ArangoDB CHEAT SHEET

Starting & Accessing

./arangod /path/to/my/db	start server
http://localhost:8529/_admin	admin front end
./arangosh	start ArangoDB shell
start_pretty_print();	format output nicely
arangod --console --log.level error /path/to/my/db	start emergency console

arangod frequently used starting options

--log.file path/to/file	set log file or + for stdout
--log.level	set log level: fatal, error, warning, info, debug, trace
--server.endpoint protocol://address:port	set address and port for Clients

Database management methods

db._createDatabase(*database-name*)	create database
db._dropDatabase(*database-name*)	drop a database
db._useDatabase(*database-name*)	change into an existing database
db._databases()	list all databases

Collection management methods

db._create(*collection-name*, *properties*)	create collection (with optional *properties*)
db._createEdgeCollection(*collection-name*, *properties*)	create an edge collection
db._collection(*collection-name*|*collection-id*)	get collection
db._collections()	List all collections
db.*collection-name*	get a collection by name
db._drop(*collection-name*|*collection-id*)	drop collection with indexes
db._truncate(*collection-name*|*collection-id)*	remove collection, keep indexes

Collection methods

collection.drop()	drop collection with indexes
collection.truncate()	remove documents, keep indexes
collection.properties()	get all document properties
collection.properties(newProperties)	set all newProperties
collection.figures()	get all collection figures
collection.load()	load collection into memory
collection.unload()	start to unload a collection
collection.rename(new-name)	rename collection to new-name

Document methods

collection.document(*document*)	get document by identifier
collection.save(*data*)	create new document
collection.replace(*document*, *data*)	replace existing document
collection.update(*document*)	partially update
collection.remove(*document*)	remove document
db._document(*document*|*document-handle*)	get document by identifier|handle
db._replace(*document*|*document-handle*,*data*)	replace existing document
db._update(document)	partially update document
db._remove(*document*)	remove document

Edges

edge-collection.save(document)	save new edge
edge-collection.edges(vertex)	find edges connected to vertex
edge-collection.inEdges(vertices)	find all edges ending in vertex (inbound)
edge-collection.outEdges(vertices)	find all edges starting at vertex (outbound)

Collection methods

collection.all()	select all documents and return cursor
collection.any()	select a random document
collection.byExample(*example*)	select all documents that match *example*
collection.firstExample(*example*)	select first document that matches *example*
collection.removeByExample(*example*)	remove all documents that match *example*
collection.removeByExample (*example*, *newValue*)	replace all documents that match *example*
collection.updateByExample (*example*, *newValue*)	update all documents that match *example*
collection.count()	return the number of documents
collection.toArray()	convert the collection into an array of documents

AQL queries

db._query(*query*).toArray()	executes AQL query
db._query(*query, bindParameters*).toArray()	executes AQL query with bind parameters
db._explain(*query, bindParameters*)	prints the execution plan of the query