MongoDB JSON Schema Validation – Cheat Sheet

Definition: MongoDB JSON Schema Validation is a feature that allows you to define rules (schema) for collections. It ensures data structure, field types, and constraints are followed before inserting or updating documents.

■ Example Schema Validation

db.createCollection("users", { validator: { $jsonSchema: { bsonType: "object", required: ["name", "email", "age"], properties: { name: { bsonType: "string", description: "Name must be a string" }, email: { bsonType: "string", pattern: "^.+@.+\\..+$", description: "Valid email required" }, age: { bsonType: "int", minimum: 18, maximum: 100, description: "Age must be 18-100" }, isActive: { bsonType: "bool", description: "Boolean field" } } } }, validationLevel: "strict", validationAction: "error" })

■■ Common JSON Schema Keywords

Keyword	Purpose	Example
bsonType	Specifies MongoDB data type	"bsonType": "string"
required	Defines required fields	["name", "email"]
properties	Describes each field's validation	"properties": { name: {...} }
pattern	Regex for string validation	"pattern": "^.+@.+\\..+$"
minimum / maximum	Numeric range validation	"minimum": 18
enum	Allowed fixed values	"enum": ["admin", "user"]
validationLevel	When validation applies	"strict" or "moderate"
validationAction	What happens on error	"error" or "warn"

■ Advantages

Ensures Data Integrity: Prevents invalid or inconsistent data. **Reduces Errors:** Early detection before insertion. **Application Consistency:** Ensures backend and DB rules match. **Self-Documenting:** Schema itself describes data structure.

■ Validation Modes

Setting	Options	Description
validationLevel	strict / moderate	Strict = validate all; Moderate = only new/updated docs
validationAction	error / warn	Error = block invalid docs; Warn = log warning only

■ Summary

Command Used: $jsonSchema **Applies To:** Collection-level validation **Used For:** Enforcing field types, required fields, and value rules **Recommended Mode:** validationLevel: "strict" + validationAction: "error"