

Kafka REST API SwaggerUI

Table of Contents

| | |
|---|----|
| 1. Overview | 1 |
| 1.1. Version information | 1 |
| 1.2. Contact information..... | 1 |
| 1.3. URI scheme..... | 1 |
| 1.4. Tags | 1 |
| 2. Chapter of manual content 1 | 2 |
| 2.1. Sub chapter | 2 |
| 3. Chapter of manual content 2 | 3 |
| 4. Resources | 4 |
| 4.1. Kafka-controller | 4 |
| 4.1.1. List brokers in this cluster | 4 |
| 4.1.2. Get the message from the offset of the partition in the topic, decoder is not supported yet .. | 4 |
| 4.1.3. Delete old Consumer Group | 5 |
| 4.1.4. getLastCommitTimestamp | 6 |
| 4.1.5. Reset consumer group offset, earliest/latest can be used | 7 |
| 4.1.6. List consumer groups from zk and kafka | 8 |
| 4.1.7. Describe consumer groups, showing lag and offset | 8 |
| 4.1.8. Add a partition to the topic..... | 9 |
| 4.1.9. Check the partition reassignment process | 10 |
| 4.1.10. Execute the partition reassignment..... | 11 |
| 4.1.11. Generate plan for the partition reassignment..... | 12 |
| 4.1.12. List topics | 12 |
| 4.1.13. Create a topic..... | 13 |
| 4.1.14. Describe a topic by fetching the metadata and config | 14 |
| 4.1.15. Delete a topic (you should enable topic deletion | 14 |
| 4.1.16. Create topic configs | 15 |
| 4.1.17. Get topic configs | 16 |
| 4.1.18. Update topic configs..... | 16 |
| 4.1.19. Delete topic configs | 17 |
| 4.1.20. Get topic config by key | 18 |
| 4.1.21. Delete a topic config by key | 19 |
| 4.1.22. Create a topic config by key | 19 |
| 4.1.23. Update a topic config by key | 20 |
| 4.1.24. Tell if a topic exists..... | 21 |
| 4.1.25. Write a message to the topic, for testing purpose | 22 |
| 4.1.26. List topics Brief | 23 |

| | |
|---|----|
| 4.2. Zookeeper-controller | 23 |
| 4.2.1. Get the connection state of zookeeper | 23 |
| 4.2.2. Get the environment information of zookeeper | 24 |
| 4.2.3. List a zookeeper path | 24 |
| 4.2.4. Get the service state of zookeeper | 25 |
| 5. Definitions | 26 |
| 5.1. AddPartition | 26 |
| 5.2. BrokerInfo | 26 |
| 5.3. ConsumerGroupDesc | 26 |
| 5.4. GeneralResponse | 27 |
| 5.5. HostAndPort | 27 |
| 5.6. Map«int,long» | 28 |
| 5.7. ReassignWrapper | 28 |
| 5.8. TopicAndPartition | 28 |
| 5.9. TopicBrief | 28 |
| 5.10. TopicDetail | 28 |
| 5.11. TopicMeta | 29 |
| 5.12. TopicPartitionInfo | 29 |
| 5.13. ZkServerClient | 29 |
| 5.14. ZkServerEnvironment | 30 |
| 5.15. ZkServerStat | 30 |

Chapter 1. Overview

Kafka REST API SwaggerUI

1.1. Version information

Version : 0.1.0

1.2. Contact information

Contact : gnuhpc

Contact Email : gnuahpc@gmail.com

1.3. URI scheme

Host : localhost:8080

BasePath : /

1.4. Tags

- kafka-controller : Kafka Controller
- zookeeper-controller : Zookeeper Controller

Chapter 2. Chapter of manual content 1

This is some dummy text

2.1. Sub chapter

Dummy text of sub chapter

Chapter 3. Chapter of manual content 2

This is some dummy text

Chapter 4. Resources

4.1. Kafka-controller

Kafka Controller

4.1.1. List brokers in this cluster

```
GET /kafka/brokers
```

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|--------------------------------------|
| 200 | OK | < BrokerInfo > array |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.2. Get the message from the offset of the partition in the topic, decoder is not supported yet

```
GET /kafka/consumer/{topic}/{partition}/{offset}
```

Parameters

| Type | Name | Description | Schema |
|------|----------------------------------|-------------|----------------|
| Path | offset <i>required</i> | offset | integer(int64) |

| Type | Name | Description | Schema |
|-------|-------------------------------------|-------------|----------------|
| Path | partition <i>required</i> | partition | integer(int32) |
| Path | topic <i>required</i> | topic | string |
| Query | decoder <i>optional</i> | decoder | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------|
| 200 | OK | string |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.3. Delete old Consumer Group

```
DELETE /kafka/consumergroup/{consumergroup}
```

Parameters

| Type | Name | Description | Schema |
|------|---|---------------|--------|
| Path | consumergroup <i>required</i> | consumergroup | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------------|
| 200 | OK | GeneralResponse |
| 204 | No Content | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.4. getLastCommitTimestamp

```
GET /kafka/consumergroup/{consumergroup}/{topic}/lastcommittime
```

Parameters

| Type | Name | Description | Schema |
|------|---|---------------|--------|
| Path | consumergroup <i>required</i> | consumergroup | string |
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|--|
| 200 | OK | < string, < string, integer(int64) > map > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.5. Reset consumer group offset, earliest/latest can be used

```
PUT /kafka/consumergroup/{consumergroup}/{type}/{topic}/{partition}/{offset}
```

Parameters

| Type | Name | Description | Schema |
|------|---|---------------|----------------|
| Path | consumergroup <i>required</i> | consumergroup | string |
| Path | offset <i>required</i> | offset | string |
| Path | partition <i>required</i> | partition | integer(int32) |
| Path | topic <i>required</i> | topic | string |
| Path | type <i>required</i> | type | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------------|
| 200 | OK | GeneralResponse |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- /

4.1.6. List consumer groups from zk and kafka

```
GET /kafka/consumergroups
```

Parameters

| Type | Name | Description | Schema |
|-------|---------------------------------|-------------|--------|
| Query | topic <i>optional</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|----------------------------------|
| 200 | OK | < string, < string > array > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- *application/json*

Produces

- /

4.1.7. Describe consumer groups, showing lag and offset

```
GET /kafka/consumergroups/{consumerGroup}/{type}
```

Parameters

| Type | Name | Description | Schema |
|------|---|---------------|--------|
| Path | consumerGroup <i>required</i> | consumerGroup | string |

| Type | Name | Description | Schema |
|-------|---------------------------------|-------------|--------|
| Path | type <i>required</i> | type | string |
| Query | topic <i>optional</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---|
| 200 | OK | < string, < ConsumerGroupDesc > array > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- [application/json](#)

Produces

- [/](#)

4.1.8. Add a partition to the topic

POST /kafka/partitions/add

Parameters

| Type | Name | Description | Schema |
|------|--|--------------|------------------------------|
| Body | addPartition <i>required</i> | addPartition | AddPartition |

Responses

| HTTP Code | Description | Schema |
|-----------|-------------|---------------------------|
| 200 | OK | TopicMeta |

| HTTP Code | Description | Schema |
|-----------|--------------|------------|
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.9. Check the partition reassignment process

```
PUT /kafka/partitions/reassign/check
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------------|-------------|--------|
| Body | reassignStr <i>required</i> | reassignStr | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------------------|--------------------------------|
| -1 | Reassignment Failed | No Content |
| 0 | Reassignment In Progress | No Content |
| 1 | Reassignment Completed | No Content |
| 200 | OK | < string, integer(int32) > map |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |

| HTTP Code | Description | Schema |
|-----------|-------------|------------|
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.10. Execute the partition reassignment

```
PUT /kafka/partitions/reassign/execute
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------------|-------------|--------|
| Body | reassignStr <i>required</i> | reassignStr | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|--------------------------------|
| 200 | OK | < string, integer(int32) > map |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.11. Generate plan for the partition reassignment

POST /kafka/partitions/reassign/generate

Parameters

| Type | Name | Description | Schema |
|------|---|-----------------|---------------------------------|
| Body | reassignWrapper <i>required</i> | reassignWrapper | ReassignWrapper |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------|
| 200 | OK | < string > array |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.12. List topics

GET /kafka/topics

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------|
| 200 | OK | < string > array |
| 401 | Unauthorized | No Content |

| HTTP Code | Description | Schema |
|-----------|-------------|------------|
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.13. Create a topic

POST /kafka/topics/create

Parameters

| Type | Name | Description | Schema |
|-------|---------------------------------------|-------------|-----------------------------|
| Query | reassignStr <i>optional</i> | reassignStr | string |
| Body | topic <i>required</i> | topic | TopicDetail |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------|
| 201 | Created | TopicMeta |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.14. Describe a topic by fetching the metadata and config

```
GET /kafka/topics/{topic}
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------|
| 200 | OK | TopicMeta |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.15. Delete a topic (you should enable topic deletion)

```
DELETE /kafka/topics/{topic}
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------------|
| 200 | OK | GeneralResponse |
| 204 | No Content | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.16. Create topic configs

POST /kafka/topics/{topic}/conf

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|------------------------|
| Path | topic <i>required</i> | topic | string |
| Body | prop <i>required</i> | prop | < string, object > map |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------------|
| 200 | OK | < string, object > map |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.17. Get topic configs

```
GET /kafka/topics/{topic}/conf
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------------|
| 200 | OK | < string, object > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.18. Update topic configs

```
PUT /kafka/topics/{topic}/conf
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|------------------------|
| Path | topic <i>required</i> | topic | string |
| Body | prop <i>required</i> | prop | < string, object > map |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------------|
| 200 | OK | < string, object > map |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.19. Delete topic configs

```
DELETE /kafka/topics/{topic}/conf
```

Parameters

| Type | Name | Description | Schema |
|------|------------------------------------|-------------|------------------|
| Path | topic <i>required</i> | topic | string |
| Body | delProps <i>required</i> | delProps | < string > array |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------------|
| 200 | OK | < string, object > map |
| 204 | No Content | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.20. Get topic config by key

```
GET /kafka/topics/{topic}/conf/{key}
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | key <i>required</i> | key | string |
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------------|
| 200 | OK | < string, object > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.21. Delete a topic config by key

```
DELETE /kafka/topics/{topic}/conf/{key}
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | key <i>required</i> | key | string |
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------|
| 200 | OK | boolean |
| 204 | No Content | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.22. Create a topic config by key

```
POST /kafka/topics/{topic}/conf/{key}={value}
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | key <i>required</i> | key | string |
| Path | topic <i>required</i> | topic | string |
| Path | value <i>required</i> | value | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------|
| 200 | OK | < string, object > map |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.23. Update a topic config by key

```
PUT /kafka/topics/{topic}/conf/{key}={value}
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | key <i>required</i> | key | string |
| Path | topic <i>required</i> | topic | string |

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | value <i>required</i> | value | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------------|
| 200 | OK | < string, object > map |
| 201 | Created | No Content |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.24. Tell if a topic exists

```
GET /kafka/topics/{topic}/exist
```

Parameters

| Type | Name | Description | Schema |
|------|---------------------------------|-------------|--------|
| Path | topic <i>required</i> | topic | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------|
| 200 | OK | boolean |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |

| HTTP Code | Description | Schema |
|-----------|-------------|------------|
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.1.25. Write a message to the topic, for testing purpose

```
POST /kafka/topics/{topic}/write
```

Parameters

| Type | Name | Description | Schema |
|------|-----------------------------------|-------------|--------|
| Path | topic <i>required</i> | topic | string |
| Body | message <i>required</i> | message | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---------------------------------|
| 201 | Created | GeneralResponse |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `text/plain`

Produces

- `/`

4.1.26. List topics Brief

```
GET /kafka/topicsbrief
```

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|--------------------------------------|
| 200 | OK | < TopicBrief > array |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.2. Zookeeper-controller

Zookeeper Controller

4.2.1. Get the connection state of zookeeper

```
GET /zk/connstate
```

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------|
| 200 | OK | string |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.2.2. Get the environment information of zookeeper

```
GET /zk/env
```

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|---|
| 200 | OK | < string, ZkServerEnvironment > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.2.3. List a zookeeper path

```
GET /zk/ls/{path}
```

Parameters

| Type | Name | Description | Schema |
|------|--------------------------------|-------------|--------|
| Path | path <i>required</i> | path | string |

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|------------------|
| 200 | OK | < string > array |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

4.2.4. Get the service state of zookeeper

```
GET /zk/stat
```

Responses

| HTTP Code | Description | Schema |
|-----------|--------------|--|
| 200 | OK | < string, ZkServerStat > map |
| 401 | Unauthorized | No Content |
| 403 | Forbidden | No Content |
| 404 | Not Found | No Content |

Consumes

- `application/json`

Produces

- `/`

Chapter 5. Definitions

5.1. AddPartition

| Name | Schema |
|--|----------------|
| numPartitionsAdded <i>optional</i> | integer(int32) |
| replicaAssignment <i>optional</i> | string |
| topic <i>optional</i> | string |

5.2. BrokerInfo

| Name | Schema |
|--|-------------------|
| endPoints <i>optional</i> | < string > array |
| host <i>optional</i> | string |
| id <i>optional</i> | integer(int32) |
| jmxPort <i>optional</i> | integer(int32) |
| port <i>optional</i> | integer(int32) |
| rack <i>optional</i> | string |
| securityProtocol <i>optional</i> | object |
| startTime <i>optional</i> | string(date-time) |
| version <i>optional</i> | integer(int32) |

5.3. ConsumerGroupDesc

| Name | Schema |
|---|-------------------------|
| consumerId <i>optional</i> | string |
| currentOffset <i>optional</i> | integer(int64) |
| groupName <i>optional</i> | string |
| host <i>optional</i> | string |
| lag <i>optional</i> | integer(int64) |
| logEndOffset <i>optional</i> | integer(int64) |
| partitionId <i>optional</i> | integer(int32) |
| state <i>optional</i> | enum (RUNNING, PENDING) |
| topic <i>optional</i> | string |
| type <i>optional</i> | enum (NEW, OLD) |

5.4. GeneralResponse

| Name | Schema |
|---------------------------------|-------------------------|
| msg <i>optional</i> | string |
| state <i>optional</i> | enum (success, failure) |

5.5. HostAndPort

| Name | Schema |
|------------------------------------|----------------|
| hostText <i>optional</i> | string |
| port <i>optional</i> | integer(int32) |

5.6. Map«int,long»

Type : < string, integer(int64) > map

5.7. ReassignWrapper

| Name | Schema |
|-----------------------------------|--------------------------|
| brokers <i>optional</i> | < integer(int32) > array |
| topics <i>optional</i> | < string > array |

5.8. TopicAndPartition

Type : object

5.9. TopicBrief

| Name | Schema |
|--|----------------|
| isrRate <i>optional</i> | integer(int64) |
| numPartition <i>optional</i> | integer(int32) |
| topic <i>optional</i> | string |

5.10. TopicDetail

| Name | Schema |
|--------------------------------------|------------------------|
| factor <i>optional</i> | integer(int32) |
| name <i>optional</i> | string |
| partitions <i>optional</i> | integer(int32) |
| prop <i>optional</i> | < string, object > map |

5.11. TopicMeta

| Name | Schema |
|---|--|
| partitionCount <i>optional</i> | integer(int32) |
| replicationFactor <i>optional</i> | integer(int32) |
| topicCustomConfigs <i>optional</i> | < string, object > map |
| topicName <i>optional</i> | string |
| topicPartitionInfos <i>optional</i> | < TopicPartitionInfo > array |

5.12. TopicPartitionInfo

| Name | Schema |
|--|------------------|
| endOffset <i>optional</i> | integer(int64) |
| in_sync <i>optional</i> | boolean |
| isr <i>optional</i> | < string > array |
| leader <i>optional</i> | string |
| messageAvailable <i>optional</i> | integer(int64) |
| partitionId <i>optional</i> | integer(int32) |
| replicas <i>optional</i> | < string > array |
| startOffset <i>optional</i> | integer(int64) |

5.13. ZkServerClient

| Name | Schema |
|------------------------------------|----------------|
| host <i>optional</i> | string |
| ops <i>optional</i> | integer(int32) |
| port <i>optional</i> | integer(int32) |
| queued <i>optional</i> | integer(int32) |
| received <i>optional</i> | integer(int32) |
| sent <i>optional</i> | integer(int32) |

5.14. ZkServerEnvironment

| Name | Schema |
|--------------------------------------|------------------------|
| attributes <i>optional</i> | < string, string > map |

5.15. ZkServerStat

| Name | Schema |
|---------------------------------------|--|
| avgLatency <i>optional</i> | integer(int32) |
| buildDate <i>optional</i> | string |
| clients <i>optional</i> | < ZkServerClient > array |
| connections <i>optional</i> | integer(int32) |
| maxLatency <i>optional</i> | integer(int32) |
| minLatency <i>optional</i> | integer(int32) |
| mode <i>optional</i> | enum (Leader, Follower, Observer) |

| Name | Schema |
|---------------------------------------|----------------|
| nodes <i>optional</i> | integer(int32) |
| outstanding <i>optional</i> | integer(int32) |
| received <i>optional</i> | integer(int32) |
| sent <i>optional</i> | integer(int32) |
| version <i>optional</i> | string |
| zxid <i>optional</i> | string |