

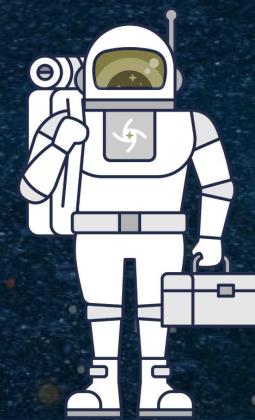


An **Api Layer** for Databases,
Exposing **Rest**, **GraphQL** and
gRPC API for your data
with no code.

DataStax
Developers



STARGATE



Cédrick Lunven

Director Developer Relations



SDK



- Trainer
- Public Speaker
- Developers Support
- Developer Applications
- Developer Tooling



- Creator of ff4j (ff4j.org)
- Maintainer for 8 years+



- Happy developer for 14 years
- Spring Petclinic Reactive & Starters
- Implementing APIs for 8 years

DataStax Developers Advocates Team



Cedrick
Lunven

Aleksandr
Volochnev

Jack
Fryer

Kirsten
Hunter

Stefano
Lottini

David
Gilardi

Ryan
Welford

Rags
Srinivas

Sonia
Siganporia

R

S

Agenda

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



Tooling
SDK, K8ssandra

05



What's NEXT ?
gRPC, CDC, sql...

Agenda

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



Tooling
SDK, K8ssandra

05



What's NEXT ?
gRPC, CDC, sql...

Start with Why – *Simon Sinek*

- **Developers**

- Do you like learning query languages (CQL, N1QL, GQL, cypher, gremlin....)
 - No. Save my JSON, give it to me back when I need it
 - ORM / Spring Data are so popular nowadays
- Do you care about how your data is stored ?
 - Physical data model part of the interface...yikes
 - Create structures based on queries
- Do you like installing and running Databases locally
 - especially distributed databases
 - especially with datasets and integration tests

Start with Why

- Operators and Databases Administrators...
 - Do you allow developers to execute direct queries against your DB ?
 - Do you like opening port ranges like 0-65536 to allow communications with applications, especially in the cloud.
 - Do you like creating dedicated projects and hiring people just to create APIs to expose an existing treatment (digital transformation FTW)

And when those 2 meet each other

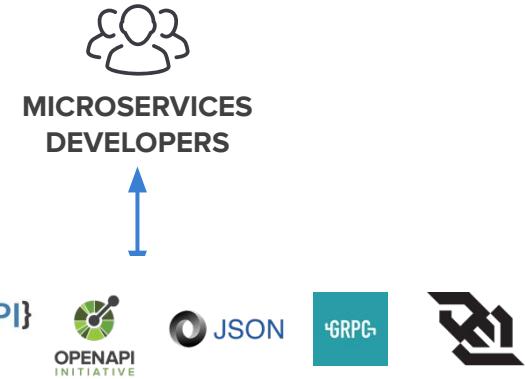


Data Gateway



Developers want the option to use modern APIs and development gateways.

Cassandra is a database designed for the new standard and the associated APIs that facilitate the first choice of developers.



Stargate Overview

An open source API framework for data

Stargate makes it easy to use a database for any application workload by adding plugin support for new APIs, data types, and access methods



MICROSERVICES
DEVELOPERS

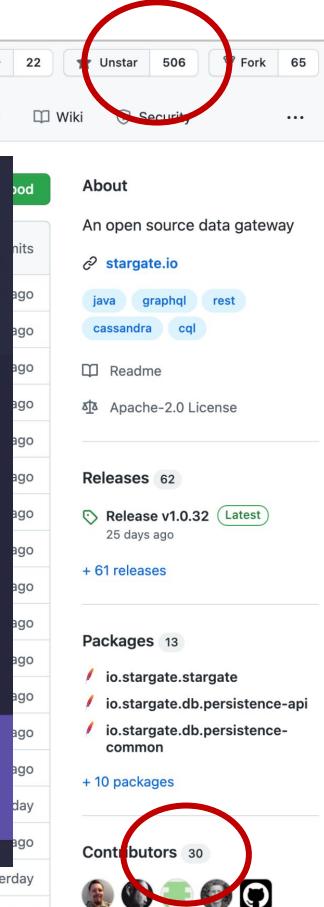
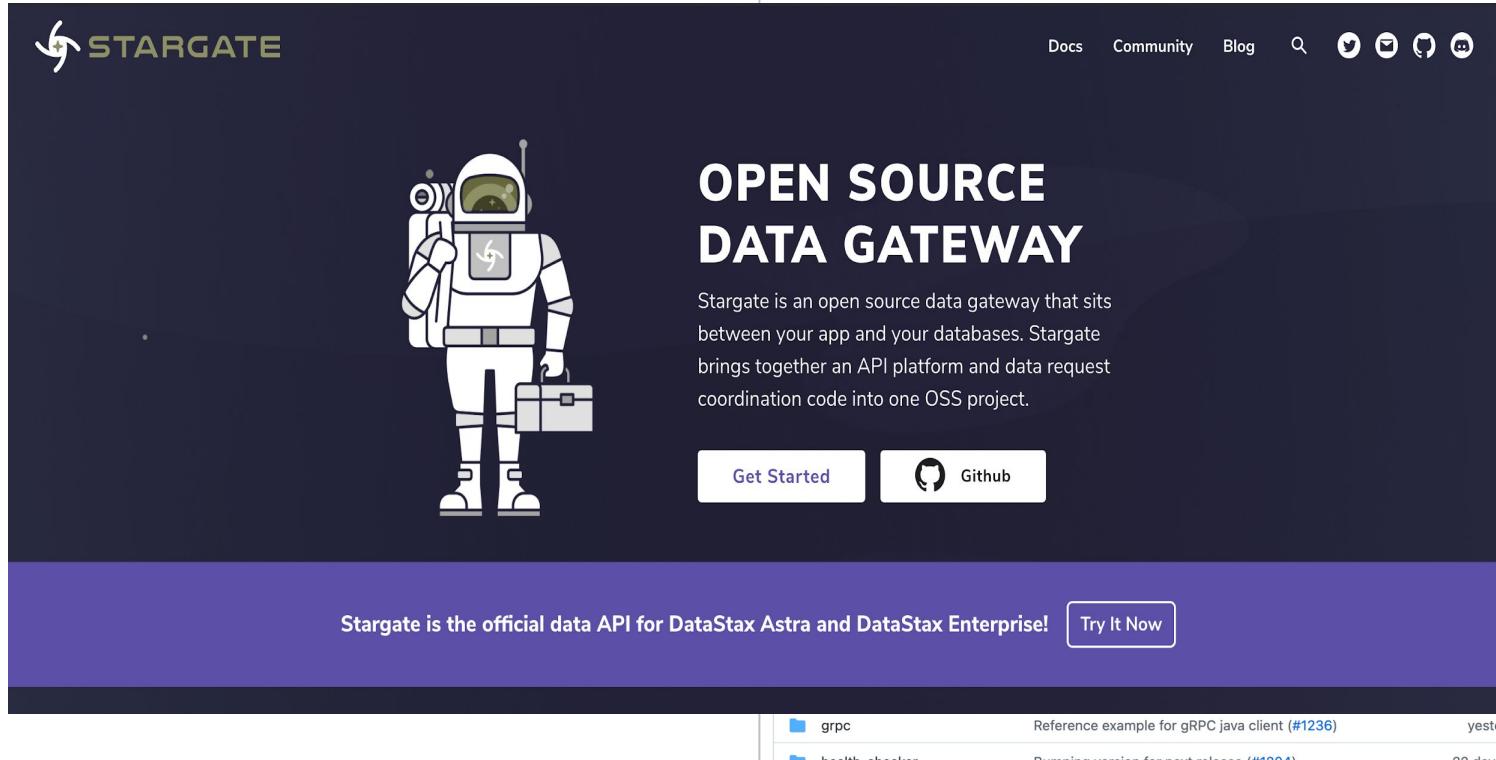


STARGATE



cassandra

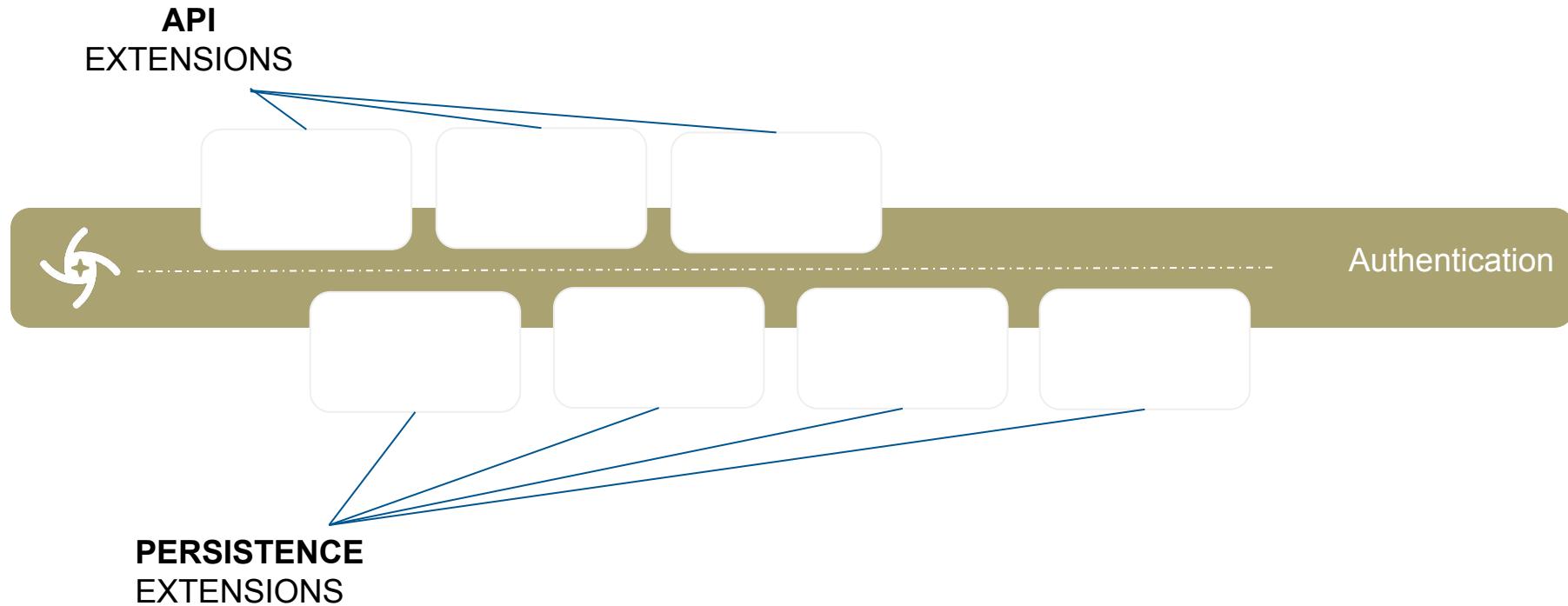
Stargate.io



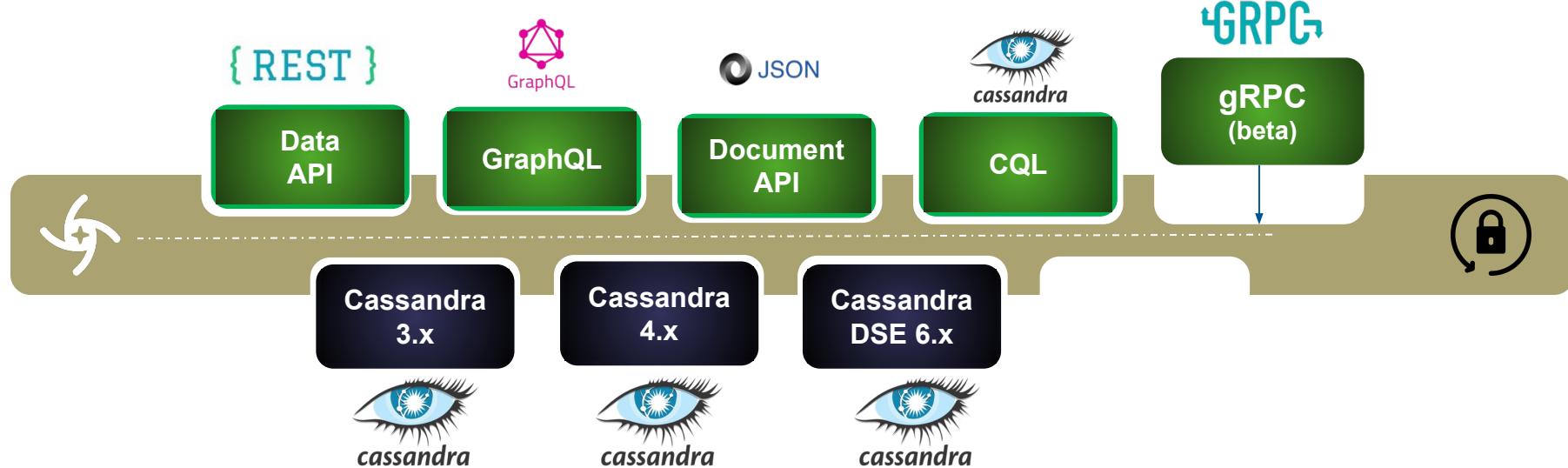
apidays

DataStax Developers

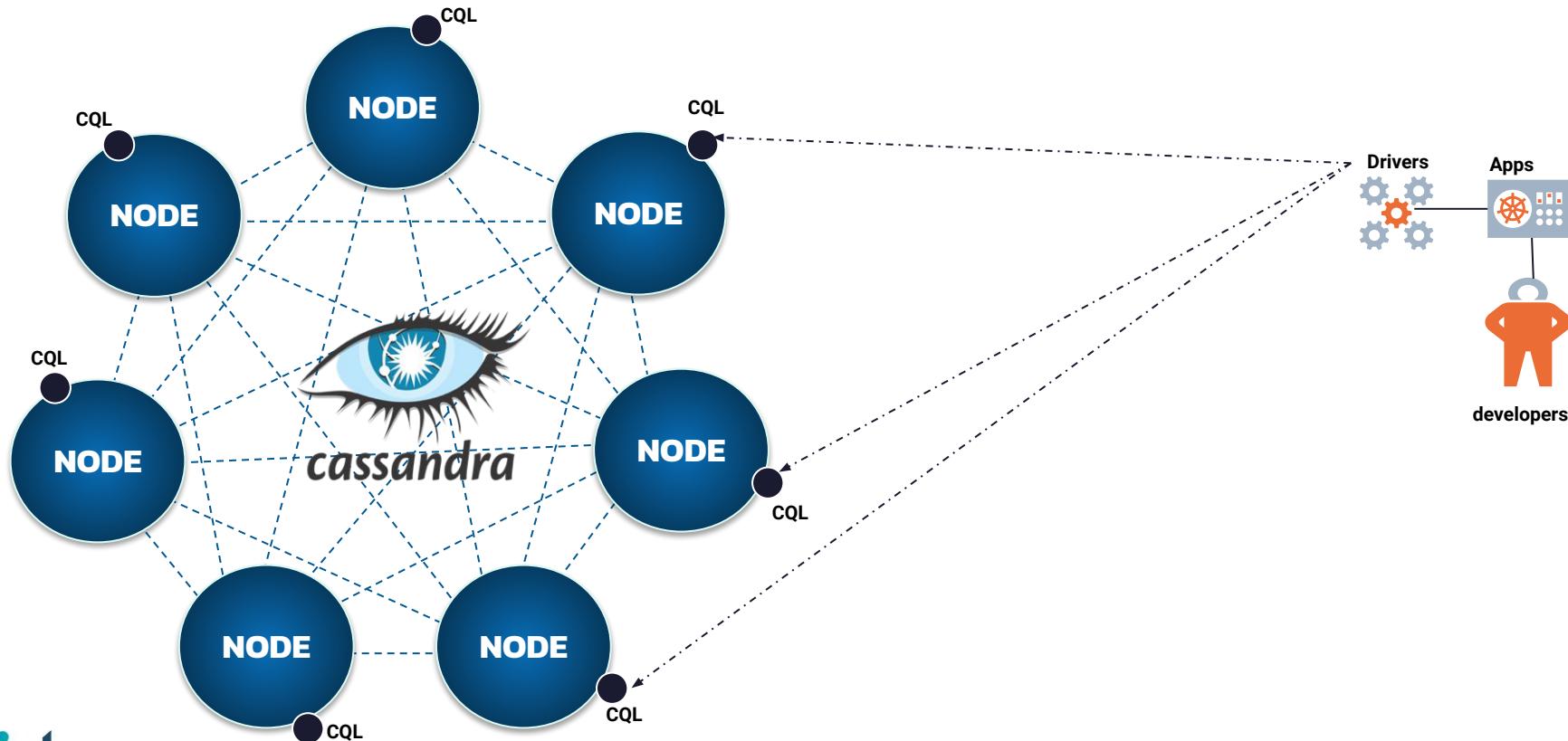
API and Persistence Extensions



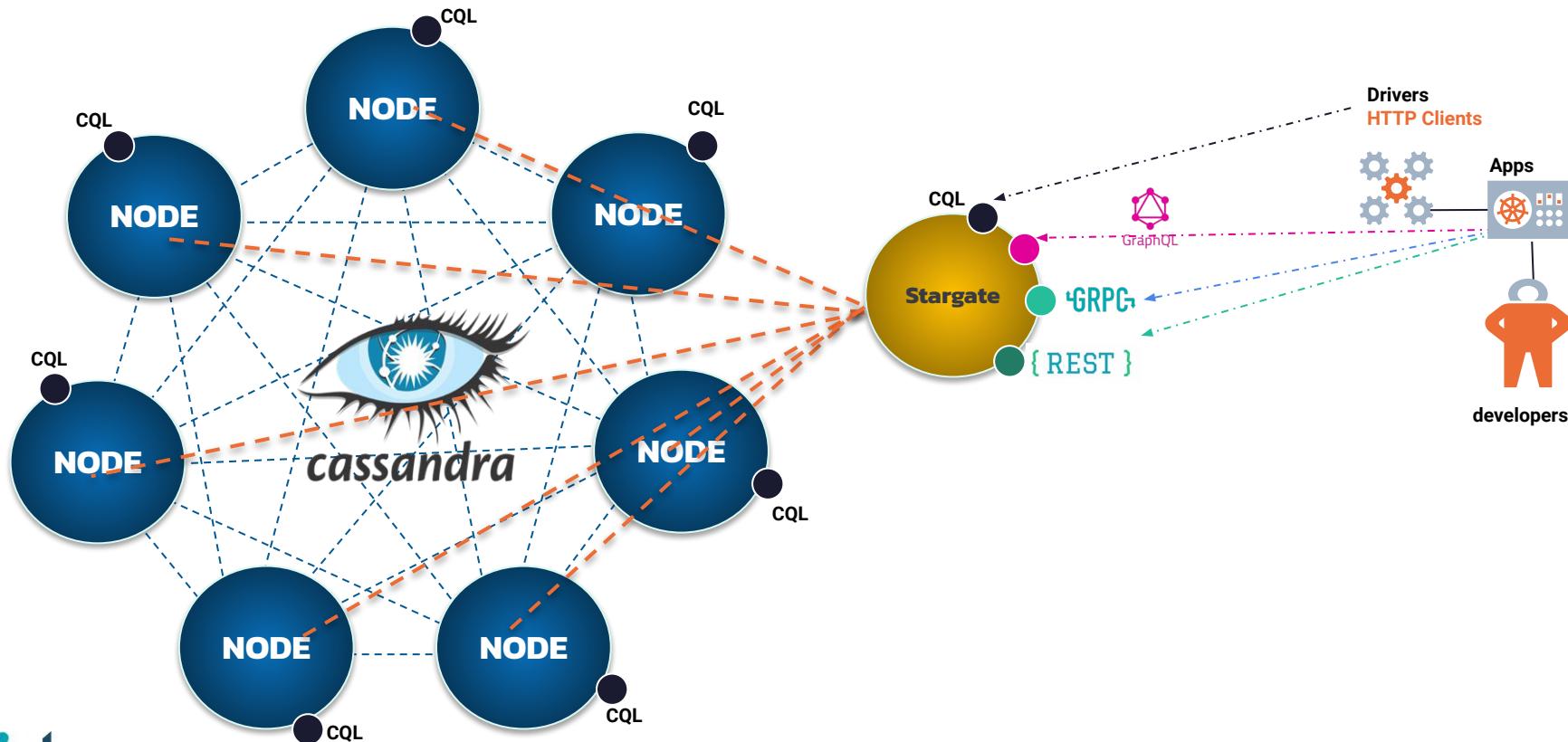
API Extensions and Persistence Extensions



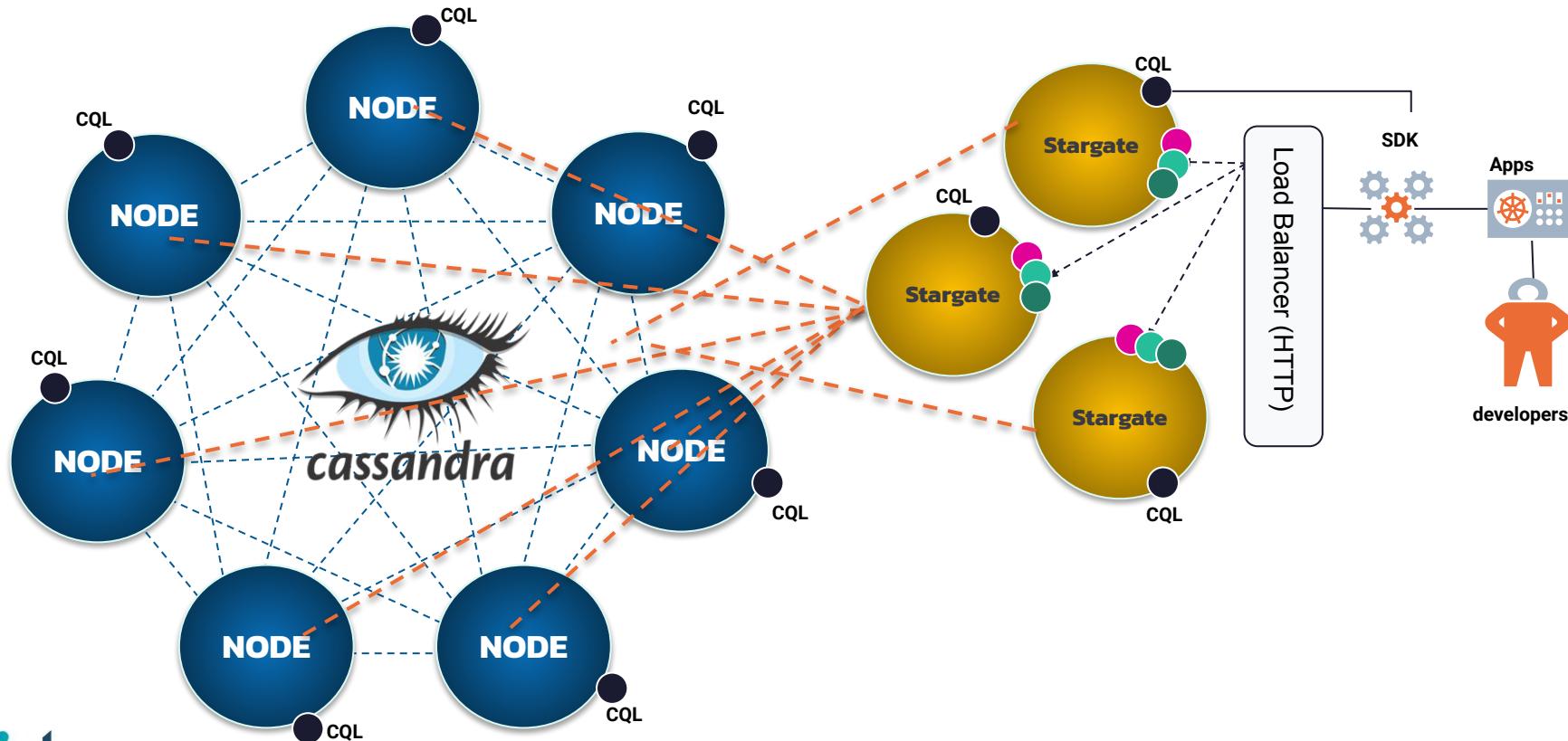
Connecting to your cluster (Before)

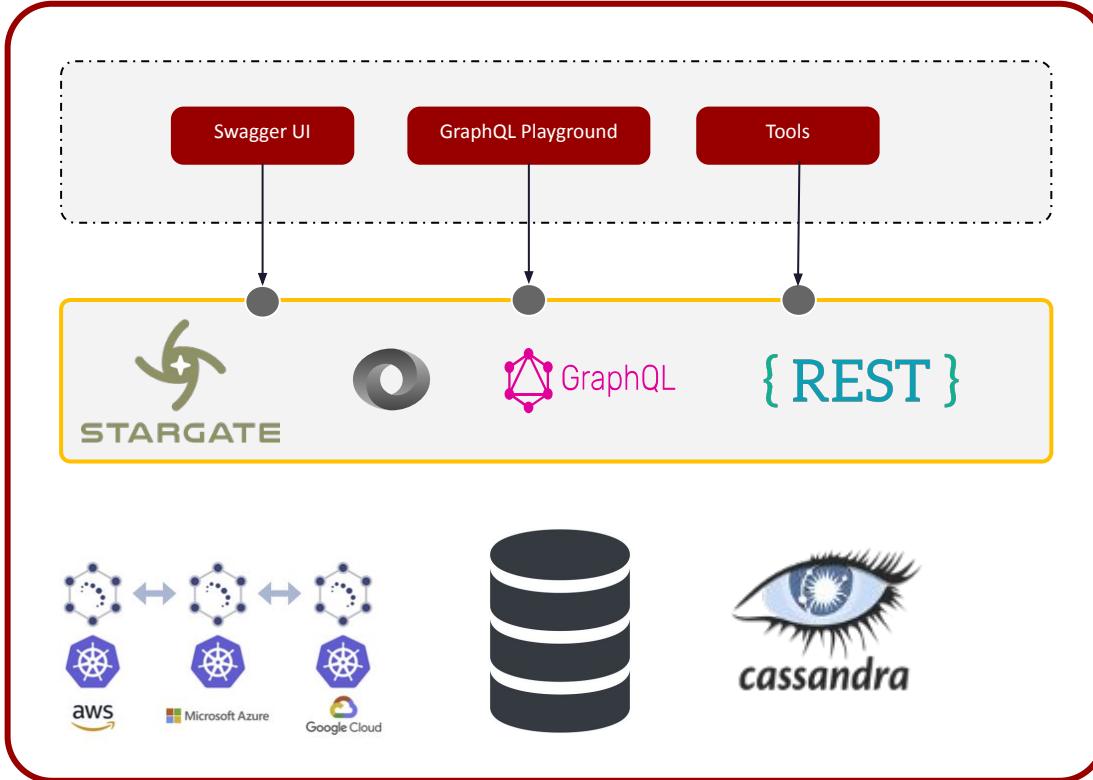


Connecting to your cluster (with Stargate)



Connecting to your cluster (with Stargates)





HouseKeeping

- Instructions and Slide
 - <https://github.com/datastaxdevs/conference-apidays2021-stargate>
- Runtime
 - <https://astra.datastax.com>

1. Create Astra Instance

ASTRA is the simplest way to run Cassandra with zero operations at all - just push the button and get your cluster. No credit card required, \$25.00 USD credit every month, roughly 5M writes, 30M reads, 40GB storage monthly - sufficient to run small production workloads.

Register (if needed) and Sign In to Astra <https://astra.datastax.com>: You can use your Github, Google accounts or register with an email.

Make sure to chose a password with minimum 8 characters, containing upper and lowercase letters, at least one number and special character

Create a "pay as you go" plan

Follow this [guide](#), to set up a pay as you go database with a free \$25 monthly credit.

- Select the pay as you go option: Includes \$25 monthly credit - no credit card needed to set up.

You will find below which values to enter for each field.

- For the database name - `free_db`. While Astra allows you to fill in these fields with values of your own choosing, please follow our recommendations to ensure the application runs properly.

- For the keyspace name - `free`. It's really important that you use the name "free" for the code to work.

You can technically use whatever you want and update the code to reflect the keyspace. This is really to get you on a happy path for the first run.

- For provider and region: Choose and provider (either GCP or AWS). Region is where your database will reside physically (choose one close to you or your users).

Hands-on

#1/2 Getting your instance

1. Create Astra Instance

Astra is the simplest way to run Cassandra with zero operations at all - just push the button and get your cluster. No credit card required, \$25.00 USD credit every month, roughly 5M writes, 30M reads, 40GB storage monthly - sufficient to run small production workloads.

- Register (if needed) and Sign In to Astra <https://astra.datastax.com>: You can use your Github, Google accounts or register with an email.

Make sure to choose a password with minimum 8 characters, containing upper and lowercase letters, at least one number and special character

- Create a "pay as you go" plan

Follow this [guide](#), to set up a pay as you go database with a free \$25 monthly credit.

- **Select the pay as you go option:** Includes \$25 monthly credit - no credit card needed to set up.

You will find below which values to enter for each field.

- **For the database name** - `free_db`. While Astra allows you to fill in these fields with values of your own choosing, please follow our recommendations to ensure the application runs properly.
- **For the keyspace name** - `keyspace1`. It's really important that you use the name "free" for the

2. Working with Cassandra

- Check that our keyspace exist

```
describe keyspaces;
```

- Create Entities

```
use keyspace1;
```

```
CREATE TYPE IF NOT EXISTS video_format (
    width   int,
    height  int
);
```

```
CREATE TABLE IF NOT EXISTS videos (
    videoid  uuid,
    title    text,
    upload   timestamp,
    email    text,
    url      text,
    tags     set <text>,
    frames   list<int>,
    formats  map <text,frozen<video_format>>,
    PRIMARY KEY (videoid)
);
```

```
describe keyspace1;
```

Agenda

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



Tooling
SDK, K8ssandra

05



What's NEXT ?
gRPC, CDC, sql...

Stargate “REST” Api(s)

- Expose existing CQL schema as a RESTful endpoint, or create new CQL schema
- Most familiar API style for most teams
- Swagger integration

schemas		
GET	/v1/keyspaces/{keyspaceName}/tables/{tableName}/columns	Retrieve all columns
POST	/v1/keyspaces/{keyspaceName}/tables/{tableName}/columns	Add a column
GET	/v1/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Retrieve a column
PUT	/v1/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Update a column
DELETE	/v1/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Delete a column
GET	/v1/keyspaces	Return all keyspaces
GET	/v1/keyspaces/{keyspaceName}/tables	Return all tables
POST	/v1/keyspaces/{keyspaceName}/tables	Add a table
GET	/v1/keyspaces/{keyspaceName}/tables/{tableName}	Return a table
DELETE	/v1/keyspaces/{keyspaceName}/tables/{tableName}	Delete a table
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns	Get all columns

Quick start: https://stargate.io/docs/stargate/1.0/quickstart/quick_start-rest.html

REST Api(s) exposed DDL and best practices DML

GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns	Get all columns
POST	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns	Create a column
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Get a column
PUT	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Update a column
DELETE	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/columns/{columnName}	Delete a column
DELETE	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/indexes/{indexName}	Drop an index from keyspace
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/indexes	Get all indexes for a given table
POST	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}/indexes	Add an index to a table's column
GET	/v2/schemas/keyspaces/{keyspaceName}	Get a keyspace
DELETE	/v2/schemas/keyspaces/{keyspaceName}	Delete a keyspace
GET	/v2/schemas/keyspaces	Get all keyspaces
POST	/v2/schemas/keyspaces	Create a keyspace
GET	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}	Get a table
PUT	/v2/schemas/keyspaces/{keyspaceName}/tables/{tableName}	Replace a table definition

DDL (Schema)

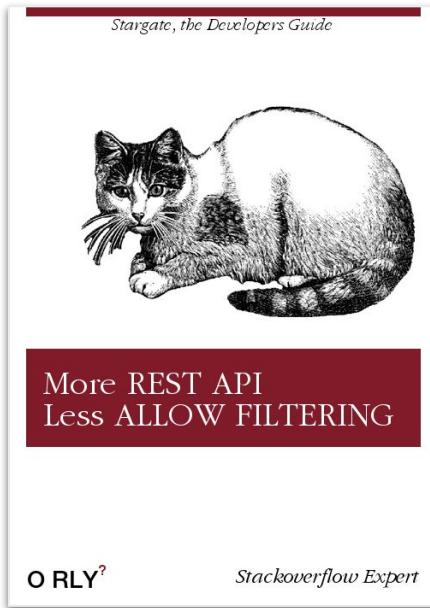
apidays

GET	/v2/keyspaces/{keyspaceName}/{tableName}/rows	Retrieve all rows
GET	/v2/keyspaces/{keyspaceName}/{tableName}	Search a table
POST	/v2/keyspaces/{keyspaceName}/{tableName}	Add row
GET	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Get row(s)
PUT	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Replace row(s)
DELETE	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Delete row(s)
PATCH	/v2/keyspaces/{keyspaceName}/{tableName}/{primaryKey}	Update part of a row(s)

DataStax Developers

Hands-on

#3 Working with Rest API



3. Working with REST API

To use the API we will need a token please create a token following the instructions here:

- [Create a token for your app](#) to use in the settings screen

Copy the token value (eg `AstraCS:KDfdKeNREyWQvDpDrBqwBsUB:ec80667c....`) in your clipboard and save the CSV this value would not be provided afterward.

Expected output

The screenshot shows the DataStax Astra UI. In the top navigation bar, there is a dropdown for 'Current Organization' set to 'cedrick.lunven@datast...'. Below the navigation, there are tabs for 'Dashboard', 'Databases', and 'Create Database'. Under 'Databases', a database named 'free_db' is selected. On the left, there is a 'Sample App Gallery' section with a link to 'Get started with a sample appl?'. In the center, under 'Application Tokens', there is a 'Generate a new token' button. A modal window is open, displaying the generated token details:

Client ID	Value
Client ID	<code>nVVMnBmBzmtTuzkonglmK</code>
Client Secret	<code>TnwTFjhOu1C1ZyTNPQqFT27D0EQ_X5Aig+...bhtXZOslnbj0Wic9y-KoYvKMfBrjJ3lobiSX2lMQuPSelN4xqndoDrRHNI_3_xQ+5ZcyrejOZRp0cGO+bG5.</code>
Token	<code>AstroCS:nVVMnBmBzmtTuzkonglmK#ff7b8de7410fe86dff779098cc89b6d61a6cfd539a2ceae8c072d2bbc4fb37e6</code>

At the bottom of the modal, there is a 'Download CSV' button.

Agenda

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



Tooling
SDK, K8ssandra

05



What's NEXT ?
gRPC, CDC, sql...

Cassandra as a Multi-Models NoSql Database



Column Oriented

- Cassandra is contains **Tables** with a Partition key
- Defined as Column Oriented (you can add columns)



Key-Value

- This is like KEY and multiple values, fit the key-value pattern



TimeSeries DataBase

- Cassandra fits time series use cases (high throughput) and data models (CL)
- Provide: TTL, aggregation through Spark, split OLTP and OLAP

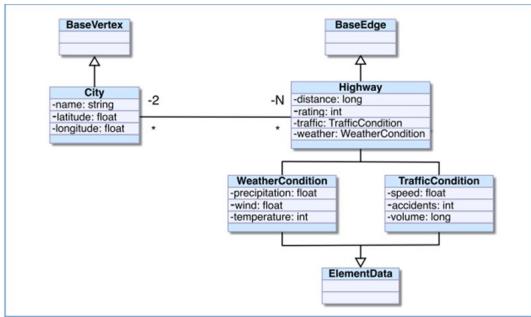


Graph DataBase

25

- For 5 years Datastax has been implemented a graph Engine, fork of TitanDB on top of Cassandra storage, querin with **gremlin**.

Working with JSON and Cassandra

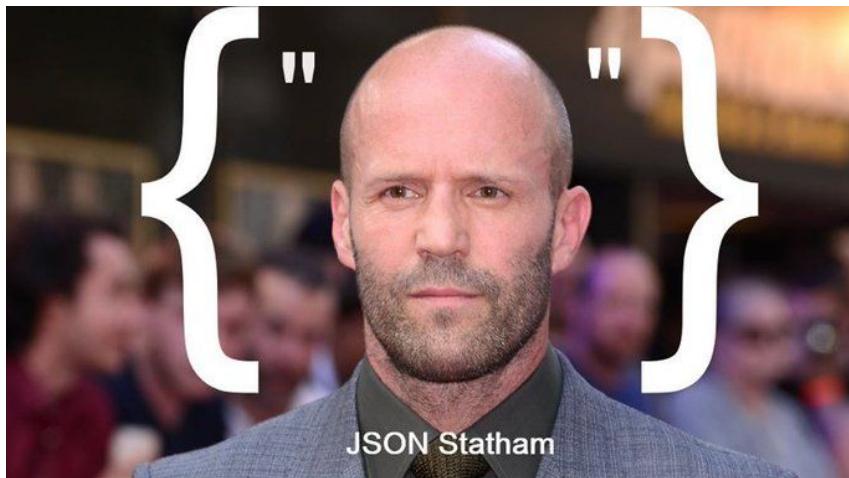


- Cassandra already handles both **JSON** and **nested objects**
 - INSERT JSON, SELECT JSON
 - Set<>, List<>, Map<>, and User Defined type (UDT) even nested
- But...
 - Strongly coupled with a SCHEMA VALIDATION
 - Dangerous with tombstones on updates

```
select json title,url,tags from videos;
```

```
INSERT INTO videos JSON '{
    "videoid": "e466f561-4ea4-4eb7-8dcc-126e0fbfd578",
    "email": "clunven@example.com",
    "title": "A JSON videos",
    "upload": "2020-02-26 15:09:22 +00:00",
    "url": "http://google.fr",
    "frames": [1,2,3,4],
    "tags": [ "cassandra", "accelerate", "2020"]
}';
```

“Schemaless”



“What rules ?”

- You want to insert and retrieve any JSON documents efficiently
- Allow “schemaless” (Validation less)
- Write to a single document is a single batch of statements
- Read from a single document is a single SELECT statement.
- Limit Tombstones with range deletes

Stargate Document API

- Schemaless!
(validationless)
- Add any JSON document
to a collection
- Uses “Document
shredding” approach

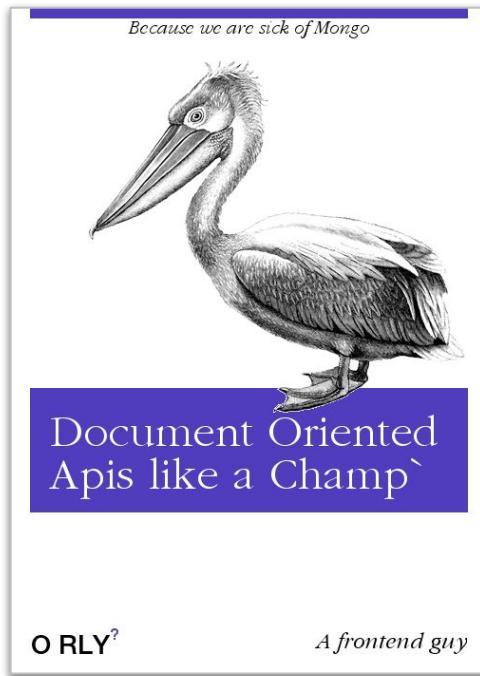
The screenshot shows the Stargate Document API documentation generated by Swagger. At the top, there's a header with the Stargate logo, the text "Supported by SMARTBEAR", the URL "/swagger.json", and a green "Explore" button. Below the header, the title "documents" is displayed. A list of API endpoints is shown, each with a color-coded button indicating the HTTP method and a link to the detailed API description. The endpoints are:

- GET /v2/namespaces/{namespace-id}/collections List collections in namespace
- POST /v2/namespaces/{namespace-id}/collections Create a new empty collection in a namespace
- GET /v2/namespaces/{namespace-id}/collections/{collection-id} Search documents in a collection
- POST /v2/namespaces/{namespace-id}/collections/{collection-id} Create a new document
- DELETE /v2/namespaces/{namespace-id}/collections/{collection-id} Delete a collection in a namespace
- POST /v2/namespaces/{namespace-id}/collections/{collection-id}/upgrade Upgrade a collection in a namespace
- POST /v2/namespaces/{namespace-id}/collections/{collection-id}/batch Write multiple documents in one request
- GET /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id} Get a document
- PUT /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id} Create or update a document with the provided document-id
- DELETE /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id} Delete a document
- PATCH /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id} Update data at the root of a document
- GET /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Get a path in a document
- PUT /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Replace data at a path in a document
- DELETE /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Delete a path in a document
- PATCH /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Update data at a path in a document
- GET /v2/namespaces/{namespace-id}/collections/{collection-id}/json-schema Get a JSON schema from a collection

Quick start: https://stargate.io/docs/stargate/1.0/quickstart/quick_start-document.html

Hands-on

#3 Working with Document API



4. Working with DOCUMENT API

This walkthrough has been realized using the [Quick Start](#)

locate the Document part in the Swagger UI

The screenshot shows the Swagger UI interface for the Document API. At the top, there's a navigation bar with the Swagger logo, a link to /swagger.json, and a green 'Explore' button. Below the navigation, the title 'documents' is displayed. A list of API endpoints is shown, each with a color-coded button indicating the HTTP method and a detailed description:

- GET /v2/namespaces/{namespace-id}/collections List collections in namespace
- POST /v2/namespaces/{namespace-id}/collections Create a new empty collection in a namespace
- GET /v2/namespaces/{namespace-id}/collections/{collection-id} Search documents in a collection
- POST /v2/namespaces/{namespace-id}/collections/{collection-id} Create a new document
- DELETE /v2/namespaces/{namespace-id}/collections/{collection-id} Delete a collection in a namespace
- POST /v2/namespaces/{namespace-id}/collections/{collection-id}/upgrade Upgrade a collection in a namespace
- GET /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Get a path in a document
- PUT /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Replace data at a path in a document
- DELETE /v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path} Delete a path in a document

At the bottom of the list, there is a note with a green checkmark: "Creating a namespace :".

Agenda (40 min)

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



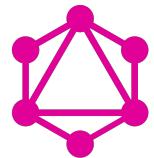
Tooling
SDK, K8ssandra

05



What's NEXT ?
gRPC, CDC, sql...

History



GraphQL

- 2012 Created by Facebook
Used internally for mobile apps
- 2015 Facebook give talk at ReactJs Conf
- 2015 Facebook open sourced GraphQL
- 2016 Github announced move to GQL

Definition

GraphQL is an application programming interface (API) query language and server-side runtime that prioritises giving customers precisely the data they request.

<https://landscape.graphql.org/?category=graph-ql-adopter&grouping=category&style=borderless>

Why ?



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

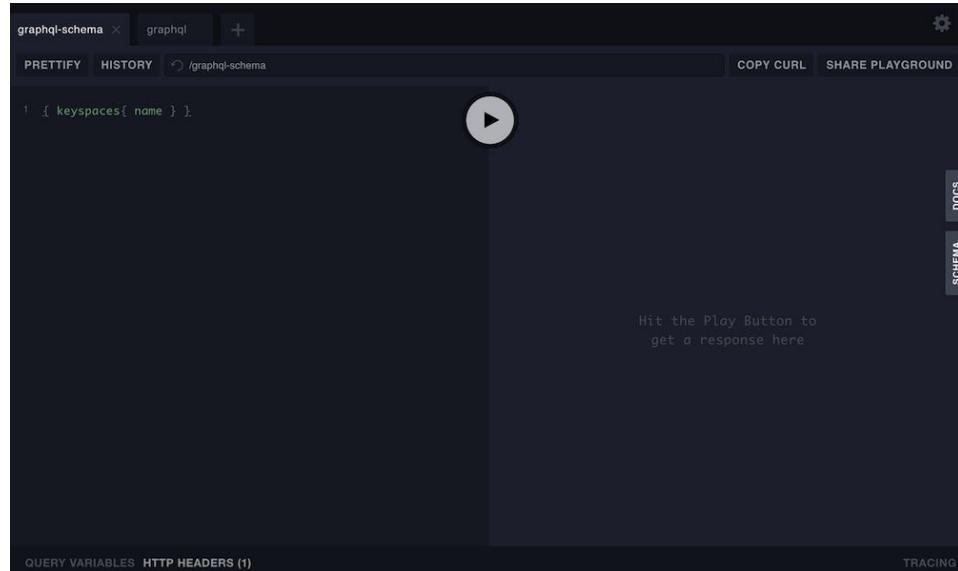
```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

Stargate GraphQL API

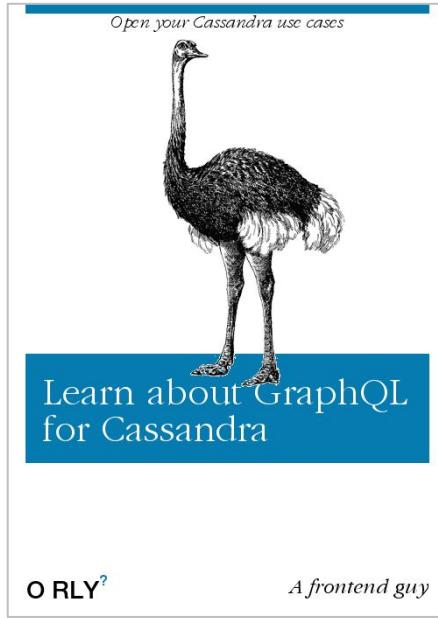
- Expose existing CQL schema as a GraphQL endpoint
- Allows more fine-grained control of what fields returned
- In progress: federation across multiple services to join data
- Easy experimentation in GraphQL playground



Quick start: https://stargate.io/docs/stargate/1.0/quickstart/quick_start-graphql.html

Hands-on

#5 Working with GraphQL API



5. Working with GRAPHQL API

This walkthrough has been realized using the [GraphQL Quick Start](#)

Open GraphQL Playground :

Open the playground

Using the GraphQL API to connect to your database

The GraphQL API allows you to interact with your data using GraphQL types, queries, and mutations. For every table in the database, a series of GraphQL objects are generated, along with queries and mutations that allow you to search and modify the table data.

Prerequisites

1. An Application token (create a new one here) with the appropriate role set (API Admin User is needed for example below).

Launching GraphQL Playground

1. Open the GraphQL Playground URL or <https://f03087a4-5c9c-4618-9dd1-98c5eb81bc5c-europe-west1.apps.astra.datasax.com/graphqlplayground>
2. Add your Token from your Application Token to the HEADER section at the bottom of the GraphQL Playground window:
{"x-cassandra-token": "<>Token<>"}

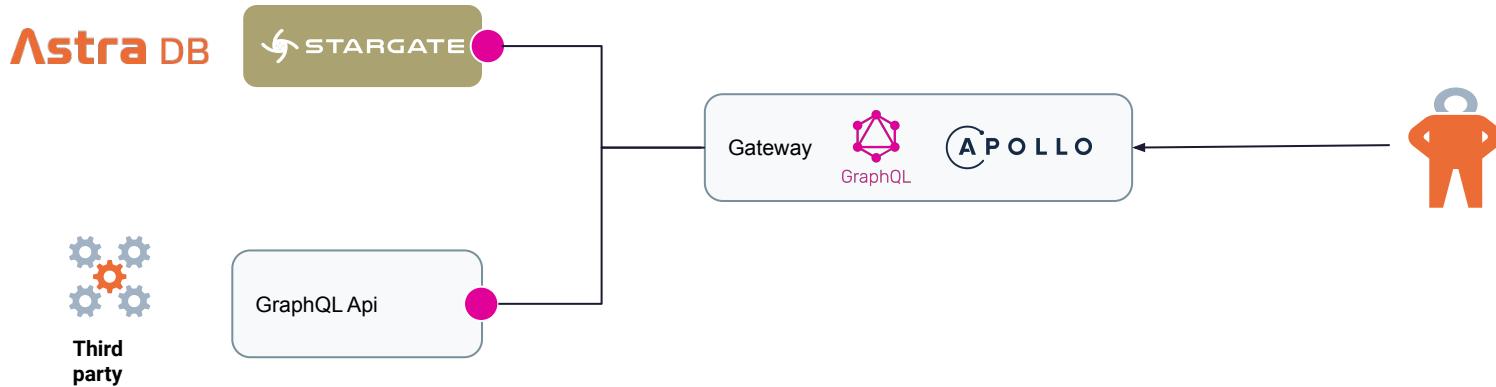
2. Replace the current GraphQL Playground URL with the URL for your region: <https://f03087a4-5c9c-4618-9dd1-98c5eb81bc5c-europe-west1.apps.astra.datasax.com/graphqlschema>

Expected output

graphql-schema

graphql

GraphQL Federation



Agenda (40 min)

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



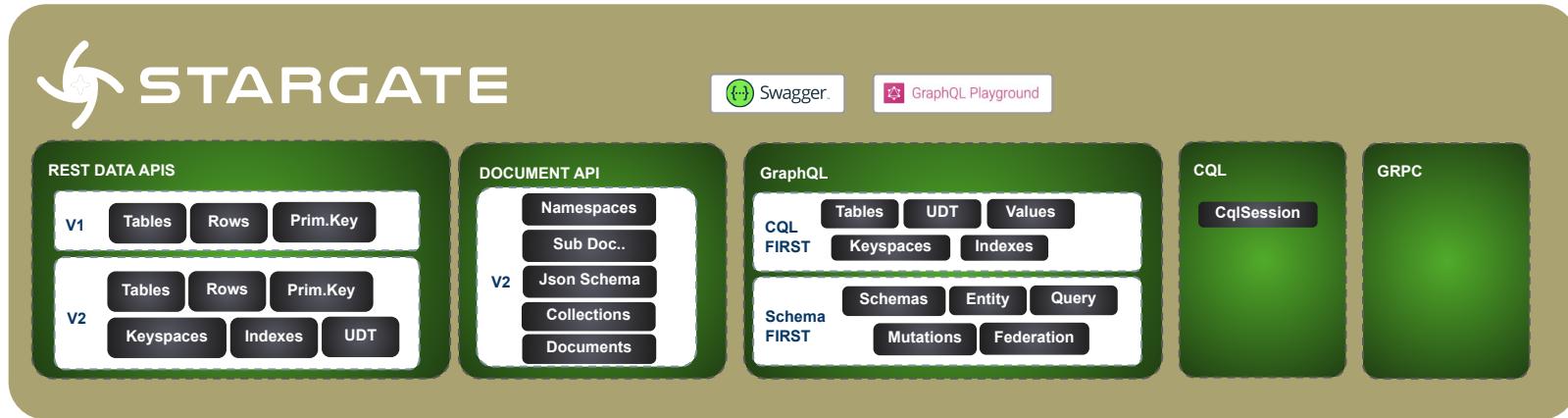
Tooling
SDK, K8ssandra

05

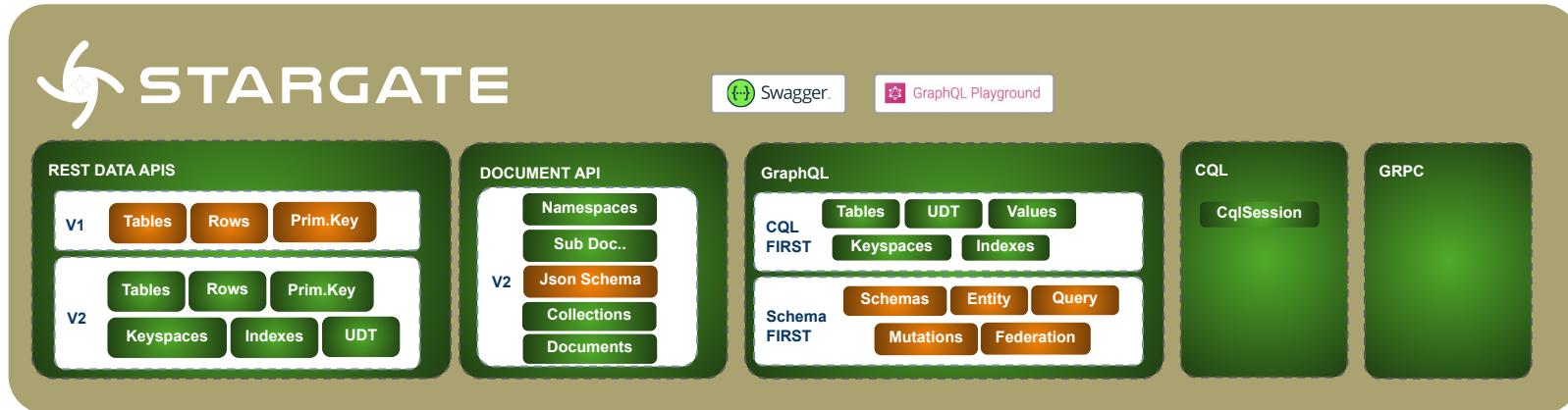


What's NEXT ?
gRPC, CDC, sql...

Features Map



SDK in Java, Javascript, Python



Sample Codes



```
StargateClient client = StargateClient.builder()
    .username("k8ssandra-superuser")           // Mandatory username
    .password("JxzrPOnvDGqfEOQ0EySQ")          // Mandatory password
    .endPointAuth("http://localhost:8081")       // Mandatory authentication url, defaulting to http://localhost
    .endPointRest("http://localhost:8082")        // Rest and Document APIs
    .endPointGraphQL("http://localhost:8080")     // GraphQL API

    // Cqlsession Only
    .addCqlContactPoint("127.0.0.", 9042)      // Contact Point
    .localDc("dc1")                            // Local Datacenter is mandatory driver 4xx+
    .keypace("ks1")                            // (optional) Set your keyspace
    .build();
```

Sample Code

```
// Retrieve an object and marshall
Optional<Address> address = colPersonClient
    .document("e8c5021b-2c91-4015-aec6-14a16e449818")
    .findSubDocument("address", Address.class);

// Building query {"age": {"$gte":30}, "lastname": {"$eq": "PersonAstra2"}}
SearchDocumentQuery query = SearchDocumentQuery.builder()
    .where("age").isGreaterOrEqualsThan(30)      // First filter to use where()
    .and("lastname").isEqualTo("PersonAstra2")   // Any extra filter to use and()
    .withPageSize(10)                            // Default and max pageSize are 20
    .build();

// Retrieve PAGE 1
DocumentResultPage<Person> currentPage = colPersonClient.findPage(query, Person.class);

// Retrieve PAGE 2 (if any)
if (currentPage.getPageState().isPresent()) {
    query.setPageState(currentPage.getPageState().get());
}
DocumentResultPage<Person> nextPage = colPersonClient.findPage(query, Person.class);

// Retrieve all documents in one call (with warning in RED ABOVE)
Stream<Person> allPersons = colPersonClient.findAll(query, Person.class)
```

Agenda (40 min)

01



Stargate Data gateway
What, Why and How

02 {REST}

Exploring Apis
Rest DML and DDL

03



Exploring Apis
Document oriented

04



Exploring Apis
GraphQL and Federation

05



Tooling
SDK, K8ssandra

05



What's NEXT ?
gRPC, CDC, sql...

What's NEXT ?

- **Apis**
 - GRPC (GA next month)
 - Relational API
 - Dynamo API
- **Architecture**
 - SSO, KMS
 - Architecture redesigned
 - ZDD
 - CDC



<https://www.datastax.com/workshops>

Weekly Workshops

The screenshot shows the DataStax Developers YouTube channel page. At the top, there's a large banner with the text "LEVEL UP with the DataStax Developers". Below the banner, the channel name "DataStax Developers" and "8,1 k abonnés" are displayed. The main navigation menu includes "ACCUEIL", "VIDÉOS" (which is underlined), "PLAYLISTS", "COMMUNAUTÉ", "CHAÎNES", and "À PROPOS". A search bar is also present. The "VIDÉOS" section lists several video thumbnails with titles like "Microservices with Cassandra + Spring", "Advanced Data Modeling in Apache Cassandra™", and "Polka DataOps Meetup: Introduction to Apache...". Each thumbnail includes a duration (e.g., 2:23:56, 2:41:51, 1:28:14) and a "REGARDER VIDÉO" button. The overall theme is white and blue.

The screenshot shows the DataStax Developers website with a dark header featuring a "SUBSCRIBE" button and the text "for weekly content on building". Below the header, there's a section titled "Upcoming Live Events" with five event cards. Each card has a "LIVE hands-on workshop </>" icon and a DataStax developer character illustration. The events are:

- Apache Cassandra™ Certification Preparation** (Multiple Dates | NoSQL | Beginner)
- Build Microservices with Apache Cassandra™!** (Feb 17 or Feb 18 | NoSQL | Beginner)
- Certification Exam Preparation Workshop** (Multiple Dates)
- Cloud-Native Workshop: Build Spring Microservices with Apache Cassandra™** (Multiple Dates)
- Learn how to build a Serverless Game!** (Feb 24 or Feb 25 | Game Development | Beginner)
- Build Microservices with Cassandra & Quarkus** (March 11 | Microservices | Beginner)
- Cloud-Native Workshop: Build Microservices w/ Apache Cassandra™ + Quarkus!** (THU MAR 11 2021 | Cloud-Native | Beginner)

Each event card includes a "Register Now" button. The overall design is clean with a dark background and white text.

apidays

<https://bit.ly/cassandra-workshop>

Join our 10k Discord Community

The Fellowship of the RINGS

main-chat-room | <https://www.youtube.com/watch?v=huJ6OfzWMt4%3E> <https://forms.gle/EtM6g1C...>

thank you everyone! we hope to see you again in a future workshop! 🎉

Jack Fryer Aujourd'hui à 14:30
NEW WORKSHOP ALERT

Cassandra meets Kubernetes!

Come and meet K8ssandra! A cloud-native distribution of Apache Cassandra™ built for running on Kubernetes

<https://www.eventbrite.co.uk/e/cloud-native-workshop-connecting-cassandra-and-kubernetes-tickets-142078180663>

Eventbrite

Cloud-Native Workshop: Connecting Cassandra and Kubernetes!

Come and meet K8ssandra! A cloud-native distribution of Apache Cassandra™ built for running on Kubernetes

LIVE hands-on workshop </>

Apache Cassandra™ meets Kubernetes!

March 3 or March 4 | Cloud-native | Beginner

LEVEL UP Developers

Cedrick Lunven Aujourd'hui à 16:18
Hey Community I will be live tonight for 50 min talk on Spring + Cassandra to the Virtual Java User Group

<https://www.youtube.com/watch?v=nuyPKDQn1gl> come and say hi ?

Envoyer un message à #main-chat-room

PRESENTER—3

- Aleks Volochnev
- David Jones-Gilardi
- jscarp

HELPER—1

- John Sanda

EN LIGNE—222

- Abhiprada
- Absurdism
- hiya
- Adalberto
- aditya_dhunna
- adnaneCord
- Adrigunz
- Aemilius Gaurav
- Aemilius gaurav
- Aguvas
- ajscilingo
- akashTaxvisor

DataStax Developers

Thank you!



GitHub

@clun



@clunven

