TOC

Composer 24 ..... **4**

Composer 24 Reference Information 4

Reference Information **5**

The Composer UI Menu **7**

UI Menu Options 7

The Top-Level Navigation Banner **10**

The Composer Supervisor Menu **11**

Composer Microservice Name Reference **12**

Composer Microservice Startup Order **14**

Composer Metadata Stores **15**

Composer Log Files Reference **16**

Composer Log Unification **17**

Log Configuration Properties 17

Log File Values 18

Other Configuration Properties 18

Zoomdata-Web 18

Query Engine 19
Set Logging Levels 20
Default Log File Names 20
Log File Migration Reference **22**
Zoomdata-web 22
Query Engine 22
EDC 23
Admin Service, Config Server, Data Writer, and Screenshot Service 23
Default Port Reference **24**
Application Configuration Object **26**
Example 26
Query Configuration Object **27**
Example 27
tz: '' 28
time: {} 28
player: {} 29
filters: [{},{}] 30
groups: [{},{}] 31

sort: {} 31

fields: [{},{}] 32

metrics: [{},{}] 32

Security Configuration Object **33**

Example 33

Supplied Color Palettes **34**

Operators **36**

State Name Reference **37**

Country Name Reference **39**

Preset Time Ranges **41**

Errors and Exceptions **44**

createQuery 44

Events Created by Client **45**

Thread Events 45

Visualization Events 45

Cautionary Note About Internal APIs **47**

Composer 24

Composer 24 Reference Information

Reference Information

The following topics provide reference material useful for managing and using Composer.

- Composer Microservice Name Reference
- Composer Microservice Startup Order
- Composer Metadata Stores
- Composer Log Files Reference
- Default Port Reference
- Application Configuration Object
- Query Configuration Object
- Data Connector Reference
- Visual Type Configuration Properties
- ComposerSymphony Data Discovery Visual Metrics And Attributes Reference
- Supplied Color Palettes
- Operators
- State Name Reference
- Country Name Reference
- Preset Time Ranges
- Errors And Exceptions

- Events Created By Client
- Cautionary Note About Internal APIs

The Composer UI Menu

Navigate to the user interface (UI) menu by selecting the hamburger icon (☰) in the upper right corner your software. Menu options vary: users who are members of the Supervisors or Administrators group, and users with specific group privileges have access to different options.

The table below describes menu options available to user groups in Composer in v23.4 and later.

Note: The default **supervisor** user is no longer installed; add users to the **Supervisors** group instead.

UI Menu Options

Menu Option	Select this option to...	Default Users	Supervisors Group	Administrators Group
Sources	Review and manage data source configurations. See Manage Visual Data Discovery Data Sources.	Yes	Yes	Yes
Visual Gallery	Review and manage shared visuals. See Use the Visual Gallery.	Yes	Yes	Yes
Library	Review and manage dashboards in the dashboard library. See Use the Dashboard Library.	Yes	Yes	Yes
Help	Access the Composer documentation.	Yes	Yes	Yes
Support	Access Technical Support.	Yes	Yes	Yes
About	View the About dialog that identifies the version of Composer you are using and provides a link to the terms of service. See About Dialog.	Yes	Yes	Yes
My Tenants	Switch to a different Composer tenant. **Note:** In previous versions of Composer, My Accounts is used.	Yes	Yes	Yes
Log out	Log out of Composer.	Yes	Yes	Yes
Connections	Review and manage connection definitions used by Composer connectors. See Manage Data Discovery Data Store Connections.	If added to a group definition.	No	Yes
Users and Groups	Manage Composer users and groups and to assign users to groups. See Manage Users and Manage User Groups in Visual Data Discovery.	If added to a group	No	Yes

Menu Option	Select this option to...	Default Users	Supervisors Group	Administrators Group
		definition.		
Custom Charts	Manage custom Composer visuals from the UI. See Manage Custom Charts.	If added to a group definition.	No	Yes
Console	Monitor data source refresh jobs. See Review Refresh Jobs.	If added to a group definition.	No	Yes
Actions	Define action templates. See Integrate Visual Data Into Your Applications.	If added to a group definition.	No	Yes
System Users	Manage users and assign them to accounts. See Manage Users.	If added to a group definition.	Yes	System Administrator only.
Multi-Tenancy	Manage tenants (formerly accounts); creation, editing, disabling, and deletion.	If added to a group definition.	Yes	System Administrator only.
Customize UI	Customize the UI. See Customize the Composer Visual Data Discovery User Interface.	If added to a group definition.	Yes	System Administrator only.
Security	Manage the authentication tools used by Composer. See Symphony Supported Authentication Tools.	If added to a group definition.	Yes	System Administrator only.
Connectors	Manage the connector servers available in Composer. See Manage Connectors and Connector Servers.	If added to a group definition.	Yes	System Administrator only.
Advanced	Review Composer server-level variables. We suggest you change these only at the request of Technical Support. See Server-Level Variables	If added to the group definition.	Yes	System Administrator only.
License	Specify a new Composer license key. See Request and Apply a New License Key.	If added to a group definition.	Yes	System Administrator only.

Note: The default **supervisor** user is no longer installed; add users to the **Supervisors** group instead. See The Composer Supervisor Menu.

The Top-Level Navigation Banner

The top-level navigation banner appears in the Composer UI when you are logged in as a user. This menu provides easy access to connection definitions, data source configuration definitions, and the dashboards in the UI. It looks like this:

Each option is described in the following table.

Option	Description
Composer logo	Select the Composer logo to return to the Home page from anywhere in the product. See Home Page.
Home	Select this option to return to the Home page from anywhere in the product. See Home Page.
Connections	Select this option to review and manage the connection definitions used by Composer connectors. See Manage Data Discovery Data Store Connections.
Sources	Select this option to review and manage the data source configuration definitions used for Composer queries. See Manage Visual Data Discovery Data Sources
Visual Gallery	Select this option to review and manage stored visuals you can use in multiple dashboards. See Manage Visuals.
Library	Select this option to review and manage dashboards in the dashboard list from anywhere in the product. See Manage Dashboards.

The Composer Supervisor Menu

When you log in as a user that belongs to the Supervisors group, you have default access to more options than are in the UI menu for other users. The supplied admin user and other system administrators are members of the Supervisors group.

Note: The default **supervisor** user is no longer installed; add users to the **Supervisors** group instead.

Navigate to the menu by selecting the UI menu icon () in the upper right corner your software. The table below describes menu options available to members of the Supervisors group.

Menu Option	Select this option to...	Supervisors Group
Sources	Review and manage data source configurations. See Manage Visual Data Discovery Data Sources.	Yes
Visual Gallery	Review and manage visuals. See Use the Visual Gallery.	Yes
Library	Review and manage dashboards in the library. See Use the Library for Dashboards.	Yes
Help	Access the Composer documentation.	Yes
Support	Access Technical Support.	Yes
About	View the About dialog that identifies the version of Composer you are using and provides a link to the terms of service. See About Dialog.	Yes
My Tenants	Switch to a different Composer tenant. **Note:** In previous versions of Composer, My Accounts is used.	Yes
Log out	Log out of Composer.	Yes
System Users	Manage users and assign them to accounts. See Manage Users.	Yes
Multi-Tenancy	Manage tenants: creation, editing, disabling, and deletion.	Yes
Customize UI	Customize the UI. See Customize the Composer Visual Data Discovery User Interface.	Yes
Security	Manage the authentication tools used by Composer. See Symphony Supported Authentication Tools.	Yes
Connectors	Manage the connector servers available in Composer. See Manage Connectors And Connector Servers.	Yes
Advanced	Review Composer server-level variables. We suggest you change these only at the request of Technical Support. See Server-Level Variables.	Yes
License	Specify a new Composer license key. See Request and Apply a New License Key.	Yes

Composer Microservice Name Reference

The following table lists the Composer microservices. The order in which microservices should be started is described in Composer Microservice Startup Order.

Microservice Name	Description
zoomdata	The primary microservice for the Composer server.
zoomdata-admin-server	The Service Monitor microservice.
zoomdata-config-server	The configuration microservice.
zoomdata-consul	The Service Discovery microservice.
zoomdata-data-writer-postgresql	The Data Writer microservice.
zoomdata-edc-apache-solr	The Apache Solr connector microservice.
zoomdata-edc-bigquery	The BigQuery connector microservice.
zoomdata-edc-cloudera-search	The Cloudera Search connector microservice.
zoomdata-edc-dremio	The Dremio connector microservice.
zoomdata-edc-drill	The Apache Drill connector microservice.
zoomdata-edc-elasticsearch-7.0	The Elasticsearch 7.0 connector microservice.
zoomdata-edc-elasticsearch-8-0	The Elasticsearch 8.0 connector microservice.
zoomdata-edc-hdfs	The HDFS connector microservice.
zoomdata-edc-hive	The Hive connector microservice.
zoomdata-edc-impala	The Cloudera Impala connector microservice.
zoomdata-edc-memsql	The MemSQL connector microservice.
zoomdata-edc-mongo	The MongoDB connector microservice.
zoomdata-edc-mssql	The Microsoft SQL Server connector microservice.
zoomdata-edc-mysql	The MySQL connector microservice.
zoomdata-edc-oracle	The Oracle connector microservice.
zoomdata-edc-phoenix-4.7	The Apache Phoenix 4.7 connector microservice.
zoomdata-edc-phoenix-4.7-queryserver	The Apache Phoenix Query Server 4.7 connector microservice.
zoomdata-edc-postgresql	The PostgreSQL connector microservice.
zoomdata-edc-redshift	The Amazon Redshift connector microservice.
zoomdata-edc-rts	The Real-Time Sales connector microservice.

Microservice Name	Description
zoomdata-edc-s3	The Amazon S3 connector microservice.
zoomdata-edc-saphana	The SAP Hana connector microservice.
zoomdata-edc-saphanacloud	The SAP S/4HANA connector microservice.
zoomdata-edc-sapiq	The SAP IQ connector microservice.
zoomdata-edc-snowflake	The Snowflake connector microservice.
zoomdata-edc-sparksql	The Spark SQL connector microservice.
zoomdata-edc-teradata	The Teradata connector microservice.
zoomdata-edc-tibcodv	The TIBCO DV connector microservice.
zoomdata-edc-trino	The Trino connector microservice.
zoomdata-edc-vertica	The Vertica connector microservice.
zoomdata-postgres	The metadata repository.
zoomdata-query-engine	The query engine microservice.
zoomdata-screenshot-service	The Screenshot microservice.

Composer Microservice Startup Order

The only microservices that must be started first are the Postgres and Service Discovery microservices. The other microservices are more tolerant and can be started in any order.

Composer recommends starting microservices in the following order. You can start microservices using CentOS or Ubuntu commands or using the CLI. See Start ComposerSymphony Microservices and Manage ComposerSymphony Microservices Using The Command Line Utility. A full list of Composer microservices can be found in Composer Microservice Name Reference.

1. The Postgres microservice (`postgresql-<version>`) used for metadata storage (where `<version>` is the version of Postgres you have installed.
2. The Service Discovery microservice (`zoomdata-consul`) used for microservice discovery.
3. The connector microservices (in the format `zoomdata-edc-<connector_name>`) used to connect to different data stores. The following CentOS example starts all available connector microservices, rather than starting them individually:

```
sudo systemctl start $(systemctl list-unit-files | grep zoomdata-edc | awk '{print $1}')
```

4. The query engine microservice (`zoomdata-query-engine`) used for query engine processing.
5. The `zoomdata-data-writer-postgresql` microservice.
6. The following optional microservices, if they are installed.

i. `zoomdata-admin-server`

ii. `zoomdata-screenshot-service`

7. The main Composer microservice (`zoomdata`).

Composer Metadata Stores

Composer uses the following metadata stores:

Store Name	Description
zoomdata	Stores Composer metadata. This is the primary metadata store for Composer.
zoomdata-config	Stores metadata associated with the Composer configuration microservice. This store name is also configurable.
zoomdata-keyset	Stores keyset data.
zoomdata-upload	Stores flat files that have been uploaded using the flat file connector.

Composer Log Files Reference

Composer uses the following log files. You can configure Composer to write these logs to the console in addition to or instead of the service files. See Configure ComposerSymphony Logs.

Log File	Contains
`edc-<connector>.log`	Specific data connector service log file.
`edc-<connector>-errors.log`	Specific data connector service log file, filtered to ERROR level log messages.
`query-engine.log`	Operational information about the query engine microservice.
`query-engine-access.log`	Standard web server access log for the query engine microservice.
`query-engine-activity.log`	Audit-related information for the query engine microservice.
`Security_service.log`	Information related to enabling or disabling a security service in the supervisor UI.
`stream-writer.log`	Upload service information.
`zoomdata.log`	A wide variety of errors and information. This is the best place to start with troubleshooting.
`zoomdata-access.log`	Standard web server access log for the Composer microservice.
`zoomdata-consul-{unixTimestamp}.log`	Information on service discovery and registration.
`zoomdata-errors.log`	Composer log information with a log level of `ERROR`.
`zoomdata-installer.<date/time>.log`	Information related to the installation. Installation errors typically appear here.
`zoomdata-upgrade.log`	Information related to upgrades. Errors with an upgrade will typically show up here.
`zoomdata-websocket.log`	Information sent and received over a WebSocket from the browser.

Composer Log Unification

Composer has unified logback configuration throughout Composer microservices. You can configure Composer to write these logs to the console in addition to or instead of the service files. See Configure ComposerSymphony Logs and Composer Log Files Reference.

Log Configuration Properties

Adjust these properties to set up logging for your environment.

Name	Default Value	Description
install.dir	.	Use to evaluate the value of `logs.dir`.
logs.dir	`${install.dir}/logs`	The location of the log file and error log file.
spring.application.name	`zoomdata-service`	Use to evaluate the default values of `logs.file-name`.
logs.file-name	`${spring.application.name}`	Name of log file. Log file suffix: `.log`. Error log file suffix: `error.log`.
log.console.level	`ALL`	The maximum level of logs to print to `stdout`. See Log File Values.
log.file.level	`ALL`	The maximum level of logs to print to the log file. See Log File Values.
log.password.regex	`(?i)PASSWORD=[a-zA-Z0-9!@#\$%\^\*\&]*`	Regular expression used to detect passwords in log output. Replaces with `PASSWORD=****`.
log.url-password.regex	`(?<=://)[^:]+:[^@]+@`	Regular expression used to detect and remove credential in URLs and remove the credentials.
log.sanitization.enabled	`true`	The default, `true`, is specified in `logging.properties`. Toggles sanitization of log message via `org.apache.commons.text.StringEscapeUtils.escapeJava`.
log.file.size	`20`	The maximum size of the log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}.log.%i`.
log.file.count	`5`	Maximum number of log files.
log.error.file.size	`5`	The maximum size of the error log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-error.log.%i`.
log.error.file.count	`5`	Maximum number of error log files.
syslog.log.level	`OFF`	Syslog log level. See Log File Values.
syslog.host	`127.0.0.1`	Syslog host name.
syslog.port	`514`	Syslog port number.

Name	Default Value	Description
syslog.suffix	`${spring.application.name}`	Syslog suffix. Used to distinguish applications.
trace.requests	`false`	Adds the MDC value of `requestId` to log output. Propagate value using `X-REQUEST-ID` http header.

Log File Values

Value	Description
`OFF`	Disables logging.
`FATAL`	Only error messages (the same as `ERROR` for logback).
`ERROR`	Application error messages that may affect processes.
`WARN`	Unexpected application issues that may not affect processes.
`INFO`	Expected activities.
`DEBUG`	Triggers capture of `WARN`, `INFO`, and `ERROR`information.
`TRACE`	Capture of `DEBUG` level of information, and low importance informational messages.
`ALL`	All available log information (the same as `TRACE` for logback).

Other Configuration Properties

Both Zoomdata-Web and Query Engine have properties you can adjust to suit your needs as well.

Zoomdata-Web

Name	Default Value	Description
activity.log.file.count	`1`	Maximum number of activity log files .
activity.log.file.size	`10`	The maximum size of the activity log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-activty.log.%i`.
activity.logs.dir	n/a	The location of the activity log file. If no location is specified, `{logs.dir}` is used.
websocket.log.file.size	`10`	The maximum size of the websocket log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-websocket.log.%i`.
websocket.log.file.count	`5`	Maximum number of websocket log files.

Name	Default Value	Description
websocket.log.level	OFF	Overall maximum log level of websocket logs.
websocket.file.level	OFF	Maximum log level of websocket logs in the file.
websocket.console.level	OFF	Maximum log level of websocket logs in the console.
access.log.file.size	10	The maximum size of the access log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-access.log.%i`.
license.log.file.size	10	The maximum size of the license log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-sessions.log.%i`.

Query Engine

Name	Default Value	Description
websocket.log.level	OFF	Overall maximum log level of websocket logs.
websocket.file.level	OFF	Maximum log level of websocket logs in the file.
websocket.console.level	OFF	Maximum log level of websocket logs in the console.
websocket.log.file.size	10	The maximum size of the websocket log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-websocket.log.%i`.
websocket.log.file.count	1	Maximum number of websocket log files.
access.log.file.size	10	The maximum size of the access log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-access.log.%i`.
slow-request.file.level	INFO	Maximum log level of slow request logs in the file.
slow-request.console.level	OFF	Maximum log level of slow request logs in the console.
slow-request.log.file.size	10	The maximum size of the slow requests log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-slow-requests.log.%i`.
slow-request.log.file.count	5	Maximum number of slow requests log files.
slow-expression-parsing.file.level	INFO	Maximum log level of slow expression parsing logs in the file.
slow-expression-parsing.console.level	OFF	Maximum log level of slow expression parsing logs in the console.
slow-expression-parsing.log.file.size	10	The maximum size of the slow expression parsing log file in MB. When that size is reached, the file is renamed using this pattern: `{log.file-name}-slow-expression-parsing.log.%i`
slow-expression-parsing.log.file.count	5	Maximum number of slow expression parsing log files.

Set Logging Levels

Every logging level can be changed using a standard spring boot approach. Specify the corresponding property, as shown below.

Note: The logback.zoomdata property, previously used to control the log level of some loggers, has been removed. Use a standard spring boot approach to specify appropriate properties.

```
# Enable debug logs for 'com.zoomdata' logger and it's children
logging.level.com.zoomdata=DEBUG
# Set log level to error for root logger (parent of all loggers) and it's children
logging.level.root=ERROR
```

Default Log File Names

Composer's microservices each have a series of default log files where you can find captured events.

Name	Microservice	Description
zoomdata.log	Zoomdata-web	Common logs.
zoomdata-errors.log	Zoomdata-web	Only server error logs.
zoomdata-access.log	Zoomdata-web	Access logs.
zoomdata-activity.log	Zoomdata-web	Activity audit logs.
zoomdata-websocket.log	Zoomdata-web	Websocket logs.
zoomdata-sessions.log	Zoomdata-web	Licensing related logs.
query-engine.log	Query Engine	Common logs.
query-engine-errors.log	Query Engine	Only server error logs.
query-engine-access.log	Query Engine	Access logs.
query-engine-websocket.log	Query Engine	Websocket logs.
query-engine-slow-expression-parsing.log	Query Engine	
query-engine-slow-requests.log	Query Engine	
admin-server.log	Admin Service	Common logs.
admin-server-errors.log	Admin Service	Only server error logs.
config-server.log	Config Server	Common logs.
config-server-errors.log	Config Server	Only server error logs.

Name	Microservice	Description
`stream-writer-postgresql.log`	Data-Writer	Postgresql: Common logs.
`stream-writer-postgresql-errors.log`	Data-Writer	Postgresql: Only server error logs.
`stream-writer-mssql.log`	Data-Writer	MSSQL: Common logs.
`stream-writer-mssql-errors.log`	Data-Writer	MSSQL: Only server error logs.
`screenshot-service.log`	Screenshot Service	Common logs.
`screenshot-service-errors.log`	Screenshot Service	Only server error logs.
`${connector.name}.log`	EDC	Common logs.
`${connector.name}-errors.log`	EDC	Only server error logs.

Log File Migration Reference

With the introduction of unified log properties and unified logback configurations, a number of log files have new names. When you upgrade your Composer environment from an earlier version, you may need to rename properties overwritten by external configurations.

Zoomdata-web

Current Name	Deprecated Name
log.file.size	zoomdata.log.file.size
log.file.count	zoomdata.log.rolling.maxIndex
log.error.file.size	zoomdata.error.log.file.size
access.log.file.size	zoomdata.access.log.file.size
license.log.file.size	zoomdata.license.log.file.size
activity.log.file.count	zoomdata.activity.log.file.max.index
activity.log.file.size	zoomdata.activity.log.file.size
activity.logs.dir	zoomdata.activity.logs.dir
websocket.log.file.size	zoomdata.websocket.log.file.size
websocket.log.file.count	zoomdata.websocket.log.rolling.maxIndex

Query Engine

Current Name	Deprecated Name
log.file.size	qe.log.file.size
log.file.count	qe.log.rolling.maxIndex
log.error.file.size	qe.error.log.file.size
websocket.log.file.count	qe.websocket.log.rolling.maxIndex
websocket.log.level	qe.websocket.log.level
websocket.log.file.size	qe.websocket.log.file.size
access.log.file.size	qe.access.log.file.size

EDC

Current Name	Deprecated Name
logs.file-name	log.file.base.name

Admin Service, Config Server, Data Writer, and Screenshot Service

Current Name	Deprecated Name
log.file.count	log.rolling.maxIndex
log.error.file.size	error.log.file.size
log.error.file.count	error.log.rolling.maxIndex

Default Port Reference

The following table lists the default ports used by Composer, sorted by port number.

Port	Microservice	Description
443	`zoomdata`	Composer server web communication. For HTTPS requests, configure your firewall to map port 443 to either port 8443 or port 8080.
5432	`zoomdata-postgres`	Composer metadata repository.
5580	`zoomdata-query-engine`	Composer query engine microservice.
8050	`zoomdata-admin-server`	Service Monitor.
8080	`zoomdata`	Composer server.
8081	`zoomdata-data-writer`	Composer Data Writer microservice.
8083	`zoomdata-screenshot-service`	Composer Screenshot microservice.
8093	`zoomdata-edc-bigquery`	BigQuery connector.
8095	`zoomdata-edc-drill`	Apache Drill connector.
8098	`zoomdata-edc-impala`	Cloudera Impala connector.
8099	`zoomdata-edc-memsql`	MemSQL connector.
8100	`zoomdata-edc-mssql`	Microsoft SQL Server connector.
8101	`zoomdata-edc-mysql`	MySQL connector.
8102	`zoomdata-edc-oracle`	Oracle connector.
8105	`zoomdata-edc-postgresql`	PostgreSQL connector, Flat File and API uploads for processing.
8108	`zoomdata-edc-rts`	Real Time Sales connector.
8109	`zoomdata-edc-saphana`	SAP Hana connector.
8111	`zoomdata-edc-teradata`	Teradata connector.
8112	`zoomdata-edc-vertica`	Vertica connector.
8115	`zoomdata-edc-apache-solr`	Apache Solr connector.
8116	`zoomdata-edc-sparksql`	Spark SQL connector.
8121	`zoomdata-edc-sapiq`	SAP IQ connector.
8123	`zoomdata-edc-mongo`	MongoDB connector.
8124	`zoomdata-edc-phoenix-4.7`	Apache Phoenix connector.
8125	`zoomdata-edc-phoenix-4.7-queryserver`	Apache Phoenix Query Server (QS) connector.

Port	Microservice	Description
8126	zoomdata-edc-hdfs	HDFS connector.
8129	zoomdata-edc-s3	Amazon S3 connector.
8131	zoomdata-edc-snowflake	Snowflake connector.
8132	zoomdata-edc-hive	Hive connector.
8138	zoomdata-edc-couchbase	Couchbase connector.
8139	zoomdata-edc-elasticsearch-7.0	Elasticsearch 7.0 connector.
8140	zoomdata-edc-tibcodv	TIBCO DV connector.
8142	zoomdata-edc-dremio	Dremio connector.
8147	zoomdata-edc-elasticsearch-8-0	Elasticsearch 8.0 connector.
8148	zoomdata-edc-trino	Trino connector.
8201	zoomdata-edc-cloudera-search	Cloudera Search connector.
8202	zoomdata-edc-redshift	Amazon Redshift connector.
8205	zoomdata-edc-saphanacloud	SAP S/4HANA connector.
8300	zoomdata-consul	Internal port used by the Consul for inter-node communication in a distributed environment.
8301	zoomdata-consul	Internal port used by the Consul for inter-node communication in a distributed environment.
8302	zoomdata-consul	Internal port used by the Consul for inter-node communication in a distributed environment.
8443	zoomdata	Composer HTTPS requests.
8500	zoomdata-consul	Consul.
8888	zoomdata-config-server	Composer configuration microservice.

Application Configuration Object

The application configuration object contains the parameters that your client needs to identify a Composer server.

Example

```
var applicationConfig = {
secure: true,
host: 'www.yourcompanysite.com',
port: 8443,
path: '/zoomdata'
};
```

The values for each key should be as follows:

- **secure**: true to use HTTPS (secure) protocol, otherwise `false`
- **host**: the base URL where your Composer server is hosted
- **port**: the port that your Composer uses to communicate; by default, this is `8443`
- **path**: the path from your base URL for your Composers server; by default, this is `'/zoomdata'`

Query Configuration Object

The query configuration is contained in a JSON object including specific key:value pairs. You do not need to include unused keys. The exceptions are:

- the `time` key, which must specify a `timeField` value if a `player` is to be used by the query
- you must use one and only one of the following keys: `fields`, `groups`, and `dimensions`. Using none or more than one of these keys results in an error.

Additionally, if you use the query configuration object to create a query that is then used to supply data to an embedded visual, it must have the following keys:

- `filters`, though it can be defined as `[]` if you do not want to apply any filter
- `metrics`, without which the visual will not have data to present

Example

```
var yourConfigObject = {
       // time zone
       tz: 'UTC',

       // field used for time
       // start and finish times/dates
       // format: {+/-}YYYY-MM-DD HH:mm:ss.SSS
       // + means inclusive, - means exclusive
       time: {
               timeField: '_ts',
               from: '+2016-10-23 22:30:15.500',
               to: '-2016-11-23 22:30:15.500'
       },

       // rate of querying, in seconds
       // timeWindowScale: 'ROLLING' means that start time advances with end time
       // timeWindowScale: 'PINNED' means that start time does not advance with end time
       player: {
               speed: 1,
               pauseAfterRead: false,
               timeWindowScale: 'ROLLING'
       },

       filters: [
```

```
                { path: 'usersentiment',  operation: 'BETWEEN', value: [-0.5, 0.5] },
                { path: 'color',  operation: 'IN',  value: [‘red’, 'green', 'blue'] },
                { path: 'age', operation: 'LT', value: 21 },
                { path: 'color', operation: 'NOTIN', value: ['red', 'green', 'blue'] }
        ],

        // count does not take a metric function. Other metrics require one.
        // func can be 'sum', 'avg', 'min', 'max', 'count', 'countd'
        metrics: [
                { name: 'price', func: 'avg' },
                { name: 'count' }
        ],

        // metricFunc is used to sort by metric values. It is required if you sort by a metric other than count. It accepts the
same options as metric's func key.
        groups: [
                { name: 'usercity', limit: 200, sort: { dir: 'asc', name: 'usercity' } },
                { name: 'department', limit: 10, sort: { dir: 'asc', name: 'count'  } },
                { name: 'color', limit: 5, sort: { dir: 'asc', name: 'usersentiment', metricFunc: 'avg' } }
        ],

        // Your query must have groups or fields, but may not have both groups and fields.
        fields: [
                { name: 'price', limit: 50 },
                { name: 'usercity', limit: 50 }
        ],
};
```

Each possible parameter is described below.

tz: ''

Sets the timezone used by the web app. Use ISO standard abbreviations. The value defaults to the Composer server time.

time: {}

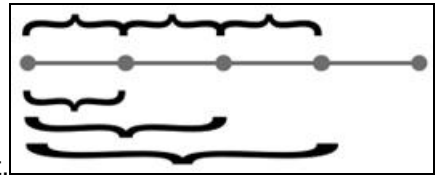
A time object, which specifies the time range to be included in the query. Essentially, this parameter filters the query based on time using the time field, from, and to elements.

Key	Usage	Notes	Examples
`timeField`	String.	This field refers to the data source columns	

Key	Usage	Notes	Examples
	Sets the field to be used as the time measurement.	containing the time measurement, rather than any label that appears in the user interface.	
`from`	String. Sets the earliest time from which data should be included in the query.	Use the format `[-/+]YYYY-MM-DD HH:mm:ss.SSS` (preceded by a + or - to include or exclude the specified time).	`+2024-10-23 10:30:15.500` to start at 23 October 2024 10:30:15.5, inclusive
`to`	String. Sets the latest time to which data should be included in the query.	Use the format `[-/+]YYYY-MM-DD HH:mm:ss.SSS` (preceded by a + or - to include or exclude the specified time).	`-2025-01-10 10:30:15.500` to start at 10 January 2025 10:30:15.5, exclusive.

player: {}

A player object, which specifies parameters for retrieving data from a source. It consists of the speed, pauseAfterRead, and timeWindowScale elements. If you include a player object in the query, you must also include a time object that includes at least the timeField element.

Key	Usage	Notes	Examples
`speed`	Integer. Sets the interval at which data should be retrieved, in seconds.	Accepted values are: 1 (every second), 60 (every minute), 3600 (every hour), and 86400 (daily).	60
`pauseAfterRead`	Boolean. Sets whether the data stream, if applicable, should be paused after initial retrieval.	`data set empty` message may result when `pauseAfterRead` is set to false with a non-live data source.	true
`timeWindowScale`	String. `'ROLLING'` or `'PINNED'`. Sets whether the time interval of retrieved data “rolls” or stays “pinned” to its original starting point (the from element of the time object).	The retrieval of a rolling time window starts at the point where the last retrieval finished. The retrieval of a pinned time window always starts at the same point, which is provided by the from element. The image below shows rolling time windows above the timeline, and pinned windows beneath it. 	`'ROLLING'` `'PINNED'`

filters: [{},{}]

Array of filter objects. Filter objects are applied to the data query using logical-and operations so that data is passed from the data query only if it satisfies all applied filters. Each filter consists of a path, an operation, and a value.

<table>
<tr><th>Key</th><th>Usage</th><th>Notes</th><th>Examples</th></tr>
<tr><td>path</td><td>String.
The name of the group or metric on which the filter operates.</td><td>This name is that of the data source column containing the group or metric of the filter, rather than the label that appears in the user interface.</td><td>customer_gender
customer_age
user_occupation</td></tr>
<tr><td>operation</td><td>String.
The logical operator used by the filter.</td><td>Valid options include the following:
<table>
<tr><th>logical operation</th><th>key value</th></tr>
<tr><td><</td><td>'LT'</td></tr>
<tr><td><=</td><td>'LE'</td></tr>
<tr><td>==</td><td>'EQUALS' (case sensitive)</td></tr>
<tr><td>==</td><td>'EQUALSI' (case insensitive)</td></tr>
<tr><td>>=</td><td>'GE'</td></tr>
<tr><td>></td><td>'GT'</td></tr>
<tr><td>in a set</td><td>'IN'</td></tr>
<tr><td>not in a set</td><td>'NOTIN'</td></tr>
<tr><td>between two values</td><td>'BETWEEN'</td></tr>
<tr><td>!=</td><td>'NOTEQUALS'</td></tr>
<tr><td>text search</td><td>'TEXT_SEARCH'</td></tr>
<tr><td>filter-level AND</td><td>'AND'</td></tr>
<tr><td>filter-level OR</td><td>'OR'</td></tr>
</table></td><td>'EQUALSI'
'BETWEEN'
'NOTIN'</td></tr>
</table>

Key	Usage	Notes	Examples
`value`	Single value or array of values. The value(s) considered by the logical operation.	If two or more values are needed, such as for a logical set or for a 'BETWEEN' operation, they should be provided in an array. Single values, such as required for an 'EQUALS' or 'LT' operation, should be provided as simple values.	`'female' [21,65]` `['teacher','lawyer', 'plumber']`

groups: [{},{}]

Array of group objects. Group objects specify which data is returned by a query and how it is grouped. Each group object has a name, limit, and a sort object.

Note: If `groups` is defined, `fields` and `dimensions` must be omitted.

Key	Usage	Notes	Examples
`name`	String. The name of the attribute to be used for grouping data.	This name is the name of the data source column containing the attribute, rather than the label that appears in the user interface.	`'home_state'`
`limit`	The maximum number of distinct items to be included in the data set.	If the limit is set lower than the total number of items in a group, not all members of the group will be included in the returned data set. For example, if the limit is set to 25, then the group us_state could not return data from all fifty US states.	50
`sort`	A sort object. See below.		`{dir: 'desc', name: 'home_state'}`

sort: {}

A sort object. Sort objects describe the way in which a group is ordered. Each sort object consists of the name of the group or metric for the sorting and the direction in which the group's data is sorted.

Key	Usage	Notes	Examples
`name`	String. The name of the group or metric on which the	This name is that of the data source column containing the group or metric by which groups are to be sorted, rather than	`'home_state'`

Key	Usage	Notes	Examples
	filter operates.	the label that appears in the user interface.	
`dir`	String. The direction of the sorting.	Valid options are `'asc'` and `'desc'` for ascending and descending sorts, respectively.	`'asc'`
`metricFunc`	String. Required to sort by a metric value. Not permitted if you sort by count or by a group.	Valid options include the following: `'min'`, `'max'`, `'avg'`, `'sum'`, `'calc'`, `'distinct_count'`, `'last_value'`, `'percentiles'`	`'sum'`

Array of group objects. Group objects specify which data is returned by a query and how it is grouped. Each group object has a name, limit, and a sort object.

fields: [{},{}]

Array of field objects. A field object is used to take a whole column of data without grouping it by the items found in the column. Each field object consists of the name of a column.

Note: If `fields` is defined, `groups` and `dimensions` must be omitted.

Key	Usage	Notes	Examples
`name`	String.	This name is that of the data source column containing the field.	`'home_state'`
limit	Integer.	The maximum number of records to report.	50

metrics: [{},{}]

Array of metric objects. Each metric object indicates a column to be returned and used by the data query as a metric. Each metric object can have a name and function.

Key	Usage	Notes	Examples
`name`	String. The name of the metric.	This name is that of the data source column containing the metric, rather than the label that appears in the user interface.	`'user_age'`
`function`	String. Determines the function applied to the metric.	Valid options include the following: `'min'`, `'max'`, `'avg'`, `'sum'`, `'calc'`, `'distinct_count'`, `'last_value'`, `'percentiles'`	`'avg'`

Security Configuration Object

The security configuration object contains the required access token you generate for your client to access Composer.

Note: insightsoftware recommends using Trusted Access for all embed-related workflows.

Example

```
            var credentialsConfig = {
access_token: TRUSTED_ACCESS_TOKEN,
};
```

The security configuration object should contain the required trusted access token you generate.

- **access_token**: Trusted access token for accessing the Composer server. See Generate A User's Access Token. The client created should have credentials matching those used to create the trusted access token.

Supplied Color Palettes

Composer comes with a number of color palettes. You can add a color palette to your environment using themes. See Manage UI Themes.

The following table lists the supplied color palettes that are available for use in visuals. If a palette name has `-accessible` appended to its name, it is colorblind friendly.

Palette Name	Hex Color Values
Accent	['#7fc97f', '#beaed4', '#fdc086', '#ffff99', '#386cb0', '#f0027f', '#bf5b17', '#666666']
Blues-accessible	['#f7fbff', '#deebf7', '#c6dbef', '#9ecae1', '#6baed6', '#4292c6', '#2171b5', '#08519c', '#08306b']
BrBG-accessible	['#543005', '#8c510a', '#bf812d', '#dfc27d', '#f6e8c3', '#f5f5f5', '#c7eae5', '#80cdc1', '#35978f', '#01665e', '#003c30']
BuGn-accessible	['#f7fcfd', '#e5f5f9', '#ccece6', '#99d8c9', '#66c2a4', '#41ae76', '#238b45', '#006d2c', '#00441b']
BuPu-accessible	['#f7fcfd', '#e0ecf4', '#bfd3e6', '#9ebcda', '#8c96c6', '#8c6bb1', '#88419d', '#810f7c', '#4d004b']
Dark2-accessible	['#1b9e77', '#d95f02', '#7570b3', '#e7298a', '#66a61e', '#e6ab02', '#a6761d', '#666666']
GnBu-accessible	['#f7fcf0', '#e0f3db', '#ccebc5', '#a8ddb5', '#7bccc4', '#4eb3d3', '#2b8cbe', '#0868ac', '#084081']
Greens-accessible	['#f7fcf5', '#e5f5e0', '#c7e9c0', '#a1d99b', '#74c476', '#41ab5d', '#238b45', '#006d2c', '#00441b']
Greys-accessible	['#ffffff', '#f0f0f0', '#d9d9d9', '#bdbdbd', '#969696', '#737373', '#525252', '#252525', '#000000']
Oranges-accessible	['#fff5eb', '#fee6ce', '#fdd0a2', '#fdae6b', '#fd8d3c', '#f16913', '#d94801', '#a63603', '#7f2704']
OrRd-accessible	['#fff7ec', '#fee8c8', '#fdd49e', '#fdbb84', '#fc8d59', '#ef6548', '#d7301f', '#b30000', '#7f0000']
Paired-accessible	['#a6cee3', '#1f78b4', '#b2df8a', '#33a02c', '#fb9a99', '#e31a1c', '#fdbf6f', '#ff7f00', '#cab2d6', '#6a3d9a', '#ffff99', '#b15928']
Pastel1	['#fbb4ae', '#b3cde3', '#ccebc5', '#decbe4', '#fed9a6', '#ffffcc', '#e5d8bd', '#fddaec', '#f2f2f2']
Pastel2	['#b3e2cd', '#fdcdac', '#cbd5e8', '#f4cae4', '#e6f5c9', '#fff2ae', '#f1e2cc', '#cccccc']
PiYG-accessible	['#8e0152', '#c51b7d', '#de77ae', '#f1b6da', '#fde0ef', '#f7f7f7', '#e6f5d0', '#b8e186', '#7fbc41', '#4d9221', '#276419']
PRGn-accessible	['#40004b', '#762a83', '#9970ab', '#c2a5cf', '#e7d4e8', '#f7f7f7', '#d9f0d3', '#a6dba0', '#5aae61', '#1b7837', '#00441b']
PuBu-accessible	['#fff7fb', '#ece7f2', '#d0d1e6', '#a6bddb', '#74a9cf', '#3690c0', '#0570b0', '#045a8d', '#023858']
PuBuGn-accessible	['#fff7fb', '#ece2f0', '#d0d1e6', '#a6bddb', '#67a9cf', '#3690c0', '#02818a', '#016c59', '#014636']
PuOr-accessible	['#7f3b08', '#b35806', '#e08214', '#fdb863', '#fee0b6', '#f7f7f7', '#d8daeb', '#b2abd2', '#8073ac', '#542788', '#2d004b']
PuRd-accessible	['#f7f4f9', '#e7e1ef', '#d4b9da', '#c994c7', '#df65b0', '#e7298a', '#ce1256', '#980043', '#67001f']
Purples-accessible	['#fcfbfd', '#efedf5', '#dadaeb', '#bcbddc', '#9e9ac8', '#807dba', '#6a51a3', '#54278f', '#3f007d']
RdBu-accessible	['#67001f', '#b2182b', '#d6604d', '#f4a582', '#fddbc7', '#f7f7f7', '#d1e5f0', '#92c5de', '#4393c3', '#2166ac', '#053061']
RdGy	['#67001f', '#b2182b', '#d6604d', '#f4a582', '#fddbc7', '#ffffff', '#e0e0e0', '#bababa', '#878787', '#4d4d4d', '#1a1a1a']

Palette Name	Hex Color Values
RdOrYl-accessible	['#eea8b6', '#de536f', '#bd2525', '#772b28', '#e44838', '#ee5502', '#ffbf1f', '#ffdc85']
RdPu-accessible	['#fff7f3', '#fde0dd', '#fcc5c0', '#fa9fb5', '#f768a1', '#dd3497', '#ae017e', '#7a0177', '#49006a']
RdYlBu-accessible	['#a50026', '#d73027', '#f46d43', '#fdae61', '#fee090', '#ffffbf', '#e0f3f8', '#abd9e9', '#74add1', '#4575b4', '#313695']
RdYlGn	['#a50026', '#d73027', '#f46d43', '#fdae61', '#fee08b', '#ffffbf', '#d9ef8b', '#a6d96a', '#66bd63', '#1a9850', '#006837']
Reds-accessible	['#fff5f0', '#fee0d2', '#fcbba1', '#fc9272', '#fb6a4a', '#ef3b2c', '#cb181d', '#a50f15', '#67000d']
Set2-accessible	['#66c2a5', '#fc8d62', '#8da0cb', '#e78ac3', '#a6d854', '#ffd92f', '#e5c494', '#b3b3b3']
Set1	['#e41a1c', '#377eb8', '#4daf4a', '#984ea3', '#ff7f00', '#ffff33', '#a65628', '#f781bf', '#999999']
Set3	['#8dd3c7', '#ffffb3', '#bebada', '#fb8072', '#80b1d3', '#fdb462', '#b3de69', '#fccde5', '#d9d9d9', '#bc80bd', '#ccebc5', '#ffed6f']
Spectral	['#9e0142', '#d53e4f', '#f46d43', '#fdae61', '#fee08b', '#ffffbf', '#e6f598', '#abdda4', '#66c2a5', '#3288bd', '#5e4fa2']
YlGn-accessible	['#ffffe5', '#f7fcb9', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#006837', '#004529']
YlGnBu-accessible	['#ffffd9', '#edf8b1', '#c7e9b4', '#7fcdbb', '#41b6c4', '#1d91c0', '#225ea8', '#253494', '#081d58']
YlOrBr-accessible	['#ffffe5', '#fff7bc', '#fee391', '#fec44f', '#fe9929', '#ec7014', '#cc4c02', '#993404', '#662506']
YlOrRd-accessible	['#ffffcc', '#ffeda0', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#bd0026', '#800026']
DefaultQualitative	['#0095b7', '#a0b774', '#f4c658', '#fe8b3e', '#cf2f23', '#756c56', '#007896', '#47a694']
DefaultSequential-accessible	['#ffdc9c', '#ffc65f', '#efc15e', '#9eb778', '#7eb184', '#43a79b', '#008db6', '#097bb1']

Operators

Valid operators used in Composer are described below.

Operator	Applies to	Determines whether the data in the field you select...
Begins With	strings	Begins with the string you have specified.
Between	numbers	Falls numerically between the two numbers you specified.
Contains	strings	Contains the string you have specified.
Does Not Begin With	strings	Does ***not*** begin with the string you have specified.
Does Not Contain	strings	Does ***not*** contain the string you have specified.
Does Not End With	strings	Does ***not*** end with the string you have specified.
Ends With	strings	Ends with the string you have specified.
Equal	numbers	Is equal to the number you specified.
Exclude	numbers & strings	Includes the number or string you specify. If it does, the data is excluded from the output.
Greater Than	numbers	Is greater than the number you specified.
Greater Than or Equal	numbers	Is greater than or equal to the number you specified.
Include	numbers & strings	Includes the number or string you specify. If it does, the data is included in the output.
Less Than	numbers	Is less than the number you specified.
Less Than or Equal	numbers	Is less than or equal to the number you specified.
Not Between	numbers	Does ***not*** fall numerically between the two numbers you specified.
Not Equal	numbers	Is ***not*** equal to the number you specified.

State Name Reference

State names must be spelled out with the first letter capitalized. For example, **California** is an accepted format, not **CA**. The following state name formats are accepted for US Region maps.

State Name
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
District of Columbia
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi

State Name
Missouri
Montana
Nebraska
Nevada
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Ohio
Oklahoma
Oregon
Pennsylvania
Puerto Rico
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virginia
Washington
West Virginia
Wisconsin
Wyoming

Country Name Reference

Country names must be spelled out with the first letter capitalized. The following country name formats are accepted for world maps.

Country Name	Country Name	Country Name	Country Name
Afghanistan	Egypt	Liberia	Saudi Arabia
Albania	El Salvador	Libya	Senegal
Algeria	Equatorial Guinea	Lithuania	Serbia
Angola	Eritrea	Luxembourg	Sierra Leone
Argentina	Estonia	Macedonia	Slovakia
Armenia	Ethiopia	Madagascar	Slovenia
Australia	Falkland Islands	Malawi	Solomon Islands
Austria	Fiji	Malaysia	Somalia
Azerbaijan	Finland	Mali	Somaliland
Bahamas	France	Mauritania	South Africa
Bangladesh	French Southern and Antarctic Lands	Mexico	South Sudan
Belarus	Gabon	Moldova	Spain
Belgium	Georgia	Mongolia	Sri Lanka
Belize	Germany	Montenegro	Sudan
Benin	Ghana	Morocco	Suriname
Bhutan	Greece	Mozambique	Swaziland
Bolivia	Greenland	Myanmar	Sweden
Bosnia and Herzegovina	Guatemala	Namibia	Switzerland
Botswana	Guinea	Nepal	Syria
Brazil	Guinea-Bissau	Netherlands	Taiwan
Brunei Darussalam	Guyana	New Caledonia	Tajikistan
Bulgaria	Haiti	New Zealand	Tanzania
Burkina Faso	Honduras	Nicaragua	Thailand
Burundi	Hungary	Niger	The Gambia
Cambodia	Iceland	Nigeria	Timor-Leste
Cameroon	India	Northern Cyprus	Togo

Country Name	Country Name	Country Name	Country Name
Canada	Indonesia	Norway	Trinidad and Tobago
Central African Republic	Iran	Oman	Tunisia
Chad	Iraq	Pakistan	Turkey
Chile	Ireland	Palestine	Turkmenistan
China	Israel	Panama	Uganda
Colombia	Italy	Papua New Guinea	Ukraine
Costa Rica	Jamaica	Paraguay	United Arab Emirates
Côte d'Ivoire	Japan	Peru	United Kingdom
Croatia	Jordan	Philippines	United States
Cuba	Kazakhstan	Poland	Uruguay
Cyprus	Kenya	Portugal	Uzbekistan
Czech Republic	Kosovo	Puerto Rico	Vanuatu
Dem. Rep. Korea	Kuwait	Qatar	Venezuela
Democratic Republic of the Congo	Kyrgyzstan	Republic of Congo	Vietnam
Denmark	Lao PDR	Republic of Korea	Western Sahara
Djibouti	Latvia	Romania	Yemen
Dominican Republic	Lebanon	Russian Federation	Zambia
Ecuador	Lesotho	Rwanda	Zimbabwe

Preset Time Ranges

The following table describes all the preset time ranges available in Composer.

Note: If you are using a field that has time zone information disabled (select **Not Specified**), only the time-related information is shown in the user interface and exported with your data. Time zone labels are not included.

Preset type	Data Cached?	Start date	End date
Current Hour	No	Beginning of the current hour	Current time
Current Minute	No	Beginning of the current minute	Current time
Current Month	No	First day of the current month (for example 08/01/2024 12:00:00 AM)	Last day of the current month (for example 08/31/2024 11:59:59 PM)
Current Month-to-Date	No	First day of the current month (for example 08/01/202412:00:00 AM)	Current date and time
Current Quarter	No	First day of the current quarter (for example, 07/01/202412:00:00 AM)	Last day of the current quarter (for example 09/30/202411:59:59 PM)
Current Quarter-to-Date	No	First day of the current quarter (for example 07/01/202412:00:00 AM)	Current date and time
Current Week	No	Sunday of the week with current day (for example Sunday, 08/11/202412:00:00 AM)	Saturday of the week with current day (for example 08/17/202411:59:59 PM)
Current Week-to-Date	No	Sunday of the week with current day (for example Sunday, 08/11/202412:00:00 AM)	Current date and time
Current Year	No	First day of the current year (for example 01/01/202412:00:00 AM)	Last day of the current year (for example 12/31/202411:59:59 PM)
Current Year-to-Date	No	First day of the current year (for example 01/01/202412:00:00 AM)	Current date and time
Max Available Range	No	Minimum time value in the data set known to Composer	Maximum time value in the data set known to Composer
Previous Hour	No	Beginning of the previous hour (for example Aug 20 2024 12:00:00 PM)	End of the previous hour (for example Aug 20 2024 12:59:59 PM)
Previous Minute	No	Beginning of the previous minute	End of the previous minute
Previous Month	Yes	First day of the previous month (for example 07/01/202412:00:00 AM)	Last day of the previous month (for example 07/31/2024 11:59:59 PM)

Preset type	Data Cached?	Start date	End date
Previous Month-to-Date	No	First day of the previous month (for example 07/01/2024 12:00:00 AM)	The current date. For example, if today is 08/05/2024 12:00:00 AM, the information returned is from 07/01/2022 12:00:00 AM through 08/05/2024 12:00:00 AM.
Previous Quarter	Yes	First day of the previous quarter (for example 04/01/2024 12:00:00 AM)	Last day of the previous quarter (for example, 06/30/2024 11:59:59 PM)
Previous Quarter-to-Date	No	First day of the previous quarter (for example 04/01/202412:00:00 AM)	The current date. For example, if today is 08/05/2024 12:00:00 AM, the information returned is from 04/01/2022 12:00:00 AM through 08/05/2024 12:00:00 AM.
Previous Week	Yes	Sunday of the previous week (for example, 08/04/2024 12:00:00 AM)	Saturday of the previous week (for example 08/10/2024 11:59:59 PM)
Previous Week-to-Date	No	Sunday or Monday of the previous week, depending on locale (for example 07/27/2024 12:00:00 AM)	The current date. For example, if today is 08/02/2024 12:00:00 AM, the information returned is from 07/25/2022 12:00:00 AM through 08/02/2024 12:00:00 AM.
Previous Year	Yes	First day of the previous year (for example 01/01/2023 12:00:00 AM)	Last day of the previous year (for example, 12/31/2023 11:59:59 PM)
Previous Year-to-Date	No	First day of the previous year (for example 01/01/2023 12:00:00 AM)	The current date. For example, if today is 08/05/202412:00:00 AM, the information returned is from 01/01/2023 12:00:00 AM through 08/05/2024 12:00:00 AM.
Rolling 7 days	No	7 days before the current date and time	Current date and time
Rolling 24 hours	No	24 hours before the current date and time	Current date and time
Rolling 30 days	No	30 days before the current date and time	Current date and time
Rolling 90 days	No	90 days before the current date and time	Current date and time
Rolling 365 days	No	365 days before the current date and time	Current date and time
Rolling Hour	No	60 minutes before the current date and time	Current date and time
Rolling Minute	No	One minute before the current date and time	Current date and time
Today	No	Start of the current day (for example, 08/20/2024 12:00:00 AM)	Current date and time
Yesterday	Yes	Beginning of the previous day (for example,	End of the previous day (08/19/2024 11:59:59

Preset type	Data Cached?	Start date	End date
		08/19/2024 12:00:00 AM)	PM)

Errors and Exceptions

When working with the SDK, the Composer client might return one or more error messages. The following tables describe all the error messages the SDK might return, and some possible solutions.

createQuery

Error	Description	Solution
metrics must be an array	The metrics attribute of the query configuration object is not an array.	Make sure metrics is an array, even if it's just an array of one.
invalid metrics	Metrics function, for example "avg" or "min", is not one of the expected values.	The metrics function must be one of the following: `'min'` , `'max'` , `'avg'` , `'sum'` , `'calc'` , `'distinct_count'` , `'last_value'` , `'percentiles'`
groups is not defined fields is not defined	The query configuration object has neither a groups attribute or a fields attribute.	Either groups or fields must be defined. See the query configuration page for more information about the query configuration object.
groups is not defined fields must be an array of field names	The query configuration object has a fields attribute that is not an array of field names, and doesn't define a groups attribute.	Make sure your fields attribute is an array, or add a groups attribute to the query configuration object.
groups must be an array of groups configurations fields is not defined	The query configuration object has a groups attribute that isn't an array of groups configurations, and doesn't have a fields attribute.	Make sure your groups attribute is an array. If you don't want a grouped query, add a fields attribute with an array of field names.
filters must be an array	The query configuration object has a filters attribute that isn't an array of filter configurations.	Make sure your filters attribute is an array, even if it's an empty array `[]` or an array of just one configuration.
element expected	Configuration passed to visualize() does not have an 'element' attribute that is a webpage element.	Check out the XYZ link to visualize sample.
visualization doesn't exist	Using visualize(), you tried to create a visualization that doesn't exist.	Check the name of your visualization and make sure it is valid. If using a custom chart, make sure you copied the name exactly.
data set empty	The data set that you are trying to access is empty.	This message will result if you have a non-live (static) data source and the query to access it contains `pauseAfterRead` set to `false` . A query to static data sources should have `pauseAfterRead` set to true, otherwise the query attempts to read and reread the source.

Events Created by Client

Thread Events

Event	Description
thread:start	Thread is starting.
thread:stop	Thread is stopping.
thread:pause	Thread is pausing. Relevant only for time bar visuals.
thread:unpause	Thread is resuming. Relevant only for time bar visuals.
thread:updateSpeed	Thread's speed is updating. Relevant only for time bar visuals.
thread:message	Thread is sending data, which is returned as a packet by the listener to a callback function.
thread:error	An error occurred on the thread. The error message is included in the event, which is returned as a packet by the listener to a callback function.
thread:startVisDone	Thread has finished starting the visual.
thread:stopVisDone	Thread has finished stopping the visual.
thread:noData	Thread has found no data.
thread:dirtyData	Data is currently being read. Not fully sharpened yet.
thread:notDirtyData	All the data has been read and is fully sharpened.
thread:pauseDone	Thread has finished pausing. For time bar sources.
thread:unpauseDone	Thread has finished resuming. For time bar sources.
thread:pagination	Pagination for most visuals.
thread:pivotPaging	Separate paging event for Pivot Table visual.
thread:viewport	Viewport has updated.
thread:timeline	Reports time window currently being queried.

Visualization Events

Event	Description
showReflinePopup	Show Reference Line pop-up.
interaction	Interaction with the visual. Includes event details.

Event	Description
actionList:open	Action list opened.
actionList:close	Action list closed.
tooltip:show	Display tooltip.
tooltip:update	Updating tooltip.
tooltip:move	Moving tooltip.
tooltip:hide	Hide tooltip.
filters:add	Add a filter.
filters:remove	Remove a filter.
showWarning	Show a warning.
releaseInteractiveElement	Released an interactive element you no longer want to be interactive.
registerInteractiveElement	Registered a new interactive element.
requestData	Requesting data.
axisLabels:create	Create axis labels for your visual.
axisLabel:create	Create an axis label for your visual.
axisLabels:remove	Remove all axis labels from visual.
stop	Visual is stopping.
request:pickAttribute	Visual has requested attribute picker.
request:pickMetric	Visual has requested metric picker.
request:registered	Visual queries whether some element is interactive.
request:getTicksStep	Visual is requesting the tick step.

Cautionary Note About Internal APIs

Composer offers a number of REST APIs so you can extend and embed Composer in your own custom applications. These APIs are denoted as publicly available using the `/api/` path (for example, `www.yourcompany.com/composer/api/accounts`).