System Event Audit Messages

On this page

NOTE:
Available only in MongoDB Enterprise.

Audit Message

The event auditing feature can record events in JSON format. To configure auditing output, see Configure Auditing

The recorded JSON messages have the following syntax:

```
{
  atype: <String>,
  ts : { "$date": <timestamp> },
  local: { ip: <String>, port: <int> },
  remote: { ip: <String>, port: <int> },
  users : [ { user: <String>, db: <String> }, ... ],
  roles: [ { role: <String>, db: <String> }, ... ],
  param: <document>,
  result: <int>
}
```

Field	Type	Description
`atype`	string	Action type. See Audit Event Actions, Details, and Results.
`ts`	document	Document that contains the date and UTC time of the event, in ISO 8601 format.

Field	Type	Description
`local`	document	Document that contains the local `ip` address and the `port` number of the running instance.
`remote`	document	Document that contains the remote `ip` address and the `port` number of the incoming connection associated with the event.
`users`	array	Array of user identification documents. Because MongoDB allows a session to log in with different user per database, this array can have more than one user. Each document contains a `user` field for the username and a `db` field for the authentication database for that user.
`roles`	array	Array of documents that specify the roles granted to the user. Each document contains a `role` field for the name of the role and a `db` field for the database associated with the role.
`param`	document	Specific details for the event. See Audit Event Actions, Details, and Results.
`result`	integer	Error code. See Audit Event Actions, Details, and Results.

Audit Event Actions, Details, and Results

The following table lists for each `atype` or action type, the associated `param` details and the `result` values, if any.

atype	**param**	**result**
`authenticate`	`{ user: <user name>, db: <database>, mechanism: <mechanism> }`	`0` - Success `18` - Authentication Failed

atype	param	result
authCheck	{ command: <name>, ns: <database>.<collection>, args: <command object> } ns field is optional. args field may be redacted.	0 - Success 13 - Unauthorized to perform the operation. By default, the auditing system logs only the authorization failures. To enable the system to log authorization successes, use the auditAuthorizationSuccess parameter. [1]
createCollection	{ ns: <database>.<collection> }	0 - Success
createDatabase	{ ns: <database> }	0 - Success
createIndex	{ ns: <database>.<collection>, indexName: <index name>, indexSpec: <index specification> }	0 - Success
renameCollection	{ old: <database>.<collection>, new: <database>.<collection> }	0 - Success
dropCollection	{ ns: <database>.<collection> }	0 - Success
dropDatabase	{ ns: <database> }	0 - Success
dropIndex	{ ns: <database>.<collection>, indexName: <index name> }	0 - Success

atype	param	result
createUser	{ user: <user name>, db: <database>, customData: <document>, roles: [{ role: <role name>, db: <database> }, ...] } The customData field is optional.	0 - Success
dropUser	{ user: <user name>, db: <database> }	0 - Success
dropAllUsersFromDatabase	{ db: <database> }	0 - Success
updateUser	{ user: <user name>, db: <database>, passwordChanged: <boolean>, customData: <document>, roles: [{ role: <role name>, db: <database> }, ...] } The customData field is optional.	0 - Success

atype	param	result
grantRolesToUser	{ user: <user name>, db: <database>, roles: [{ role: <role name>, db: <database> }, ...] }	0 - Success
revokeRolesFromUser	{ user: <user name>, db: <database>, roles: [{ role: <role name>, db: <database> }, ...] }	0 - Success

atype	param	result
createRole	{ role: <role name>, db: <database>, roles: [{ role: <role name>, db: <database> }, ...], privileges: [{ resource: <resource document>, actions: [<action>, ...] }, ...] } The roles and the privileges fields are optional. For details on the resource document, see Resource Document. For a list of actions, see Privilege Actions.	0 - Success

atype	param	result
updateRole	{ role: <role name>, db: <database>, roles: [{ role: <role name>, db: <database> }, ...], privileges: [{ resource: <resource document>, actions: [<action>, ...] }, ...] } The roles and the privileges fields are optional. For details on the resource document, see Resource Document. For a list of actions, see Privilege Actions.	0 - Success
dropRole	{ role: <role name>, db: <database> }	0 - Success
dropAllRolesFromDatabase	{ db: <database> }	0 - Success

atype	param	result
grantRolesToRole	{ role: <role name>, db: <database>, roles: [{ role: <role name>, db: <database> }, ...] }	0 - Success
revokeRolesFromRole	{ role: <role name>, db: <database>, roles: [{ role: <role name>, db: <database> }, ...] }	0 - Success
grantPrivilegesToRole	{ role: <role name>, db: <database>, privileges: [{ resource: <resource document>, actions: [<action>, ...] }, ...] } For details on the resource document, see Resource Document. For a list of actions, see Privilege Actions.	0 - Success

atype	param	result
revokePrivilegesFromRole	{ role: <role name>, db: <database name>, privileges: [{ resource: <resource document>, actions: [<action>, ...] }, ...] } For details on the resource document, see Resource Document. For a list of actions, see Privilege Actions.	0 - Success
enableSharding	{ ns: <database> }	0 - Success
shardCollection	{ ns: <database>.<collection>, key: <shard key pattern>, options: { unique: <boolean> } }	0 - Success
addShard	{ shard: <shard name>, connectionString: <hostname>:<port>, maxSize: <maxSize> } When a shard is a replica set, the connectionString includes the replica set name and can include other members of the replica set.	0 - Success
removeShard	{ shard: <shard name> }	0 - Success
shutdown	{ } Indicates commencement of database shutdown.	0 - Success

atype	param	result
applicationMessage	{ msg: <custom message string> } See logApplicationMessage.	0 - Success

[1] Enabling auditAuthorizationSuccess degrades performance more than logging only the authorization failures.