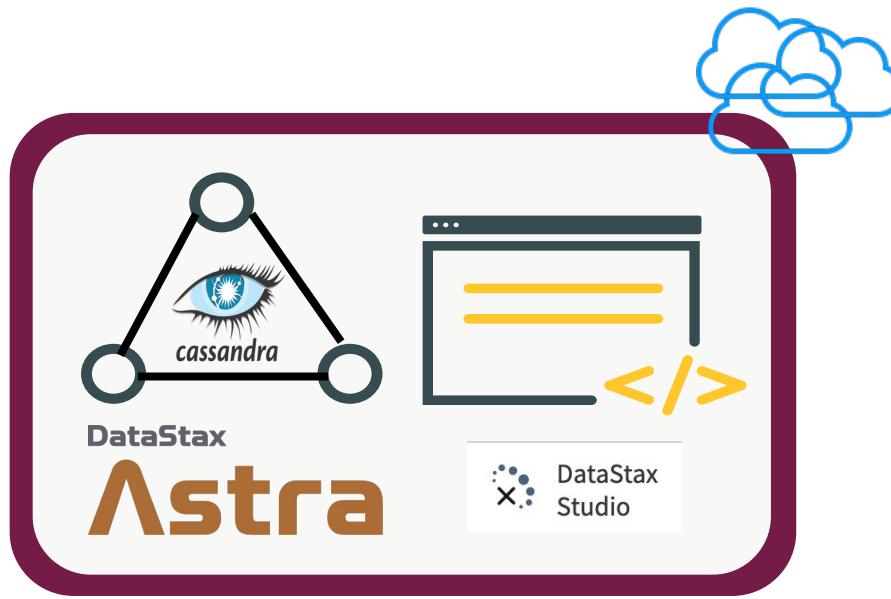
Connect Todo Application

- ◆ Create your Astra Instance
- ◆ Connectivity to Cassandra
- ◆ TodoRepository with Spring Data

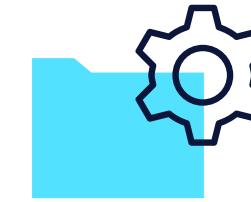
The Todo Application



Introducing ASTRA



Fully managed Cassandra without the ops!



Eliminate Operations

everything from provisioning to backups is fully automated



Secure Your Data

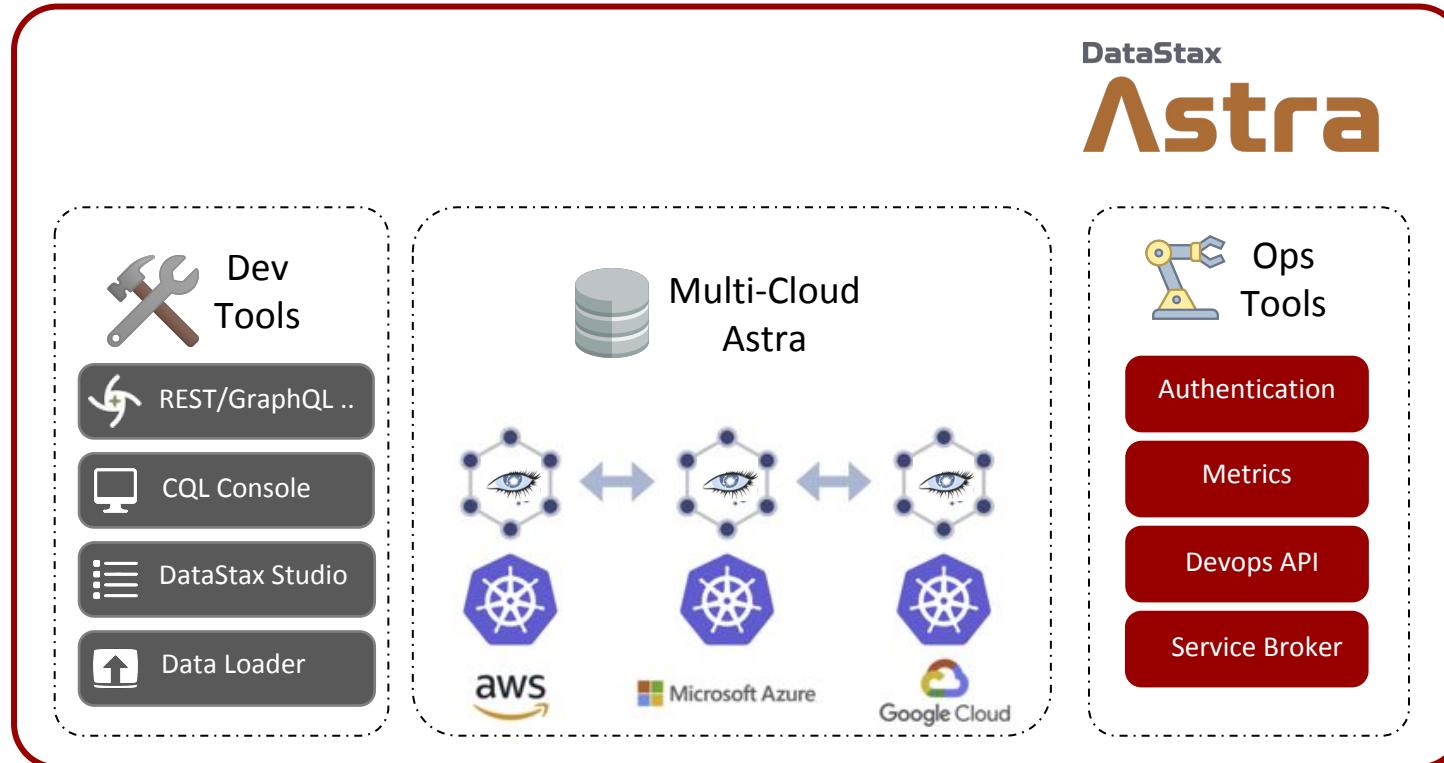
with the most advanced security available for Cassandra



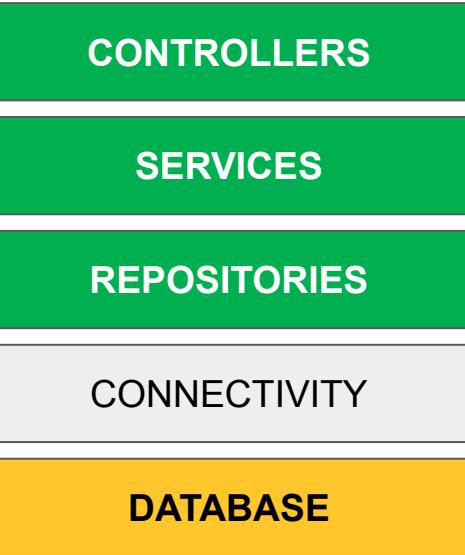
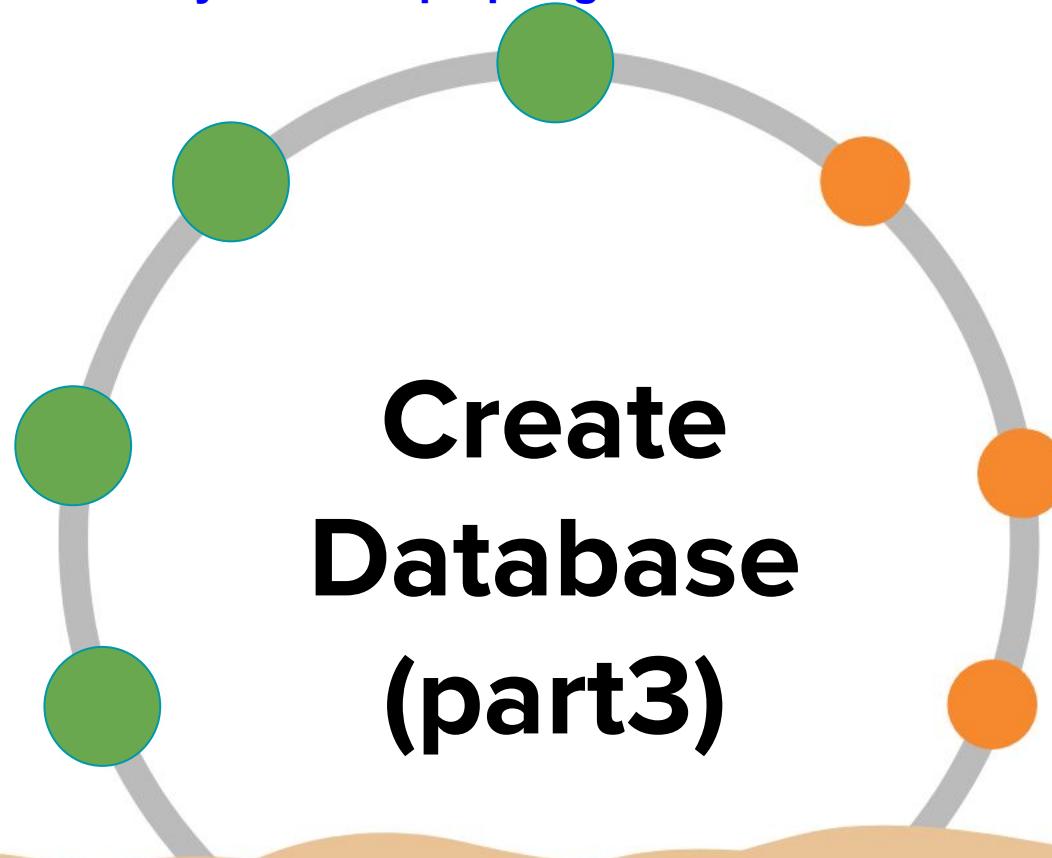
Simplify App Development

with auto-configured developer tools that deploy with a click

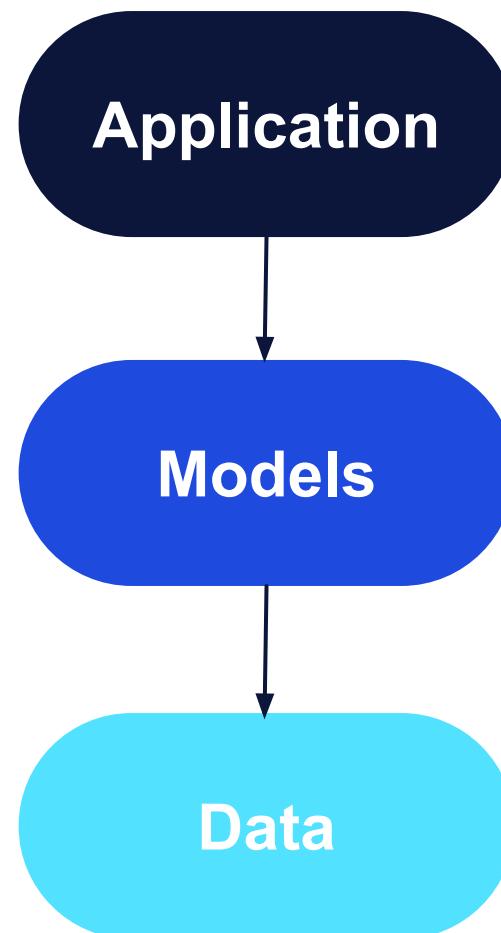
DATASTAX ASTRA



<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



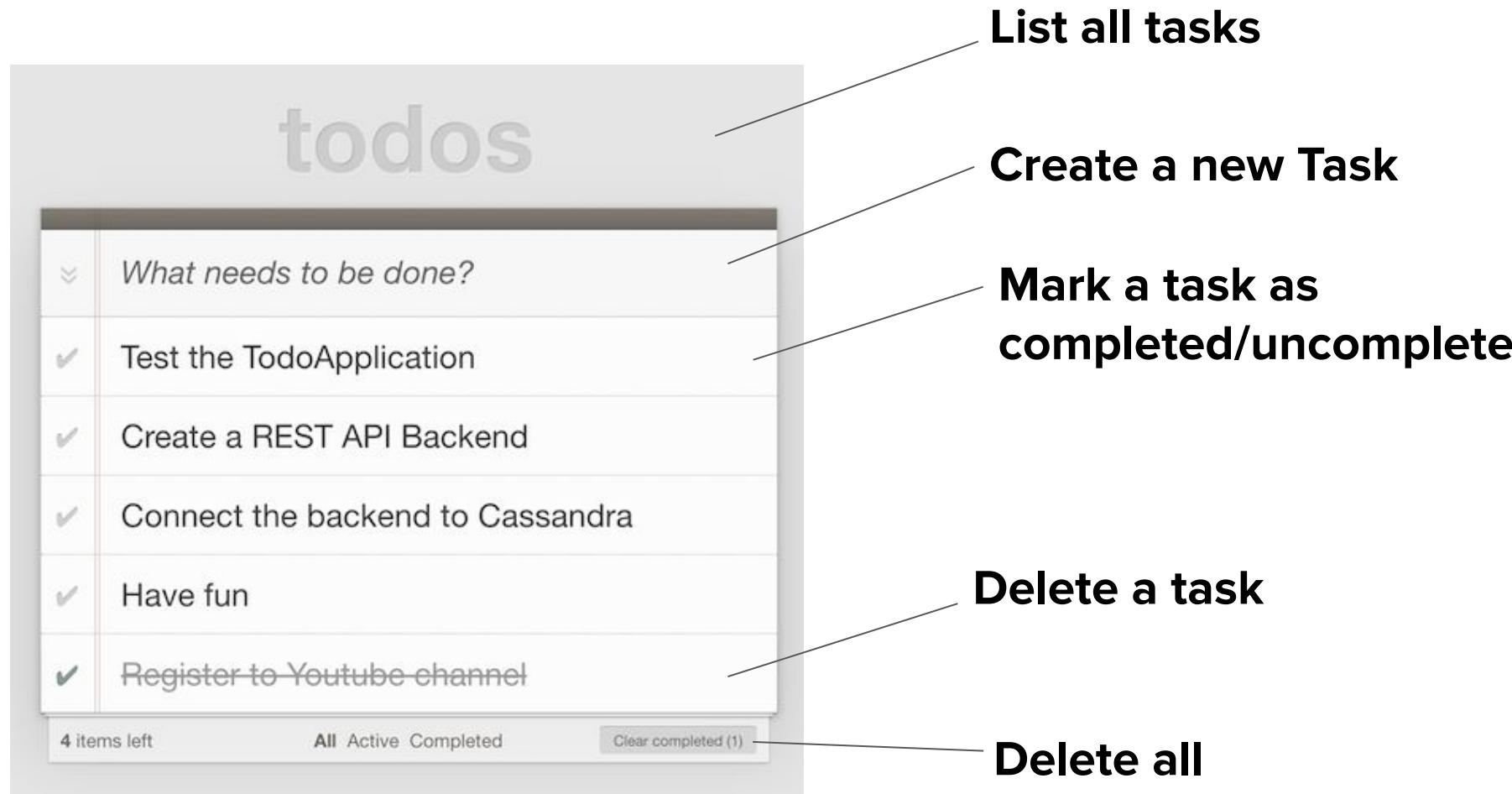
Cassandra Data Model Methodology



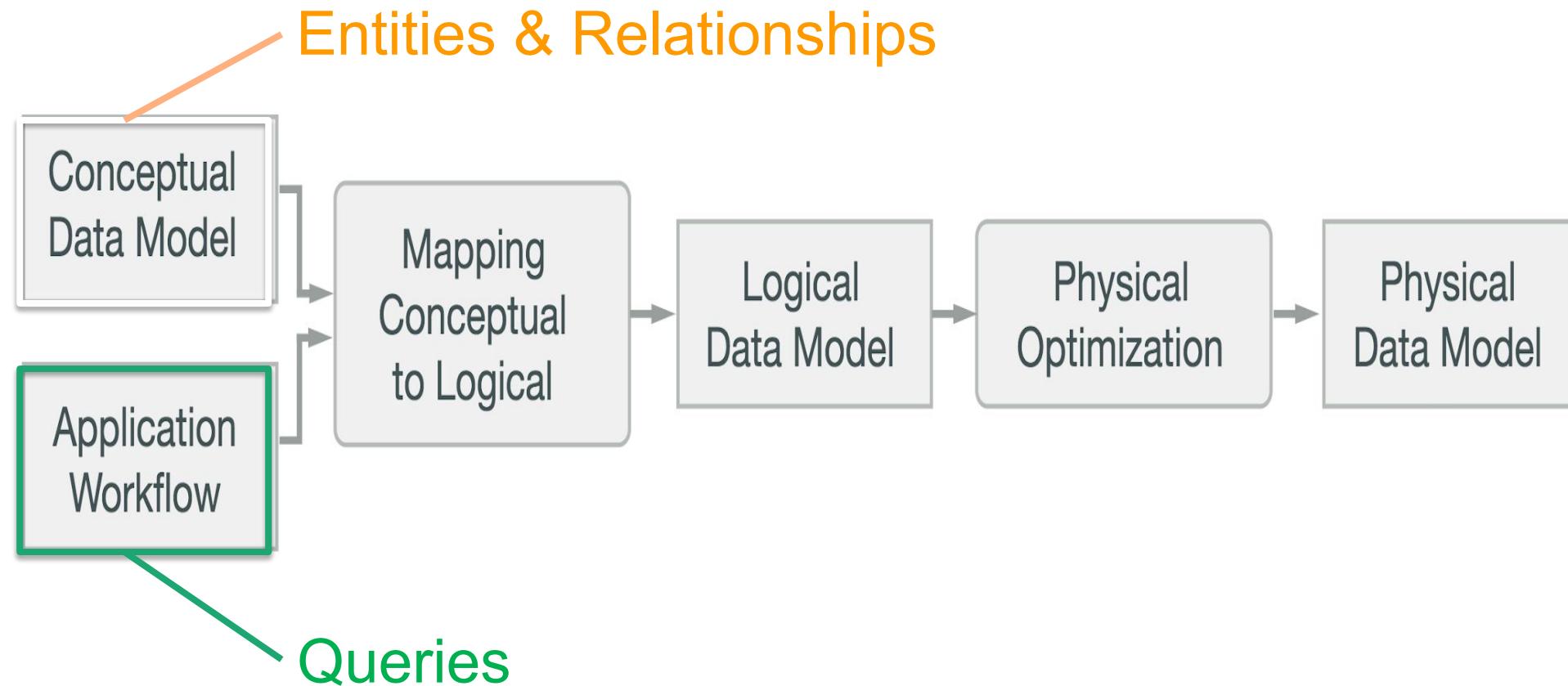
id	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Math



What are the queries? CRUD for ToDos



Designing the Cassandra datamodel



The final task model

todo_tasks		
uid	UUID	K
title	TEXT	
completed	BOOLEAN	
offset	INT	

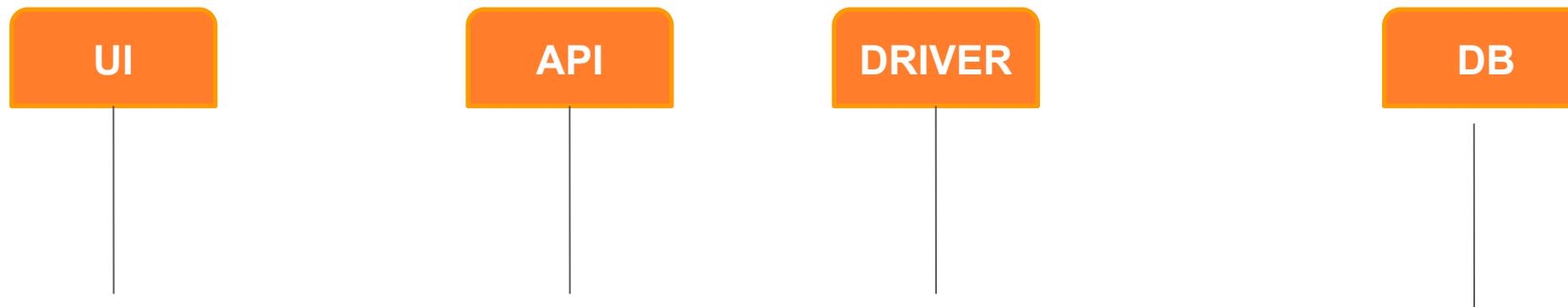
```
CREATE TABLE todoapp.todo_tasks (
    uid          uuid,
    title        text,
    completed    boolean,
    offset       int,
    PRIMARY KEY ((uid))
);
```

Logical Architecture

A screenshot of a web application titled 'todos'. It displays a list of tasks with checkboxes and labels: 'What needs to be done?', 'Explain the use case', 'Create the Data model', 'Define the queries to perform', 'Create the DDL', 'Connect to Cassandra', 'Create the CRUD repository', and 'Run the API'. At the bottom, there are buttons for 'All', 'Active', and 'Completed' tasks, and a note '7 items left'.

A screenshot of the DevWorkshop :: TodoBackend Rest API interface. It shows a Swagger UI for a Spring WebMVC application. The 'Todos' endpoint is highlighted, showing various HTTP methods and their corresponding URLs and descriptions. The URL is `/api/v1/todos/{taskId}`.

A screenshot of the DataStax Astra dashboard. It shows a database named 'bottlestox_db' with details like 'Status: Active', 'Date Created: Dec 7, 2020', 'Compute Size: Free', 'Replication Factor: 1', 'Current Capacity: 5 GB', and 'Current Usage: 1 GB'. The 'Regions' section shows 'Provider: GCP' and 'Region: us-east1'. A red arrow points to the 'Cluster ID' field, which contains the value '7898f114-e5de-4e45-b271-999de...'. Other sections include 'Summary', 'Health', 'Connect', 'CQL Console', 'Studio', and 'Settings'.



Datastax Drivers

One of set drivers to connect them all - January 2020



Connectivity

- Token & Datacenter Aware
- Load Balancing Policies
- Retry Policies
- Reconnection Policies
- Connection Pooling
- Health Checks
- Authentication | Authorization
- SSL

Query

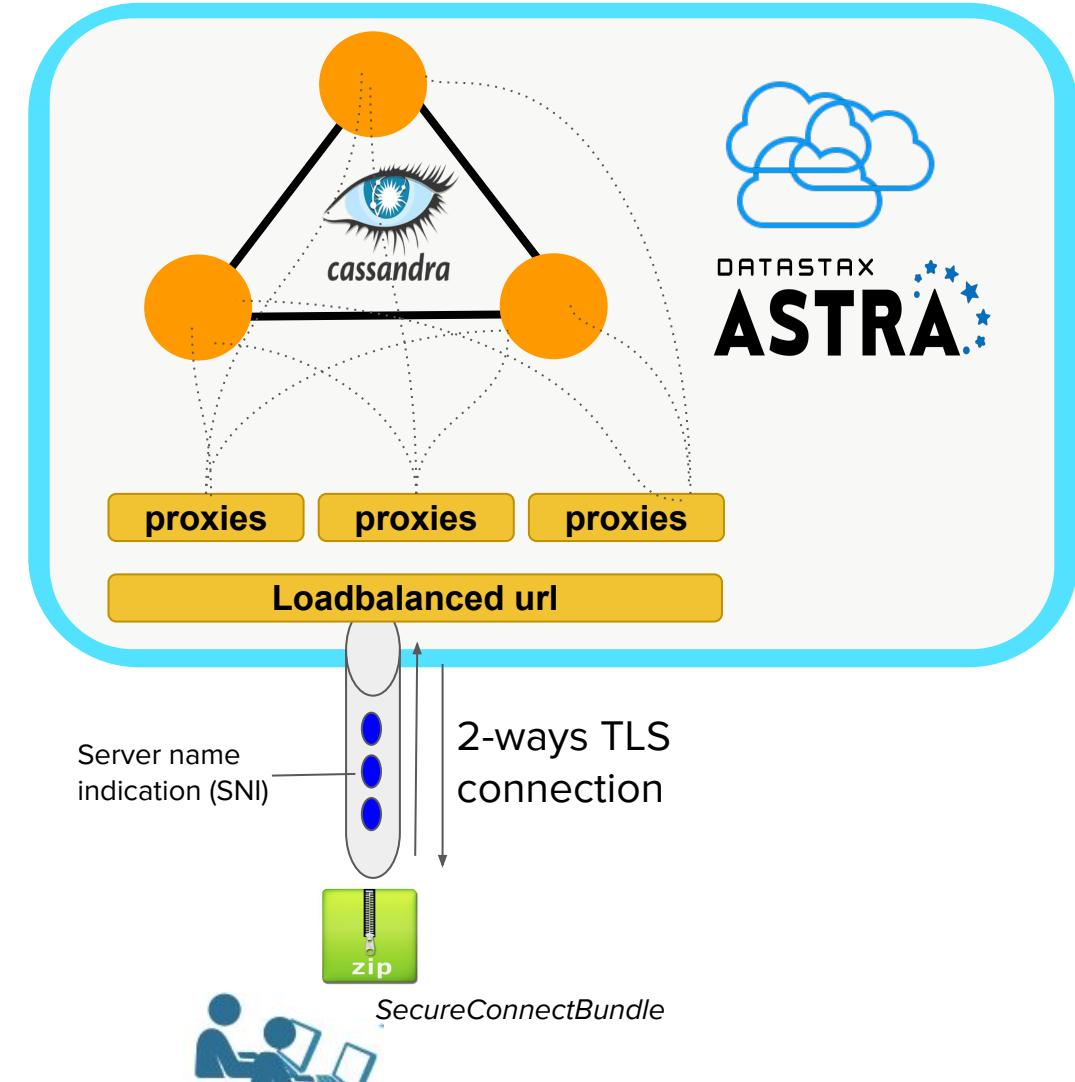
- CQL Support
- Schema Management
- Sync/Async/Reactive API
- Query Builder
- Compression
- Paging

Parsing Results

- Lazy Load
- Object Mapper
- Spring Support
- Paging

Contact points with Astra

- Download and use the secure connect bundle
- Pass user name, password, keyspace name and secure connect bundle to build the cql session



Working with the driver: Open a session

```
// Explicit Settings

CqlSession cqlSession = CqlSession.builder()

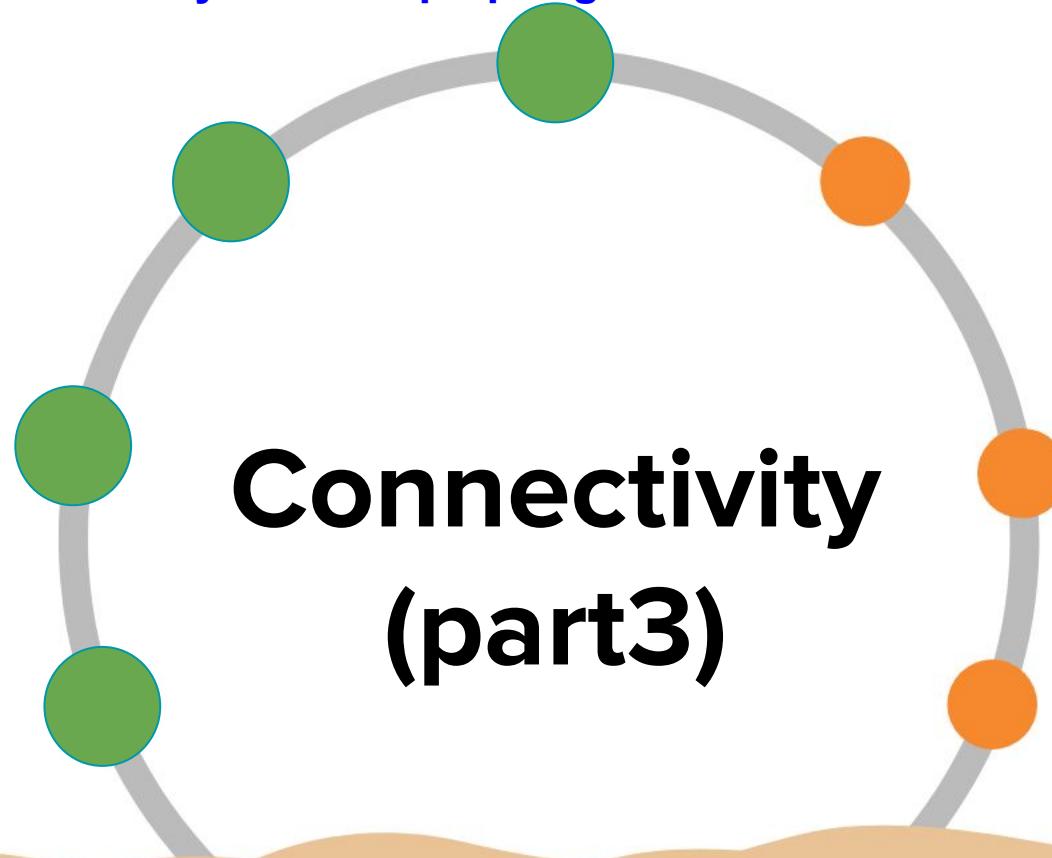
.withCloudSecureConnectBundle(Paths.get("/tmp/secure-connect-bundle.zip"))

.withKeyspace("todoapp")

.withAuthCredentials("KVUser", "KVPassword")

.build();
```

<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



CONTROLLERS

SERVICES

REPOSITORIES

CONNECTIVITY

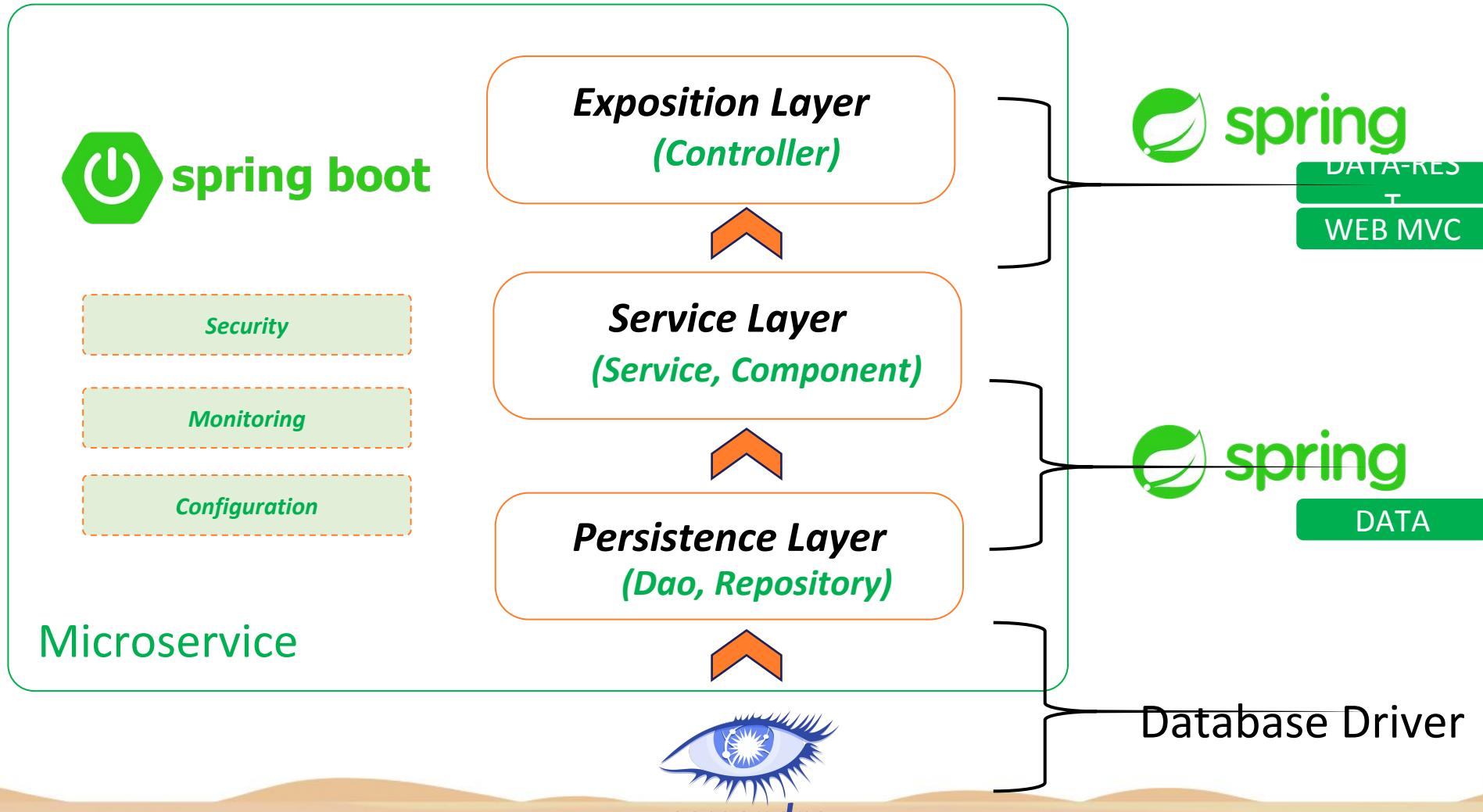
DATABASE

<https://docs.spring.io/spring-data/cassandra/docs/3.1.2/reference/html/#>



Spring Data

Spring Data



AbstractCassandraConfiguration (without Spring Boot) vs Spring-Boot convention vs Your own definition of CQLSESSION

<https://docs.spring.io/spring-data/cassandra/docs/3.0.4.RELEASE/reference/html/#cassandra.cassandra-java-config>

spring-boot-starter-data-cassandra

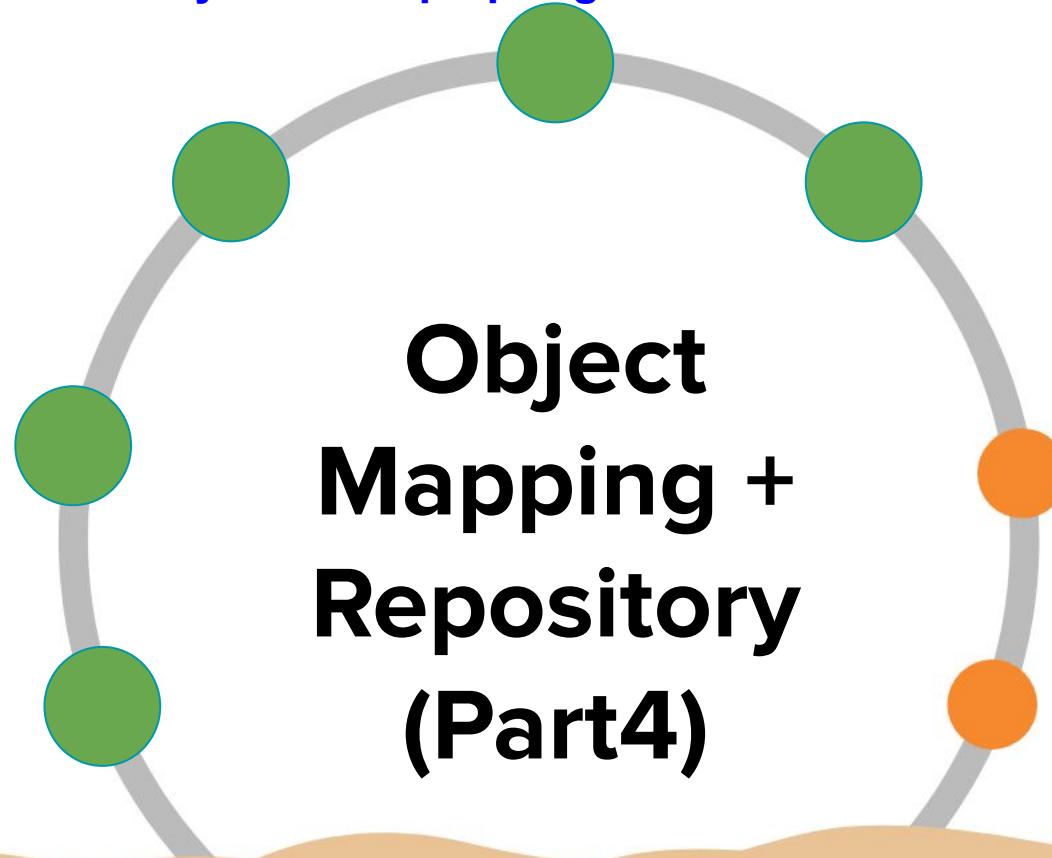
```
spring.data.cassandra.schema-action=CREATE_IF_NOT_EXISTS  
spring.data.cassandra.request.timeout=10s  
spring.data.cassandra.connection.connect-timeout=10s  
spring.data.cassandra.connection.init-query-timeout=10s  
spring.data.cassandra.keyspace-name=todokeyspace  
spring.data.cassandra.username=todouser  
spring.data.cassandra.password=todo_password1  
  
datastax.astra.secure-connect-bundle=/tmp/creds.zip
```

Entity and Repository

```
@Entity  
public class Task {  
  
    @Id  
    @PrimaryKeyColumn(  
        name = "uid", ordinal = 0,  
        type = PrimaryKeyType.PARTITIONED)  
    private UUID uid;  
  
    private String title;  
  
    private boolean complete;  
  
    private int offset;  
  
    private Task() {}  
  
    //...}
```

```
public interface TaskRepository extends  
CrudRepository<Task, UUID> {  
  
    @Query("SELECT * FROM todos_tasks WHERE uid=?0")  
    Optional<TaskSpringData> findByTaskById0(UUID  
    taskid);  
  
}
```

<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



CONTROLLERS

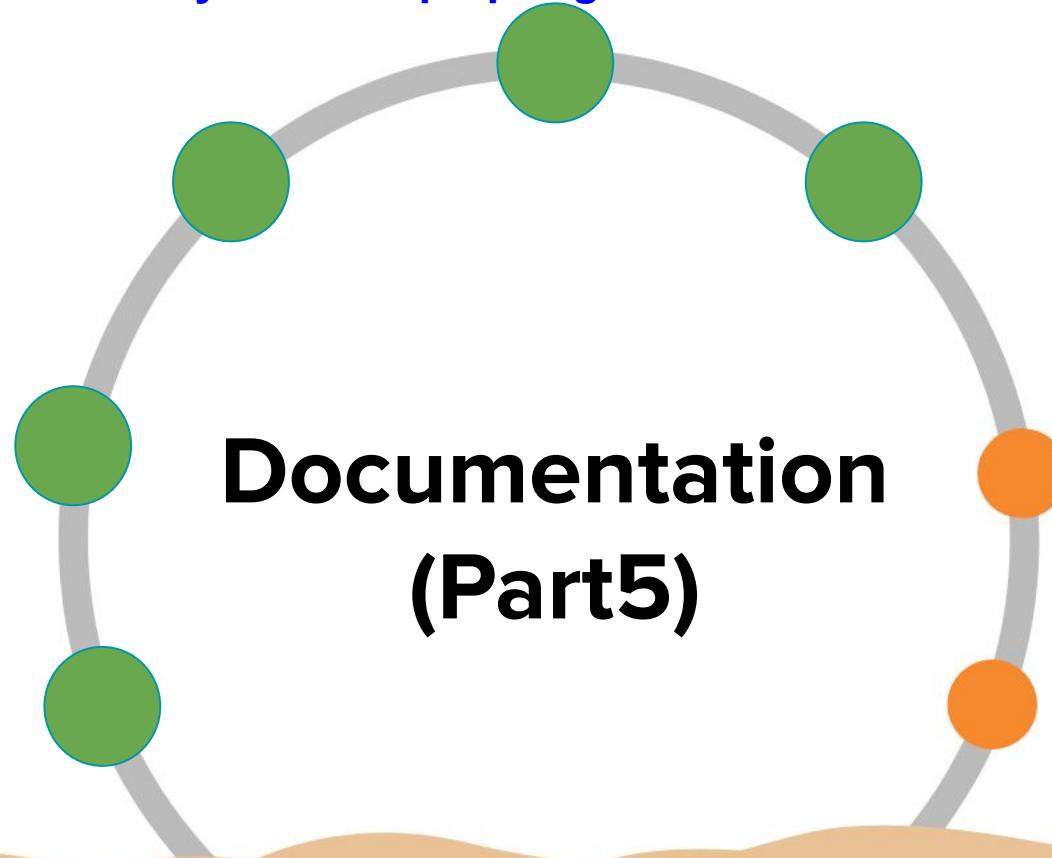
SERVICES

REPOSITORIES

CONNECTIVITY

DATABASE

<https://github.com/DataStax-Academy/workshop-spring-data-cassandra>



CONTROLLERS

SERVICES

REPOSITORIES

CONNECTIVITY

DATABASE

menti.com

67 24 20 4



#CassandraWorkshopSeries



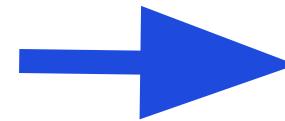
Available on the iPhone
App Store

GET IT ON
Google play

Menti Top 3!

- Top 3 people in Menti, please email Jack to claim your prize!

jack.fryer@datastax.com



**Note: Please allow up to 2 weeks for your
prize to arrive. Shipping times vary.**



MORE LEARNING!!!!

Developer site: datastax.com/dev

- Developer Stories
- New hands-on learning scenarios with Katacoda

- Try it Out
 - Cassandra Fundamentals
 - New Data Modeling course in progress,
sneak preview at
- <https://katacoda.com/datastax/courses/cassandra-data-modeling>

Classic courses available at [DataStax Academy](#)



Katacoda

Developer Resources

LEARN

New hands-on learning at www.datastax.com/dev
Classic courses available at DataStax Academy

ASK/SHARE

Join community.datastax.com
Ask/answer community user questions - share your expertise

CONNECT

Follow us
We are on Youtube - Twitter - Twitch!

MATERIALS

Slides and practice questions for this course are available at
<https://github.com/DataStax-Academy/workshop-spring-data>



Thank You

