

DBA Service 对外 API

Table of Contents

- Overview 1
 - Version information 1
 - Contact information 1
 - URI scheme 1
 - Tags 1
- Paths 2
 - 持久化——指定Executor ID的Executor(执行器) 2
 - Description 2
 - Parameters 2
 - Responses 2
 - Consumes 2
 - Produces 2
 - Tags 2
 - 持久化——指定Executor ID的Executor(执行器) 2
 - Description 2
 - Parameters 3
 - Responses 3
 - Produces 3
 - Tags 3
 - 持久化——所有InceptorServer 3
 - Description 3
 - Parameters 3
 - Responses 3
 - Consumes 4
 - Produces 4
 - Tags 4
 - 持久化——所有InceptorServer 4
 - Description 4
 - Parameters 4
 - Responses 4
 - Produces 4
 - Tags 4
 - 返回InceptorServer的统计数据 5
 - Description 5
 - Responses 5
 - Consumes 5
 - Produces 5
 - Tags 5
 - 返回InceptorServer的统计数据 5
 - Description 5
 - Responses 5
 - Produces 5
 - Tags 6
 - 持久化——指定Query ID的Query(查询) 6
 - Description 6
 - Parameters 6
 - Responses 6
 - Consumes 6
 - Produces 6
 - Tags 6
 - 持久化——指定Query ID的Query(查询) 6
 - Description 6
 - Parameters 7
 - Responses 7
 - Produces 7
 - Tags 7
 - 返回Query(查询)的统计数据 7
 - Description 7
 - Responses 7
 - Consumes 7

Produce	8
Tags	8
返回Query(查询)的统计数据	8
Description	8
Responses	8
Produce	8
Tags	8
Inceptor——Inceptor的Environment(环境配置)状态	8
Description	8
Parameters	8
Responses	9
Consumes	9
Produce	9
Tags	9
Inceptor——Inceptor的Environment(环境配置)状态	9
Description	9
Parameters	9
Responses	9
Produce	10
Tags	10
Inceptor——指定Executor ID的Executor(执行器)	10
Description	10
Parameters	10
Responses	10
Consumes	10
Produce	10
Tags	10
Inceptor——指定Executor ID的Executor(执行器)	10
Description	11
Parameters	11
Responses	11
Produce	11
Tags	11
Inceptor——Executors(执行器)	11
Description	11
Parameters	11
Responses	12
Consumes	12
Produce	12
Tags	12
Inceptor——Executors(执行器)	12
Description	12
Parameters	12
Responses	12
Produce	13
Tags	13
Inceptor——指定Job ID的Job(作业)详细信息	13
Description	13
Parameters	13
Responses	13
Consumes	13
Produce	13
Tags	13
Inceptor——指定Job ID的Job(作业)详细信息	13
Description	14
Parameters	14
Responses	14
Produce	14
Tags	14
Inceptor——所有Jobs(作业)	14
Description	14
Parameters	14
Responses	15

Consumes	15
Produces	15
Tags	15
Inceptor——所有Jobs(作业)	15
Description	15
Parameters	15
Responses	15
Produces	16
Tags	16
Inceptor——Overview数据	16
Description	16
Parameters	16
Responses	16
Consumes	16
Produces	16
Tags	16
Inceptor——Overview数据	16
Description	17
Parameters	17
Responses	17
Produces	17
Tags	17
Inceptor——InceptorServer所有数据	17
Description	17
Parameters	17
Responses	18
Consumes	18
Produces	18
Tags	18
Inceptor——InceptorServer所有数据	18
Description	18
Parameters	18
Responses	18
Produces	19
Tags	19
Inceptor——所有Sessions(会话)	19
Description	19
Parameters	19
Responses	19
Consumes	19
Produces	19
Tags	19
Inceptor——所有Sessions(会话)	19
Description	20
Parameters	20
Responses	20
Produces	20
Tags	20
Inceptor——指定Query ID的Query(查询)	20
Description	20
Parameters	20
Responses	21
Consumes	21
Produces	21
Tags	21
Inceptor——指定Query ID的Query(查询)	21
Description	21
Parameters	21
Responses	21
Produces	22
Tags	22
Inceptor——指定Query ID的Query Analysis(查询分析)	22
Description	22

Parameters	22
Responses	22
Consumes	22
Produces	22
Tags	22
Inceptor——指定Query ID的Query Analysis(查询分析)	22
Description	23
Parameters	23
Responses	23
Produces	23
Tags	23
返回所有符合筛选条件的Query(查询)的持久化数据	23
Description	23
Responses	23
Consumes	24
Produces	24
Tags	24
返回所有符合筛选条件的Query(查询)的持久化数据	24
Description	24
Responses	24
Produces	24
Tags	24
Inceptor——所有Queries(查询)	24
Description	24
Parameters	24
Responses	25
Consumes	25
Produces	25
Tags	25
Inceptor——所有Queries(查询)	25
Description	25
Parameters	25
Responses	26
Produces	26
Tags	26
Inceptor——指定Stage ID的Stage(调度阶段)	26
Description	26
Parameters	26
Responses	26
Consumes	26
Produces	27
Tags	27
Inceptor——指定Stage ID的Stage(调度阶段)	27
Description	27
Parameters	27
Responses	27
Produces	27
Tags	27
Inceptor——所有Stages(调度阶段)	27
Description	27
Parameters	28
Responses	28
Consumes	28
Produces	28
Tags	28
Inceptor——所有Stages(调度阶段)	28
Description	28
Parameters	28
Responses	29
Produces	29
Tags	29
Inceptor——Inceptor的Status(状态)数据	29
Description	29

Parameters	29
Responses	29
Consumes	29
Produces	30
Tags	30
Inceptor——Inceptor的Status(状态)数据	30
Description	30
Parameters	30
Responses	30
Produces	30
Tags	30
Inceptor——指定Task ID的Task(任务)	30
Description	30
Parameters	31
Responses	31
Consumes	31
Produces	31
Tags	31
Inceptor——指定Task ID的Task(任务)	31
Description	31
Parameters	31
Responses	32
Produces	32
Tags	32
Definitions	33
AtomicInteger	33
AtomicLong	33
ColdStageInfo	33
ColdTaskInfo	33
ColumnInfo	34
DAG图边的数据字段	34
DAG图顶点的数据字段	34
Executor(执行器)的GC(垃圾清理)信息	34
Executor(执行器)的数据字段	34
Executor(执行器)的时间轴	36
Inceptor Overview(概要信息)的数据字段	36
InceptorServer的数据字段	37
InceptorSqlAnalysisResponse	37
Job	38
Job(作业)的数据字段	38
Map«string,object»	40
Query(查询)的数据字段	40
ResponseEntity	41
Session(会话)的数据字段	42
SimpleStageInfo	42
Sql	42
Stage	44
Stage(调度阶段)数据字段	45
StageAttempt	46
StageInfo	46
Status	46
Status数据字段	47
TableInfo	47
Task	48
Task(任务)数据字段	49
Value	50
持久化Executor(执行器)的Key结构	50
持久化InceptorServer数据	50
持久化InceptorServer统计数据	50
持久化Query(查询)的统计	51
持久化的Executor(执行器)	51
持久化的Query(查询) Key值结构定义	52
持久化的Query(查询)基本信息	52

Overview

DBA Service 对外提供的标准 RESTful API

Version information

Version : 1.5.0-SNAPSHOT

Contact information

Contact : Yang Ren

URI scheme

Host : localhost:8888

BasePath : /

Tags

- 1、Inceptor API : 与Inceptor各项数据相关的API
- 2、Inceptor持久化 API : 在数据库被持久化的Inceptor各项数据相关的API

Paths

持久化——指定Executor ID的Executor(执行器)

```
POST /api/inceptor/coldExecutor
```

Description

根据指定Server ID和Executor ID获取一个Executor(执行器)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Executor(执行器)	持久化的Executor(执行器)
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- 2、Inceptor持久化 API

持久化——指定Executor ID的Executor(执行器)

```
GET /api/inceptor/coldExecutor
```

Description

根据指定Server ID和Executor ID获取一个Executor(执行器)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Executor(执行器)	持久化的Executor(执行器)
400	URL路径错误或缺少必要请求参数， 请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· **/**

Tags

· 2、Inceptor持久化 API

持久化———所有InceptorServer

POST /api/inceptor/coldServers

Description

所有被持久化的InceptorServer数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>optional</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有被持久化的InceptorServer数据的Iterable	持久化InceptorServer数据
201	Created	No Content

HTTP Code	Description	Schema
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 2、Inceptor持久化 API

持久化———所有InceptorServer

```
GET /api/inceptor/coldServers
```

Description

所有被持久化的InceptorServer数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>optional</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有被持久化的InceptorServer数据的Iterable	持久化InceptorServer数据
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 2、Inceptor持久化 API

返回InceptorServer的统计数据

```
POST /api/inceptor/coldServers/statistics
```

Description

返回类型是List

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回InceptorServer的统计数据	持久化InceptorServer统计数据
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- 2、Inceptor持久化 API

返回InceptorServer的统计数据

```
GET /api/inceptor/coldServers/statistics
```

Description

返回类型是List

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回InceptorServer的统计数据	持久化InceptorServer统计数据
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

- `*/*`

Tags

· 2、Inceptor持久化 API

持久化———指定Query ID的Query(查询)

```
POST /api/inceptor/coldSql
```

Description

根据Server Id和Query Id获取指定持久化的Query(查询)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Query ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！返回一个Query(查询)	持久化的Query(查询)数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 2、Inceptor持久化 API

持久化———指定Query ID的Query(查询)

```
GET /api/inceptor/coldSql
```

Description

根据Server Id和Query Id获取指定持久化的Query(查询)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Query ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Query(查询)	持久化的Query(查询)数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· `*/*`

Tags

· 2、Inceptor持久化 API

返回Query(查询)的统计数据

```
POST /api/inceptor/coldSqls/statistics
```

Description

返回类型是List

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回InceptorServer的统计数据	持久化Query(查询)的统计
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· `application/json`

Produces

· [*/*](#)

Tags

· 2、Inceptor持久化 API

返回Query(查询)的统计数据

```
GET /api/inceptor/coldSqls/statistics
```

Description

返回类型是List

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回InceptorServer的统计数据	持久化Query(查询)的统计
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· [*/*](#)

Tags

· 2、Inceptor持久化 API

Inceptor——Inceptor的Environment(环境配置)状态

```
POST /api/inceptor/environment
```

Description

根据指定Server ID获取Inceptor的环境配置和jar包数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含指定Inceptor所有环境配置和jar包数据的Map,即<K,V>对	ResponseEntity
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· `application/json`

Produces

· `*/*`

Tags

· 1、Inceptor API

Inceptor——Inceptor的Environment(环境配置)状态

GET /api/inceptor/environment

Description

根据指定Server ID获取Inceptor的环境配置和jar包数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含指定Inceptor所有环境配置和jar包数据的Map,即<K,V>对	ResponseEntity
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Executor ID的Executor(执行器)

```
POST /api/inceptor/executor
```

Description

根据InceptorServer的Server ID和Executor ID获取指定的一个Executor(执行器)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Executor(执行器)	Executor(执行器)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Executor ID的Executor(执行器)

```
GET /api/inceptor/executor
```

Description

根据InceptorServer的Server ID和Executor ID获取指定的一个Executor(执行器)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Executor(执行器)	Executor(执行器)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· [*/*](#)

Tags

· 1、Inceptor API

Inceptor———Executors(执行器)

POST /api/inceptor/executors

Description

根据指定Server ID获取Inceptor上Executor(执行器)数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行! 返回一个包含所有Executor(执行器)数据的List	Executor(执行器)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数, 请检查URL格式!	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· `application/json`

Produces

· `*/*`

Tags

· 1、Inceptor API

Inceptor———Executors(执行器)

GET /api/inceptor/executors

Description

根据指定Server ID获取Inceptor上Executor(执行器)数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID (包含uuid)	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行! 返回一个包含所有Executor(执行器)数据的List	Executor(执行器)的数据字段
400	URL路径错误或缺少必要请求参数, 请检查URL格式!	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Job ID的Job(作业)详细信息

POST /api/inceptor/job

Description

根据指定Server ID和Job ID获取该Job(作业)的详细信息

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Job ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Job(作业)	Job(作业)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Job ID的Job(作业)详细信息

GET /api/inceptor/job

Description

根据指定Server ID和Job ID获取该Job(作业)的详细信息

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Job ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Job(作业)	Job(作业)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· **/**

Tags

· 1、Inceptor API

Inceptor————所有Jobs(作业)

POST /api/inceptor/jobs

Description

根据指定Server ID获取Inceptor上的所有Job(作业)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Jobs(作业)数据的List	Job(作业)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- 1、Inceptor API

Inceptor————所有Jobs(作业)

```
GET /api/inceptor/jobs
```

Description

根据指定Server ID获取Inceptor上的所有Job(作业)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Jobs(作业)数据的List	Job(作业)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———Overview数据

POST /api/inceptor/overview

Description

获取Inceptor的概要信息，了解Inceptor的运行状况

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！返回一个Overview	Inceptor Overview (概要信息)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———Overview数据

GET /api/inceptor/overview

Description

获取Inceptor的概要信息，了解Inceptor的运行状况

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Overview	Inceptor Overview (概要信息)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· **/**

Tags

· 1、Inceptor API

Inceptor——InceptorServer所有数据

POST /api/inceptor/servers

Description

获取所有(包含持久化)InceptorServer数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>optional</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行! 返回一个包含所有(含持久化的)Servers数据的List	InceptorServer的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数, 请检查URL格式!	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· `application/json`

Produces

· `*/*`

Tags

· 1、Inceptor API

Inceptor——InceptorServer所有数据

GET /api/inceptor/servers

Description

获取所有(包含持久化)InceptorServer数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>optional</i>	InceptorServer的server ID (包含uuid)	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行! 返回一个包含所有(含持久化的)Servers数据的List	InceptorServer的数据字段
400	URL路径错误或缺少必要请求参数, 请检查URL格式!	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———所有Sessions(会话)

POST /api/inceptor/sessions

Description

根据Server ID和dataSize大小获取Inceptor指定数量的Session(会话)数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Sessions(会话)数据的List	Session(会话)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———所有Sessions(会话)

GET /api/inceptor/sessions

Description

根据Server ID和dataSize大小获取Inceptor指定数量的Session(会话)数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Sessions(会话)数据的List	Session(会话)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Query ID的Query(查询)

POST /api/inceptor/sql

Description

根据指定Server ID和Query ID获取该Query(查询)的详细信息

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Query ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Query(查询)	Query(查询)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Query ID的Query(查询)

```
GET /api/inceptor/sql
```

Description

根据指定Server ID和Query ID获取该Query(查询)的详细信息

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Query ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Query(查询)	Query(查询)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Query ID的Query Analysis(查询分析)

POST /api/inceptor/sqlAnalysis

Description

根据Server ID和Query ID获取一个指定Query(查询)的Analysis(分析)的详细数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Query ID	integer (int64)
Query	span <i>optional</i>		integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Query Analysis(查询分析)	InceptorSqlAnalysisResponse
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Query ID的Query Analysis(查询分析)

GET /api/inceptor/sqlAnalysis

Description

根据Server ID和Query ID获取一个指定Query(查询)的Analysis(分析)的详细数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID (包含uuid)	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Query ID	integer (int64)
Query	span <i>optional</i>		integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行! 返回一个Query Analysis(查询分析)	InceptorSqlAnalysisResponse
400	URL路径错误或缺少必要请求参数, 请检查URL格式!	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

返回所有符合筛选条件的Query(查询)的持久化数据

POST /api/inceptor/sqlSearch

Description

参数是筛选条件, 返回类型是List

Responses

HTTP Code	Description	Schema
200	成功运行! 返回所有持久化的InceptorServer数据	持久化的Query(查询)基本信息
201	Created	No Content
400	URL路径错误或缺少必要请求参数, 请检查URL格式!	No Content

HTTP Code	Description	Schema
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 2、Inceptor持久化 API

返回所有符合筛选条件的Query(查询)的持久化数据

GET /api/inceptor/sqlSearch

Description

参数是筛选条件，返回类型是List

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回所有持久化的InceptorServer数据	持久化的Query(查询)基本信息
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 2、Inceptor持久化 API

Inceptor———所有Queries(查询)

POST /api/inceptor/sqls

Description

根据指定Server ID获取Inceptor所有Queries(查询)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Queries(查询)的List	Query(查询)的数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数， 请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- `application/json`

Produces

- `*/*`

Tags

- 1、Inceptor API

Inceptor———所有Queries(查询)

GET /api/inceptor/sqls

Description

根据指定Server ID获取Inceptor所有Queries(查询)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Queries(查询)的List	Query(查询)的数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· `*/*`

Tags

· 1、Inceptor API

Inceptor——指定Stage ID的Stage(调度阶段)

```
POST /api/inceptor/stage
```

Description

根据Server ID和Stage ID获取一个指定Stage(调度阶段)的详细数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Job ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Stage(调度阶段)	Stage(调度阶段)数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· `application/json`

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———指定Stage ID的Stage(调度阶段)

GET /api/inceptor/stage

Description

根据Server ID和Stage ID获取一个指定Stage(调度阶段)的详细数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Job ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Stage(调度阶段)	Stage(调度阶段)数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———所有Stages(调度阶段)

POST /api/inceptor/stages

Description

根据Server ID获取Inceptor上所有Stage(调度阶段)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Stages(调度阶段)的List	Stage(调度阶段)数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor———所有Stages(调度阶段)

GET /api/inceptor/stages

Description

根据Server ID获取Inceptor上所有Stage(调度阶段)的数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个包含所有Stages(调度阶段)的List	Stage(调度阶段)数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· `*/*`

Tags

· 1、Inceptor API

Inceptor——Inceptor的Status(状态)数据

POST /api/inceptor/status

Description

根据指定Server ID获取Inceptor的状态数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回指定Server ID的Inceptor数据	Status数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· `application/json`

Produces

· */*

Tags

· 1、Inceptor API

Inceptor——Inceptor的Status(状态)数据

GET /api/inceptor/status

Description

根据指定Server ID获取Inceptor的状态数据

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID（包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>optional</i>	id	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！返回指定Server ID的Inceptor数据	Status数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· */*

Tags

· 1、Inceptor API

Inceptor——指定Task ID的Task(任务)

POST /api/inceptor/task

Description

根据Server ID和Task ID获取一个指定Task(任务)的详细信息

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Task ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Task(任务)	Task(任务)数据字段
201	Created	No Content
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

· application/json

Produces

· */*

Tags

· 1、Inceptor API

Inceptor——指定Task ID的Task(任务)

```
GET /api/inceptor/task
```

Description

根据Server ID和Task ID获取一个指定Task(任务)的详细信息

Parameters

Type	Name	Description	Schema
Query	dataKey <i>required</i>	InceptorServer的server ID （包含uuid）	string
Query	dataSize <i>optional</i>	想要得到的数据条数	integer (int32)
Query	id <i>required</i>	请求的Task ID	integer (int64)
Query	stringId <i>optional</i>	stringId	string

Responses

HTTP Code	Description	Schema
200	成功运行！ 返回一个Task(任务)	Task(任务)数据字段
400	URL路径错误或缺少必要请求参数，请检查URL格式！	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Produces

· [*/*](#)

Tags

· 1、Inceptor API

Definitions

AtomicInteger

Name	Schema
andDecrement <i>optional</i>	integer (int32)
andIncrement <i>optional</i>	integer (int32)

AtomicLong

Name	Schema
andDecrement <i>optional</i>	integer (int64)
andIncrement <i>optional</i>	integer (int64)

ColdStageInfo

Name	Schema
et <i>optional</i>	integer (int64)
id <i>optional</i>	integer (int64)
maxtt <i>optional</i>	integer (int64)
mintt <i>optional</i>	integer (int64)
st <i>optional</i>	integer (int64)
tts <i>optional</i>	number (double)

ColdTaskInfo

Name	Schema
d <i>optional</i>	integer (int64)
et <i>optional</i>	integer (int64)
id <i>optional</i>	integer (int64)
ir <i>optional</i>	integer (int64)
m <i>optional</i>	integer (int64)
sr <i>optional</i>	integer (int64)
st <i>optional</i>	integer (int64)
sw <i>optional</i>	integer (int64)

ColumnInfo

Name	Schema
index <i>optional</i>	integer (int32)
name <i>optional</i>	string
type <i>optional</i>	string

DAG图边的数据字段

Name	Description	Schema
f <i>optional</i>	头节点 Example: "Stage 1"	string
t <i>optional</i>	尾节点 Example: "Stage 2"	string

DAG图顶点的数据字段

Name	Description	Schema
desc <i>optional</i>	顶点简要描述 Example: "Stage 1 Detail"	string
detail <i>optional</i>	顶点详细信息 Example: "Stage 1"	string
group <i>optional</i>	顶点所属组 Example: "Stage 1"	string
id <i>optional</i>	顶点Id Example: "Stage 1"	string

Executor(执行器)的GC(垃圾清理)信息

Name	Description	Schema
action <i>optional</i>	操作	string
cause <i>optional</i>	起因	string
duration <i>optional</i>	持续时间	integer (int64)
memoryAfterGC <i>optional</i>	GC后的内存大小	integer (int64)
memoryBeforeGC <i>optional</i>	GC前的内存大小	integer (int64)
name <i>optional</i>	名字	string
rawString <i>optional</i>	列数据	string
timestamp <i>optional</i>	时间戳	integer (int64)

Executor(执行器)的数据字段

Name	Description	Schema
activeTasks <i>optional</i>	活跃中的TaskTask(任务)数	AtomicLong
completedTasks <i>optional</i>	已完成的Task(任务)数	AtomicLong
executorId <i>optional</i>	Executor ID: Executor(执行器)的唯一标识符 Example : "32"	string
failedTasks <i>optional</i>	执行失败的Task(任务)数	AtomicLong
firstSeen <i>optional</i>	DBA Service第一次收到Executor(执行器)的心跳时间戳 Example : 1576494871454	integer (int64)
firstSeenLocal <i>optional</i>	DBA Service第一次收到Executor(执行器)心跳时的本地时间戳 Example : 1576494871454	integer (int64)
gcInfo <i>optional</i>	Executor(执行器)的gcInfo(垃圾回收)	< Executor(执行器)的GC(垃圾清理)信息 > array
host <i>optional</i>	InceptorServer的主机名 Example : "baisheng2"	string
input <i>optional</i>	输入	AtomicLong
jobs <i>optional</i>	Executor(执行器)上的Job(作业)	< Job(作业)的数据字段 > array
lastSeen <i>optional</i>	DBA Service最近一次收到Executor(执行器)的心跳时间戳 Example : 1576494871454	integer (int64)
lastSeenLocal <i>optional</i>	DBA Service最近一次收到Executor(执行器)心跳时的本地时间戳 Example : 1576494871454	integer (int64)
localTasks <i>optional</i>	Executor(执行器)上的本地Task(任务)	< Task(任务)数据字段 > array
maxMemory <i>optional</i>	Executor(执行器)最大内存 Example : 2	integer (int64)
port <i>optional</i>	InceptorServer的端口号 Example : 10000	integer (int32)
shuffleReadBytes <i>optional</i>	Shuffle读数据的总大小	AtomicLong
shuffleSpillDisk <i>optional</i>	Shuffle磁盘溢出	AtomicLong
shuffleSpillMemory <i>optional</i>	Shuffle内存溢出	AtomicLong
shuffleWriteBytes <i>optional</i>	Shuffle写数据的总大小	AtomicLong
sqls <i>optional</i>	Executor(执行器)上的Query(查询)	< Query(查询)的数据字段 > array
stages <i>optional</i>	Executor(执行器)上的Stage(调度阶段)	< Stage(调度阶段)数据字段 > array
statuses <i>optional</i>	Executor(执行器)的Statues(状态)	< Status数据字段 > array
tasks <i>optional</i>	Executor(执行器)上的Task(任务)	< Task(任务)数据字段 > array
tasksActive <i>optional</i>	Executor(执行器)上执行中的Task(任务)数 Example : 2	integer (int64)
tasksCompleted <i>optional</i>	Executor(执行器)已完成的Task(任务)数 Example : 2	integer (int64)
tasksFailed <i>optional</i>	Executor(执行器)执行失败的Task(任务)数 Example : 2	integer (int64)

Name	Description	Schema
totalCores <i>optional</i>	Executor(执行器)核心数 Example : 2	integer (int32)
totalTask <i>optional</i>	Task(任务)总数	AtomicLong
totalTaskTime <i>optional</i>	所有Task(任务)执行的时间总和	AtomicLong
version <i>optional</i>	版本 Example : 1	integer (int32)

Executor(执行器)的时间轴

Name	Description	Schema
s <i>optional</i>	消息的时间戳timestamp Example : 1576752272990	integer (int64)
t <i>optional</i>	消息的类型 (Register,Added,Running) Example : 0	integer (int32)

Inceptor Overview(概要信息)的数据字段

Name	Description	Schema
executors <i>optional</i>	该Inceptor的执行器数量 Example : 4	integer (int64)
firstSeen <i>optional</i>	DBA Service第一次收到该InceptorServer心跳时间戳 Example : 1576494871454	integer (int64)
jobsActive <i>optional</i>	该Inceptor正在执行的作业总数 Example : 42	integer (int64)
jobsTotal <i>optional</i>	该Inceptor上作业的总数 Example : 1153	integer (int64)
lastSeen <i>optional</i>	DBA Service最近一次收到该InceptorServer心跳时间戳 Example : 1576497271460	integer (int64)
lastSeenWatchman Timestamp <i>optional</i>	DBA Service最近一次收到该InceptorServer心跳时的本地时间戳 Example : 1576497271527	integer (int64)
schedulingMode <i>optional</i>	调度模式 Example : "FIFO"	string
sqlsActive <i>optional</i>	该Inceptor正在执行的查询总数 Example : 11	integer (int64)
sqlsTotal <i>optional</i>	该Inceptor上查询的总数 Example : 328	integer (int64)
stagesActive <i>optional</i>	该Inceptor正在执行的调度阶段的总数 Example : 18	integer (int64)
stagesTotal <i>optional</i>	该Inceptor上调度阶段的总数 Example : 2383	integer (int64)
tasksActive <i>optional</i>	该Inceptor正在执行的任务总数 Example : 10	integer (int64)
tasksTotal <i>optional</i>	该Inceptor上任务的总数 Example : 3504	integer (int64)
watchmanTimestamp <i>optional</i>	DBA Service当前本地时间戳 Example : 1576497283705	integer (int64)

InceptorServer的数据字段

Name	Description	Schema
firstSeen <i>optional</i>	DBA Service第一次收到该InceptorServer的心跳时间戳 Example : 1577945712848	integer (int64)
host <i>optional</i>	InceptorServer的主机名 Example : "baisheng2"	string
hotData <i>optional</i>	判别该InceptorServer是否还在内存中 Example : true	boolean
key <i>optional</i>	Target(监控目标)的server Id全称 Example : "Inceptor::baisheng2::fb8ae28a-b9ec-480a-9a65-2ed392513e75"	string
localTimestamp <i>optional</i>	DBA Service收到最新一次server心跳时的本地时间戳 Example : 1577945713082	integer (int64)
port <i>optional</i>	InceptorServer的端口号 Example : 10000	integer (int32)
serverId <i>optional</i>	server ID, Target(监控目标)的编号标识 Example : 32	integer (int64)
timestamp <i>optional</i>	InceptorServer最新一次心跳的时间戳 Example : 1577945712848	integer (int64)

InceptorSqlAnalysisResponse

Name	Schema
analysisData <i>optional</i>	object
completionTime <i>optional</i>	integer (int64)
description <i>optional</i>	string
detailedDescription <i>optional</i>	string
durations <i>optional</i>	< string, integer (int64) > map
executionFinishTime <i>optional</i>	integer (int64)
executionStartTime <i>optional</i>	integer (int64)
group <i>optional</i>	string
jobs <i>optional</i>	integer (int64)
message <i>optional</i>	string
metrics <i>optional</i>	< string, string > map
sessionId <i>optional</i>	integer (int64)
sqlId <i>optional</i>	integer (int64)
sqls <i>optional</i>	integer (int64)
stages <i>optional</i>	integer (int64)

Name	Schema
state <i>optional</i>	string
submissionTime <i>optional</i>	integer (int64)
tasks <i>optional</i>	integer (int64)
user <i>optional</i>	string

Job

Name	Schema
completionTime <i>optional</i>	integer (int64)
description <i>optional</i>	string
edges <i>optional</i>	< DAG图边的数据字段 > array
jobGroup <i>optional</i>	string
jobId <i>optional</i>	integer (int64)
latestTime <i>optional</i>	integer (int64)
message <i>optional</i>	string
nodes <i>optional</i>	< DAG图顶点的数据字段 > array
poolName <i>optional</i>	string
sqlId <i>optional</i>	integer (int64)
stageInputOutput <i>optional</i>	< string, string > map
stageSources <i>optional</i>	< string, string > map
stagesCompleted <i>optional</i>	integer (int32)
stagesExpectedList <i>optional</i>	< StageInfo > array
stagesFailed <i>optional</i>	integer (int32)
stagesSkipped <i>optional</i>	integer (int32)
submissionTime <i>optional</i>	integer (int64)
updateTime <i>optional</i>	integer (int64)
version <i>optional</i>	integer (int32)

Job(作业)的数据字段

Name	Description	Schema
completionTime <i>optional</i>	Job（作业）的完成时间 Example : 1576752272990	integer (int64)
description <i>optional</i>	Job（作业）的描述信息 Example : "SQL Job 11 runJob at PhysicalOpCruxOpMaker.scala:376"	string
edges <i>optional</i>	DAG图的边集合	< DAG图边的数据字段 > array
expectedStages <i>optional</i>	简单Stage(调度阶段)信息	< SimpleStageInfo > array
group <i>optional</i>	Job（作业）所属的分组 Example : "35fb7"	string
jobGroup <i>optional</i>	该Job（作业）的组 Example : "JobGroup1"	string
jobId <i>optional</i>	Job ID: Job（作业）的唯一标识符 Example : 32	integer (int64)
jobPoolName <i>optional</i>	该Job（作业）的资源池名称 Example : "poolName"	string
jobTotalTaskTime <i>optional</i>	该Job（作业）中所有执行完毕的Task（任务）的Duration（持续时长）总和 Example : "3.9 m"	integer (int64)
message <i>optional</i>	Job（作业）执行中的异常信息 Example : "io.transwarp.nucleon.JobCanceledException: Job 11 cancelled"	string
nodes <i>optional</i>	DAG图的节点集合	< DAG图顶点的数据字段 > array
sqlId <i>optional</i>	发起该Job（作业）的Query（查询） Example : 32	integer (int64)
stages <i>optional</i>		< Stage(调度阶段)数据字段 > array
stagesActive <i>optional</i>	该Job（作业）划分的正在执行的Stage（调度阶段）总数 Example : 1	integer (int64)
stagesCompleted <i>optional</i>	该Job（作业）划分的已经执行完成的Stage（调度阶段）总数 Example : 3	integer (int64)
stagesExpected <i>optional</i>	stagesExpected Example : 32	integer (int64)
stagesFailed <i>optional</i>	该Job（作业）划分的执行失败的Stage（调度阶段）总数 Example : 1	integer (int64)
stagesSkipped <i>optional</i>	该Job（作业）划分的被跳过的Stage（调度阶段）总数 Example : 1	integer (int64)
submissionTime <i>optional</i>	Job(作业)的提交时间 Example : 1576752254435	integer (int64)
tasksActive <i>optional</i>	该Job（作业）发起的正在执行中的Task（任务）总数 Example : 20	integer (int64)
tasksCompleted <i>optional</i>	该Job（作业）发起的已经执行完成的Task（任务）总数 Example : 60	integer (int64)
tasksExpected <i>optional</i>	tasksExpected Example : "tasksExpected"	integer (int64)
tasksFailed <i>optional</i>	该Job（作业）发起的执行失败的Task（任务）总数 Example : 10	integer (int64)
tasksSkipped <i>optional</i>	该Job（作业）发起的Task（任务）中被跳过的总数 Example : 10	integer (int64)
userName <i>optional</i>	发起该Job（作业）的用户名 Example : "user1"	string

Map«string,object»

Type : < string, object > map

Query(查询)的数据字段

Name	Description	Schema
activeTasks <i>optional</i>	该查询Query发起的任务信息集合 Example : 10	integer (int64)
completionTime <i>optional</i>	查询Query完成的时间 Example : 1576752272990	integer (int64)
description <i>optional</i>	该查询Query的描述信息 Example : "from flatten bank_with_more_props.__graph_rel output"	string
detailedDescription <i>optional</i>	该查询Query的详细描述信息 Example : "from flatten bank_with_more_props.__graph_rel output"	string
durationBlockMap <i>optional</i>	Query各个模块的持续时间(K/V)	< string, string > map
durations <i>optional</i>	持续执行时间细化 (K/V)	< string, integer (int64) > map
executionFinishTime <i>optional</i>	查询Query结束执行的时间 Example : 1576752272990	integer (int64)
executionStartTime <i>optional</i>	查询Query开始执行的时间 Example : 1576752254435	integer (int64)
extraJobId <i>optional</i>	extraJobId Example : 1	integer (int64)
group <i>optional</i>	查询Query所属的组 Example : "Inceptor"	string
jobs <i>optional</i>	该查询Query发起的作业信息集合	< Job(作业)的数据字段 > array
jobsActive <i>optional</i>	该查询Query下正在活跃的作业 Example : 2	integer (int64)
jobsCompleted <i>optional</i>	该查询Query下已经完成的作业 Example : 10	integer (int64)
jobsFailed <i>optional</i>	该查询Query下执行失败的作业 Example : 1	integer (int64)
logicalPlanString <i>optional</i>	逻辑执行计划	string
logicalPlanStrings <i>optional</i>	逻辑执行计划字符串集合	< string > array
memoryPerTask <i>optional</i>	每个任务占用内存大小 Example : 5	integer (int64)
message <i>optional</i>	执行该查询Query出现的异常信息 Example : "Query is either failed, killed, or something unexpected happened.\nState: ERROR\nOperation Exception: org.apache.hive.service.cli.HiveSQLException: COMPILE FAILED: Parse error: [Error 1101] line 1:33 cannot recognize input near '.__graph_rel output' in from source"	string
messageNameStr <i>optional</i>		string
meta <i>optional</i>	meta	< string, string > map
metrics <i>optional</i>	Query各项指标 (K/V)	< string, string > map
mode <i>optional</i>	执行模式 Example : "cluster"	string

Name	Description	Schema
physicalPlanString <i>optional</i>	物理执行计划	string
physicalPlanStrings <i>optional</i>	物理执行计划字符串集合	< string > array
planInputs <i>optional</i>	计划输入	< string, integer (int64) > map
planOutputs <i>optional</i>	计划输出	< string, < string, object > map > map
sessionId <i>optional</i>	该查询Query所属的用户会话Session ID Example : 1	integer (int64)
sqlBlockJson <i>optional</i>		string
sqlId <i>optional</i>	Query ID, 查询的唯一标识符 Example : 32	integer (int64)
stages <i>optional</i>	该查询Query发起的调度阶段信息集合	< Stage(调度阶段)数据字段 > array
stagesActive <i>optional</i>	该查询Query下正在活跃的调度阶段 Example : 2	integer (int64)
stagesCompleted <i>optional</i>	该查询Query下已经完成的调度阶段 Example : 10	integer (int64)
stagesFailed <i>optional</i>	该查询Query下执行失败的调度阶段 Example : 1	integer (int64)
stagesSkipped <i>optional</i>	该查询Query下被跳过的调度阶段 Example : 3	integer (int64)
state <i>optional</i>	状态 Example : "SUCCESS/FAILED/RUNNING"	string
submissionTime <i>optional</i>	查询Query提交时间 Example : 1576752254435	integer (int64)
tableScans <i>optional</i>	扫描表	object
totalTaskTime <i>optional</i>	该查询Query的所有任务执行时间总和 Example : 1000	integer (int64)
user <i>optional</i>	发起该查询Query的用户名 Example : "anonymous"	string

ResponseEntity

Name	Schema
body <i>optional</i>	object

Name	Schema
statusCode <i>optional</i>	enum (100 CONTINUE, 101 SWITCHING_PROTOCOLS, 102 PROCESSING, 103 CHECKPOINT, 200 OK, 201 CREATED, 202 ACCEPTED, 203 NON_AUTHORITATIVE_INFORMATION, 204 NO_CONTENT, 205 RESET_CONTENT, 206 PARTIAL_CONTENT, 207 MULTI_STATUS, 208 ALREADY_REPORTED, 226 IM_USED, 300 MULTIPLE_CHOICES, 301 MOVED_PERMANENTLY, 302 FOUND, 302 MOVED_TEMPORARILY, 303 SEE_OTHER, 304 NOT_MODIFIED, 305 USE_PROXY, 307 TEMPORARY_REDIRECT, 308 PERMANENT_REDIRECT, 400 BAD_REQUEST, 401 UNAUTHORIZED, 402 PAYMENT_REQUIRED, 403 FORBIDDEN, 404 NOT_FOUND, 405 METHOD_NOT_ALLOWED, 406 NOT_ACCEPTABLE, 407 PROXY_AUTHENTICATION_REQUIRED, 408 REQUEST_TIMEOUT, 409 CONFLICT, 410 GONE, 411 LENGTH_REQUIRED, 412 PRECONDITION_FAILED, 413 PAYLOAD_TOO_LARGE, 413 REQUEST_ENTITY_TOO_LARGE, 414 URI_TOO_LONG, 414 REQUEST_URI_TOO_LONG, 415 UNSUPPORTED_MEDIA_TYPE, 416 REQUESTED_RANGE_NOT_SATISFIABLE, 417 EXPECTATION_FAILED, 418 I_AM_A_TEAPOT, 419 INSUFFICIENT_SPACE_ON_RESOURCE, 420 METHOD_FAILURE, 421 DESTINATION_LOCKED, 422 UNPROCESSABLE_ENTITY, 423 LOCKED, 424 FAILED_DEPENDENCY, 426 UPGRADE_REQUIRED, 428 PRECONDITION_REQUIRED, 429 TOO_MANY_REQUESTS, 431 REQUEST_HEADER_FIELDS_TOO_LARGE, 451 UNAVAILABLE_FOR_LEGAL_REASONS, 500 INTERNAL_SERVER_ERROR, 501 NOT_IMPLEMENTED, 502 BAD_GATEWAY, 503 SERVICE_UNAVAILABLE, 504 GATEWAY_TIMEOUT, 505 HTTP_VERSION_NOT_SUPPORTED, 506 VARIANT_ALSO_NEGOTIATES, 507 INSUFFICIENT_STORAGE, 508 LOOP_DETECTED, 509 BANDWIDTH_LIMIT_EXCEEDED, 510 NOT_EXTENDED, 511 NETWORK_AUTHENTICATION_REQUIRED)
statusCodeValue <i>optional</i>	integer (int32)

Session(会话)的数据字段

Name	Description	Schema
closeDatetime <i>optional</i>	会话关闭的日期时间 Example : "2019/12/19 15:58:18"	integer (int64)
ipAddress <i>optional</i>	IP地址 Example : "172.26.0.24"	string
message <i>optional</i>	正常、警告或报错信息 Example : "Closed by user."	string
openDatetime <i>optional</i>	会话打开的日期时间 Example : "2019/12/19 12:10:03"	integer (int64)
sessionId <i>optional</i>	用户会话编号 Example : 1	integer (int64)
sessionUUID <i>optional</i>	用户会话的唯一标识符 Example : "250aa87e-2bc9-4a8e-97a9-f469a9b976d2"	string
user <i>optional</i>	用户名 Example : "anonymous"	string

SimpleStageInfo

Type : object

Sql

Name	Schema
accurateState <i>optional</i>	string
background <i>optional</i>	boolean
completionTime <i>optional</i>	integer (int64)

Name	Schema
computeFinishTime <i>optional</i>	integer (int64)
description <i>optional</i>	string
detailedDescription <i>optional</i>	string
durationBlockMap <i>optional</i>	< string, string > map
durations <i>optional</i>	< string, integer (int64) > map
extraJobId <i>optional</i>	integer (int64)
latestTime <i>optional</i>	integer (int64)
localTime <i>optional</i>	integer (int64)
logicalPlanString <i>optional</i>	string
logicalPlanStrings <i>optional</i>	< string > array
message <i>optional</i>	string
meta <i>optional</i>	< string, string > map
metrics <i>optional</i>	< string, string > map
mode <i>optional</i>	string
physicalPlanString <i>optional</i>	string
physicalPlanStrings <i>optional</i>	< string > array
planInputs <i>optional</i>	< string, integer (int64) > map
planOutputs <i>optional</i>	< string, < string, object > map > map
sessionId <i>optional</i>	integer (int64)
sqlGroup <i>optional</i>	string
sqlId <i>optional</i>	integer (int64)
startTime <i>optional</i>	integer (int64)
state <i>optional</i>	Value
submissionTime <i>optional</i>	integer (int64)
tableScans <i>optional</i>	< string, TableInfo > map
version <i>optional</i>	integer (int32)

Stage

Name	Schema
actualTasksCompleted <i>optional</i>	AtomicInteger
actualTasksFailed <i>optional</i>	AtomicInteger
attemptId <i>optional</i>	integer (int32)
attempts <i>optional</i>	< string, StageAttempt > map
completionTime <i>optional</i>	integer (int64)
description <i>optional</i>	string
input <i>optional</i>	AtomicLong
inputRowsString <i>optional</i>	string
jobId <i>optional</i>	integer (int64)
latestTime <i>optional</i>	integer (int64)
message <i>optional</i>	string
outputRowsString <i>optional</i>	string
partitionSize <i>optional</i>	integer (int32)
poolName <i>optional</i>	string
shuffleReadBytes <i>optional</i>	integer (int64)
shuffleReadBytesAggregated <i>optional</i>	AtomicLong
shuffleWriteBytes <i>optional</i>	integer (int64)
shuffleWriteBytesAggregated <i>optional</i>	AtomicLong
stageId <i>optional</i>	integer (int64)
stageMode <i>optional</i>	string
status <i>optional</i>	enum (UNKNOWN, SUBMITTED, PRERUN, RUNNING, POSTRUN, COMPLETED, FAILED, SKIPPED)
submissionTime <i>optional</i>	integer (int64)
tasksCompleted <i>optional</i>	integer (int32)
tasksExpected <i>optional</i>	integer (int32)
tasksFailed <i>optional</i>	integer (int32)
tasksSkipped <i>optional</i>	integer (int32)

Name	Schema
totalTaskTime <i>optional</i>	AtomicLong
version <i>optional</i>	integer (int32)

Stage(调度阶段)数据字段

Name	Description	Schema
attemptId <i>optional</i>	Stage(调度阶段)所属的Attempt(意图) ID Example : 32	integer (int32)
attempts <i>optional</i>	Stage(调度阶段)的简要	< StageAttempt > array
completionTime <i>optional</i>	Stage(调度阶段)的完成时间 Example : 32	integer (int64)
description <i>optional</i>	Stage(调度阶段)的描述信息	string
executors <i>optional</i>	Stage(调度阶段)的Executors(执行器)集合 Example : 32	< Executor(执行器) 的数据字段 > array
input <i>optional</i>	Stage(调度阶段)的输入 Example : 32	integer (int64)
inputRowsString <i>optional</i>	Stage(调度阶段)的输入列字符	string
jobId <i>optional</i>	Stage(调度阶段)所属的Job(作业) ID Example : 32	integer (int64)
message <i>optional</i>	Stage(调度阶段)执行过程中的错误异常信息	string
outputRowsString <i>optional</i>	Stage(调度阶段)的输出列字符串	string
partitionSize <i>optional</i>	Stage(调度阶段)分区大小 Example : 32	integer (int32)
shuffleReadBytes <i>optional</i>	Stage(调度阶段)的shuffle读大小 Example : 32	integer (int64)
shuffleWriteBytes <i>optional</i>	Stage(调度阶段)的shuffle写大小 Example : 32	integer (int64)
sources <i>optional</i>	Stage(调度阶段)源	string
sqlGroup <i>optional</i>	Stage(调度阶段)所属的QueryGroup(查询所属组) Example : "Inceptor"	string
sqlId <i>optional</i>	Stage(调度阶段)所属的Query(查询) ID Example : 32	integer (int64)
stageId <i>optional</i>	Stage ID: Stage(调度阶段)的唯一标识符 Example : 32	integer (int64)
stageMode <i>optional</i>	Stage(调度阶段)调度模式 Example : "32"	string
stagePoolName <i>optional</i>	Stage(调度阶段)的Pool(资源池)名称 Example : "32"	string
status <i>optional</i>	Stage(调度阶段)的状态数据	string
submissionTime <i>optional</i>	Stage(调度阶段)的提交时间 Example : 32	integer (int64)
taskTimes <i>optional</i>	Stage(调度阶段)的Task(任务)时长列表	< integer (int64) > array

Name	Description	Schema
tasks <i>optional</i>	Stage(调度阶段)下的Task(任务)列表	< Task(任务)数据字段 > array
tasksActive <i>optional</i>	Stage(调度阶段)所属执行中的Task(任务) Example : 3	integer (int64)
tasksCompleted <i>optional</i>	Stage(调度阶段)所属已完成的Task(任务) Example : 2	integer (int64)
tasksExpected <i>optional</i>	Stage(调度阶段)被期望的Task(任务) Example : 2	integer (int64)
tasksFailed <i>optional</i>	Stage(调度阶段)所属执行失败的Task(任务) Example : 3	integer (int64)
tasksSkipped <i>optional</i>	Stage(调度阶段)所属被跳过的Task(任务) Example : 2	integer (int64)
totalTaskTime <i>optional</i>	Stage(调度阶段)下的所有任务持续时间总和 Example : 32	integer (int64)
userName <i>optional</i>	Stage(调度阶段)所属用户名 Example : "32"	string

StageAttempt

Name	Schema
attemptId <i>optional</i>	integer (int32)
completionTime <i>optional</i>	integer (int64)
message <i>optional</i>	string
stageId <i>optional</i>	integer (int64)
submissionTime <i>optional</i>	integer (int64)

StageInfo

Name	Schema
attemptId <i>optional</i>	integer (int64)
jobId <i>optional</i>	integer (int64)
numTasks <i>optional</i>	integer (int32)
sources <i>optional</i>	string
stageId <i>optional</i>	integer (int64)

Status

Name	Schema
curTotalHeapMem <i>optional</i>	integer (int64)
curUsedHeapMem <i>optional</i>	integer (int64)

Name	Schema
historyGCCount <i>optional</i>	integer (int64)
historyGCTimeTotalMS <i>optional</i>	integer (int64)
openedFDCount <i>optional</i>	integer (int32)
recentSysTimeTicks <i>optional</i>	integer (int64)
recentUserTimeTicks <i>optional</i>	integer (int64)
threadCount <i>optional</i>	integer (int32)
timeElapsed <i>optional</i>	integer (int64)
timestamp <i>optional</i>	integer (int64)

Status数据字段

Name	Description	Schema
curTotalHeapMemMB <i>optional</i>	堆内存总大小 Example : 10000	integer (int64)
curUsedHeapMemMB <i>optional</i>	已使用的堆内存大小 Example : 1000	integer (int64)
historyGCCount <i>optional</i>	历史GC(垃圾回收)次数 Example : 10	integer (int64)
historyGCTimeTotalMS <i>optional</i>	历史GC(垃圾回收)总耗时 Example : 1000	integer (int64)
openedFDCount <i>optional</i>	打开的文件描述符 Example : 32	integer (int32)
recentSysTimeTicks <i>optional</i>	最近系统时间标记 Example : 32	integer (int64)
recentUserTimeTicks <i>optional</i>	最近用户时间标记 Example : 32	integer (int64)
threadCount <i>optional</i>	线程数 Example : 32	integer (int32)
timeElapsed <i>optional</i>	已生存时间 Example : 1000	integer (int64)
timestamp <i>optional</i>	时间戳 Example : 1576494871454	integer (int64)

TableInfo

Name	Schema
alias <i>optional</i>	string
columnsNeeded <i>optional</i>	< ColumnInfo > array
database <i>optional</i>	string

Name	Schema
name <i>optional</i>	string

Task

Name	Schema
accumulators <i>optional</i>	< string, string > map
aggregation <i>optional</i>	string
attempt <i>optional</i>	integer (int32)
durations <i>optional</i>	< string, integer (int64) > map
executorId <i>optional</i>	string
finishTime <i>optional</i>	integer (int64)
garbageCollectionTime <i>optional</i>	integer (int64)
input <i>optional</i>	integer (int64)
latestTime <i>optional</i>	integer (int64)
launchTime <i>optional</i>	integer (int64)
message <i>optional</i>	string
metrics <i>optional</i>	< string, string > map
resultSerializationTime <i>optional</i>	integer (int64)
schedulerDelay <i>optional</i>	integer (int64)
sequence <i>optional</i>	string
shuffleReadBytes <i>optional</i>	integer (int64)
shuffleSpillDisk <i>optional</i>	integer (int64)
shuffleSpillMemory <i>optional</i>	integer (int64)
shuffleWriteBytes <i>optional</i>	integer (int64)
stageId <i>optional</i>	integer (int64)
taskId <i>optional</i>	integer (int64)
taskIndexInStage <i>optional</i>	integer (int32)
taskLocality <i>optional</i>	Value
taskLocalityAsString <i>optional</i>	string

Name	Schema
timeSpentFetchingTaskResults <i>optional</i>	integer (int64)
version <i>optional</i>	integer (int32)
writeTime <i>optional</i>	integer (int64)

Task(任务)数据字段

Name	Description	Schema
accumulators <i>optional</i>	Task(任务)累加器	< string, string > map
aggregation <i>optional</i>	Task(任务)聚合器	string
attempt <i>optional</i>	Task(任务)意图 Example : 32	integer (int32)
completionTime <i>optional</i>	Task(任务)完成时间 Example : 1576494871454	integer (int64)
durations <i>optional</i>	Task(任务)的各持续时间项	< string, integer (int64) > map
executorId <i>optional</i>	执行该Task(任务)的Executor(执行器)ID Example : "32"	string
garbageCollectionTime <i>optional</i>	Task(任务)垃圾回收时间 Example : 10000	integer (int64)
host <i>optional</i>	执行Task(任务)的节点主机名 Example : "baisheng"	string
input <i>optional</i>	Task(任务)输入 Example : 10000	integer (int64)
message <i>optional</i>	执行Task(任务)的错误异常信息	string
metrics <i>optional</i>	执行Task(任务)的各指标	< string, string > map
port <i>optional</i>	执行Task(任务)的进程端口号 Example : 10000	integer (int32)
resultSerializationTime <i>optional</i>	Task(任务)结果序列化时间 Example : 10000	integer (int64)
schedulerDelay <i>optional</i>	Task(任务)调度延迟	integer (int64)
sequence <i>optional</i>	Task(任务)顺序	string
shuffleReadBytes <i>optional</i>	Task(任务)的shuffle度大小 Example : 100000	integer (int64)
shuffleSpillDisk <i>optional</i>	Shuffle磁盘溢出 Example : 1000	integer (int64)
shuffleSpillMemory <i>optional</i>	Shuffle内存溢出 Example : 1000	integer (int64)
shuffleWriteBytes <i>optional</i>	Task(任务)的shuffle写大小 Example : 100000	integer (int64)
stageId <i>optional</i>	Task(任务)所属Stage(调度阶段)ID Example : 32	integer (int64)
submissionTime <i>optional</i>	Task(任务)提交时间 Example : 1576494871454	integer (int64)

Name	Description	Schema
taskId <i>optional</i>	Task ID: Task(任务)唯一标识符 Example : 32	integer (int64)
taskIndexInStage <i>optional</i>	Task(任务)在Stage(调度阶段)中的索引 Example : 32	integer (int32)
taskLocality <i>optional</i>	Task(任务)本地能力	string
timeSpentFetchingTaskResults <i>optional</i>	拿取Task(任务)结果耗时 Example : 1000	integer (int64)
writeTime <i>optional</i>	Task(任务)写时间 Example : 1000	integer (int64)

Value

Type : object

持久化Executor(执行器)的Key结构

Name	Description	Schema
executorId <i>optional</i>	Executor ID: Executor的唯一标识符 Example : "32"	string
serverKey <i>optional</i>	Executor(执行器)对应的InceptorServer ID值 Example : "Inceptor::baisheng2::fb8ae28a-b9ec-480a-9a65-2ed392513e75"	string

持久化InceptorServer数据

Name	Description	Schema
environment <i>optional</i>	该InceptorServer的配置环境信息	< string, string > map
firstSeen <i>optional</i>	DBA Service第一次收到该InceptorServer的心跳时间戳 Example : 1577945712848	integer (int64)
host <i>optional</i>	InceptorServer的主机名 Example : "baisheng2"	string
key <i>optional</i>	Target(监控目标)的server Id全称 Example : "Inceptor::baisheng2::fb8ae28a-b9ec-480a-9a65-2ed392513e75"	string
localTimestamp <i>optional</i>	DBA Service收到最新一次server心跳时的本地时间戳 Example : 1577945713082	integer (int64)
port <i>optional</i>	InceptorServer的端口号 Example : 10000	integer (int32)
serverId <i>optional</i>	server ID, Target(监控目标)的编号标识 Example : 32	integer (int64)
timestamp <i>optional</i>	InceptorServer最新一次心跳的时间戳 Example : 1577945712848	integer (int64)

持久化InceptorServer统计数据

Name	Description	Schema
firstSeen <i>optional</i>	DBA Service第一次收到InceptorServer的心跳的时间戳 Example : 10	integer (int64)
host <i>optional</i>	InceptorServer的主机名 Example : "baisheng2"	string

Name	Description	Schema
lastSeen <i>optional</i>	DBA Service最后一次收到InceptorServer的心跳时间戳 Example : 10	integer (int64)
port <i>optional</i>	InceptorServer的端口号 Example : 10000	integer (int32)
serverKey <i>optional</i>	server ID, Target(监控目标)的编号标识 Example : "32"	string
sqlNumber <i>optional</i>	InceptorServer的Query(查询)数 Example : 10	integer (int64)

持久化Query(查询)的统计

Name	Description	Schema
cd <i>optional</i>	Query(查询)的实际计算时间 Example : 6658	integer (int64)
ct <i>optional</i>	Query(查询)彻底完成的时间 Example : 1577945712848	integer (int64)
e <i>optional</i>	Query(查询)有错 Example : 2	integer (int32)
id <i>optional</i>	Query ID: Query的唯一标识符 Example : 32	integer (int64)
jn <i>optional</i>	Query(查询)的Job(作业)数 Example : 2	integer (int64)
k <i>optional</i>	Query(查询)对应的InceptorServer ID值 Example : "Inceptor::baisheng2::fb8ae28a-b9ec-480a-9a65-2ed392513e75"	string
ltn <i>optional</i>	Query(查询)的本地Task(任务)数 Example : 2	integer (int64)
postcd <i>optional</i>	Query(查询)的提交计算结果的时间 Example : 9536	integer (int64)
precd <i>optional</i>	Query(查询)计算之前的准备时间 Example : 2182	integer (int64)
sn <i>optional</i>	Query(查询)的Stage(调度阶段)数 Example : 2	integer (int64)
st <i>optional</i>	Query(查询)提交时间 Example : 1577945712848	integer (int64)
tn <i>optional</i>	Query(查询)的Task(任务)数 Example : 165	integer (int64)

持久化的Executor(执行器)

Name	Description	Schema
firstSeen <i>optional</i>	DBA Service第一次收到Executor心跳的时间戳	integer (int64)
firstSeenLocal <i>optional</i>	DBA Service第一次收到Executor心跳时的本地时间戳	integer (int64)
gcInfo <i>optional</i>	Executor(执行器)上垃圾清理信息	< Executor(执行器)的GC(垃圾清理)信息 > array
host <i>optional</i>	InceptorServer节点的主机名 Example : "baisheng2"	string
jobs <i>optional</i>	Executor(执行器)上Job(作业)数据	< string, Job > map

Name	Description	Schema
key <i>optional</i>	Executor(执行器)Key值	持久化Executor(执行器)的Key结构
lastSeen <i>optional</i>	DBA Service最近一次收到Executor心跳的时间戳	integer (int64)
lastSeenLocal <i>optional</i>	DBA Service第一次收到Executor心跳时的本地时间戳	integer (int64)
localTasks <i>optional</i>	Executor(执行器)上本地Task(任务)的数据	< string, Task > map
maxMemory <i>optional</i>	Executor(执行器)最大内存 Example : 20000	integer (int64)
port <i>optional</i>	InceptorServer的端口号 Example : 10000	integer (int32)
sqls <i>optional</i>	Executor(执行器)上Query(查询)数据	< string, Sql > map
stages <i>optional</i>	Executor(执行器)上Stage(调度阶段)数据	< string, Stage > map
statuses <i>optional</i>	状态数据	< Status > array
tasks <i>optional</i>	Executor(执行器)上Task(任务)数据	< string, Task > map
tasksCompleted <i>optional</i>	Executor(执行器)上已执行完成的Task(任务)总数 Example : 200	integer (int64)
tasksFailed <i>optional</i>	Executor(执行器)上执行失败的Task(任务)总数 Example : 100	integer (int64)
timelines <i>optional</i>	Executor(执行器)上的时间轴	< Executor(执行器)的时间轴 > array
totalCores <i>optional</i>	Executor(执行器)的总核心数 Example : 2	integer (int32)
version <i>optional</i>	版本	integer (int32)

持久化的Query(查询) Key值结构定义

Name	Description	Schema
serverKey <i>optional</i>	Query(查询)对应的InceptorServer ID值 Example : "Inceptor::baisheng2::fb8ae28a-b9ec-480a-9a65-2ed392513e75"	string
sqlId <i>optional</i>	Query ID: Query的唯一标识符 Example : 32	integer (int64)

持久化的Query(查询)基本信息

Name	Description	Schema
completionTime <i>optional</i>	Query(查询)彻底完成的时间 Example : 1577945712848	integer (int64)
computeFinishTime <i>optional</i>	Query(查询)计算完成的时间 Example : 1577945712848	integer (int64)
description <i>optional</i>	Query(查询)的描述信息 Example : "create table e_holodesk stored as holodesk with tablesize 10GB as select * from e limit 30000000"	string
duration <i>optional</i>	Query(查询)的持续时间 Example : 18376	integer (int64)
hasError <i>optional</i>	Query(查询)有错 Example : 2	integer (int32)

Name	Description	Schema
host <i>optional</i>	InceptorServer的主机名 Example: "baisheng2"	string
jobNumber <i>optional</i>	Query(查询)的Job(作业)数 Example: 2	integer (int64)
localTime <i>optional</i>	Query(查询)的本地Task(任务)时间 Example: 2	integer (int64)
message <i>optional</i>	错误异常信息 Example: "Query is either failed, killed, or something unexpected happened.\nState: ERROR\nOperation Exception: org.apache.hive.service.cli.HiveSQLException: Found exception during run crux query: Parser Exception: \n query: match (a) →[f]-(b) return f limit 2 \nSyntax Error on: Syntax Error : Line 1, column 10 at '' : no viable alternative at input 'match (a)→'50"	string
port <i>optional</i>	InceptorServer的端口号 Example: 10000	integer (int32)
serverKey <i>optional</i>	Query(查询)对应的InceptorServer ID值 Example: "Inceptor::baisheng2::fb8ae28a-b9ec-480a-9a65-2ed392513e75"	string
sessionId <i>optional</i>	Query(查询)对应的Session(会话) ID Example: 32	integer (int64)
sqlId <i>optional</i>	Query ID: Query的唯一标识符 Example: 32	integer (int64)
stageNumber <i>optional</i>	Query(查询)的Stage(调度阶段)数 Example: 2	integer (int64)
startTime <i>optional</i>	Query(查询)开始执行的时间 Example: 1577945712848	integer (int64)
submissionTime <i>optional</i>	Query(查询)提交时间 Example: 1577945712848	integer (int64)
taskNumber <i>optional</i>	Query(查询)的Task(任务)数 Example: 165	integer (int64)
user <i>optional</i>	发起该Query(查询)的用户 Example: "anonymous"	string

持久化的Query(查询)数据字段

Name	Description	Schema
completionTime <i>optional</i>	Query(查询)彻底完成的时间 Example: 1577945712848	integer (int64)
computeDuration <i>optional</i>	Query(查询)的实际计算时间 Example: 6658	integer (int64)
computeFinishTime <i>optional</i>	Query(查询)计算完成的时间 Example: 1577945712848	integer (int64)
description <i>optional</i>	Query(查询)的描述信息 Example: "create table e_holodesk stored as holodesk with tablesize 10GB as select * from e limit 30000000"	string
detailedDescription <i>optional</i>	详细的描述信息 Example: "create table e_holodesk stored as holodesk with tablesize 10GB as select * from e limit 30000000"	string
duration <i>optional</i>	Query(查询)的持续时间 Example: 18376	integer (int64)
durations <i>optional</i>	Query(查询)多个分项的持续时间集合(K/V) Example: "Transaction Ops\t821837\nData Move\t9536\nTest 2\t410320\nTest 3\t954552\nCompile\t879817\nTest 1\t906197"	< string, integer (int64) > map
extraJobId <i>optional</i>	Query(查询)的额外Job ID Example: 1024	integer (int64)

Name	Description	Schema
stages <i>optional</i>	Query(查询)之下的Stage(调度阶段)数据(Map<K,V>类型)	< string, Stage > map
startTime <i>optional</i>	Query(查询)开始执行的时间 Example : 1577945712848	integer (int64)
submissionTime <i>optional</i>	Query(查询)提交时间 Example : 1577945712848	integer (int64)
tableScans <i>optional</i>	Query(查询)的扫描表	< string, TableInfo > map
taskInfos <i>optional</i>	Task(任务)信息	< ColdTaskInfo > array
taskNumber <i>optional</i>	Query(查询)的Task(任务)数 Example : 165	integer (int64)
tasks <i>optional</i>	Query(查询)之下的Task(任务)数据(Map<K,V>类型)	< string, Task > map
totalTaskTime <i>optional</i>	Query(查询)的所有Task(任务)的持续时间总和 Example : 130454	integer (int64)
user <i>optional</i>	发起该Query(查询)的用户 Example : "anonymous"	string