NodeJS

Agenda

01 Events

02 Steps in Writing Event Driven Code

03 EventEmitter

04 Registering Callbacks

05 Emitting Events

06 Emit and Respond to Events

07 Benefits of Event Driven Programming

08 Streams

09 Types of Streams

10 Read Data from a file using streams

Agenda

11 Write Data to a file using streams

12 Stream Events

13 Process Stream Events

MongoDB

What is MongoDB?

What is ExpressJS?

MongoDB is Document Oriented, NoSql database management system

Since MongoDB is a NoSql Database, you do not need to provide a schema and can insert data of any shape in a document

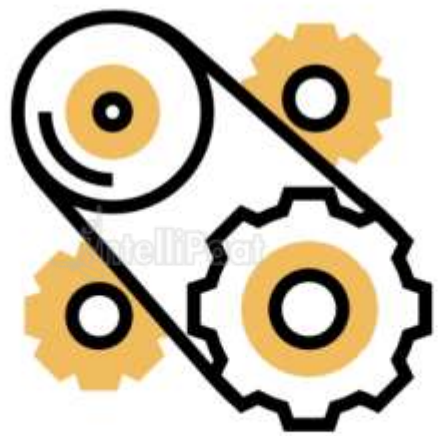
Why MongoDB?

Why MongoDB?

MongoDB provides several benefits

- Fast
- Flexible
- No Joins
- Ease of Use
- Scalable

MongoDB provides high speed reads and writes to collections and documents

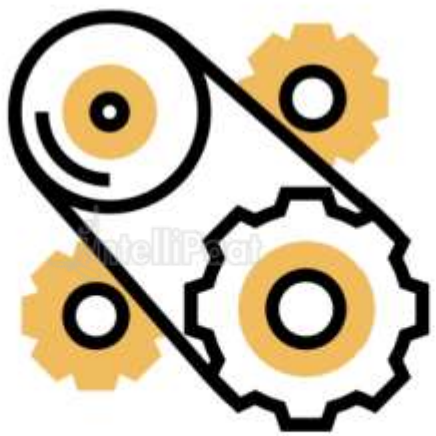


Why MongoDB?

MongoDB provides several benefits

- Fast
- Flexible
- No Joins
- Ease of Use
- Scalable

Since MongoDB is Schema Less it provides you with the flexibility of inserting a record with any shape

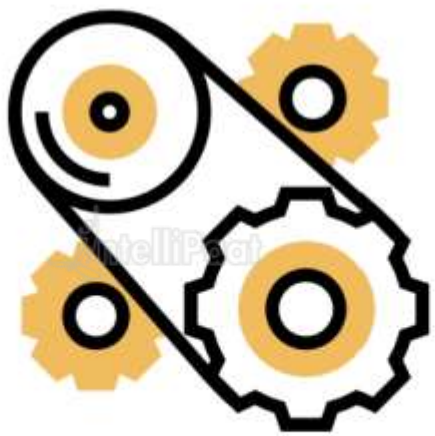


Why MongoDB?

MongoDB provides several benefits

- Fast
- Flexible
- No Joins
- Ease of Use
- Scalable

In MongoDB we can nest collections so we don't have to perform complex join queries to get results

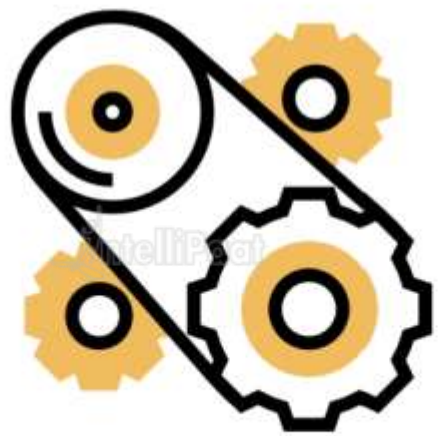


Why MongoDB?

MongoDB provides several benefits

- Fast
- Flexible
- No Joins
- Ease of Use
- Scalable

MongoDB uses query language that is similar to JavaScript and is very easy to understand and use

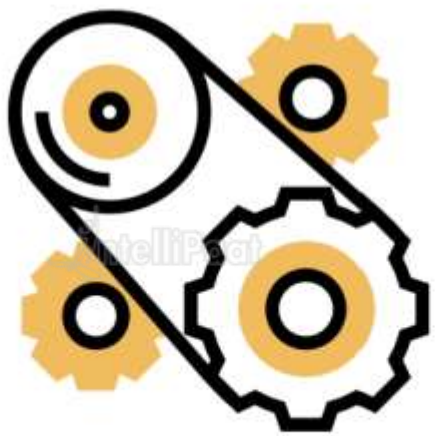


Why MongoDB?

MongoDB provides several benefits

- Fast
- Flexible
- No Joins
- Ease of Use
- Scalable

MongoDB is very easy to scale horizontally i.e. distributing data across a cluster of large number of server



Document

Document

A Document in MongoDB is a record inside a collection

Data is Stored inside these documents in a JSON like format called BSON or Binary JavaScript Object Notation

Collections

Collections

In MongoDB Database data is stored in documents and these documents are grouped inside a collection

You can think of a document as a row and a collection as a table in an SQL database

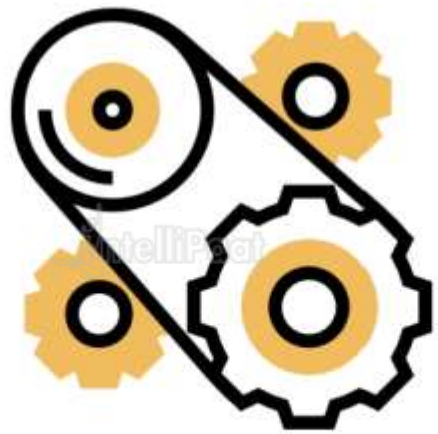
MongoDB Data Types

MongoDB Data Types

MongoDB supports all the data types that JSON supports with some additional data types, some of them are:

- Integer
- Double
- Timestamp
- Binary data
- Object ID
- Null

Integers are used to store whole number, depending on your server these can be 32 bit or 64 bit

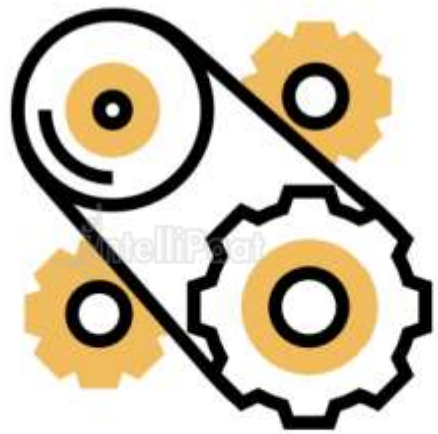


MongoDB Data Types

MongoDB supports all the data types that JSON supports with some additional data types, such as:

- Integer
- Double
- Timestamp
- Binary data
- Object ID
- Null

Doubles are used to store a floating point number

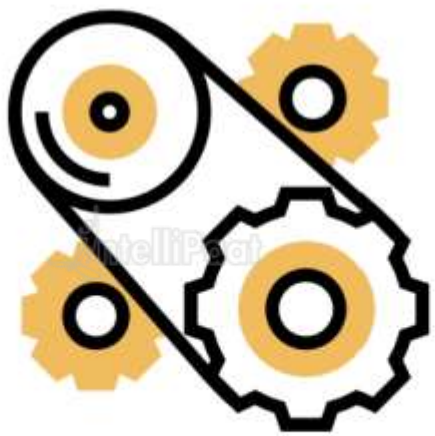


MongoDB Data Types

MongoDB supports all the data types that JSON supports with some additional data types, such as:

- Integer
- Double
- Timestamp
- Binary data
- Object ID
- Null

Timestamps are used to record when a document has been created or modified

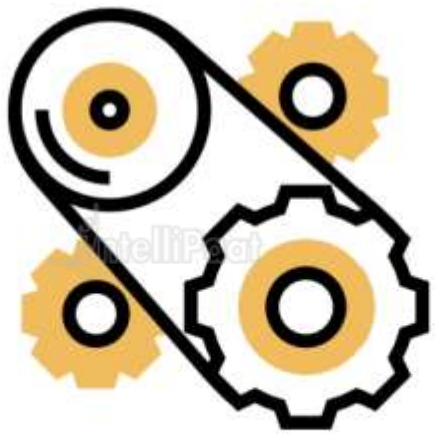


MongoDB Data Types

MongoDB supports all the data types that JSON supports with some additional data types, such as:

- Integer
- Double
- Timestamp
- Binary data
- Object ID
- Null

This is used to store binary data, such as images etc

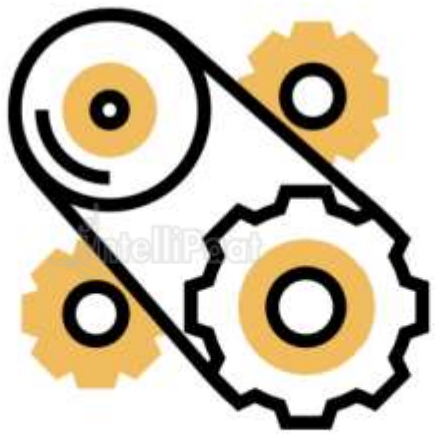


MongoDB Data Types

MongoDB supports all the data types that JSON supports with some additional data types, such as:

- Integer
- Double
- Timestamp
- Binary data
- Object ID
- Null

Object ID is a special type which is used to store the unique id of document

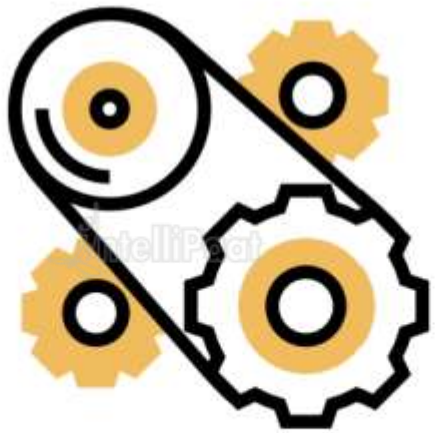


MongoDB Data Types

MongoDB supports all the data types that JSON supports with some additional data types, such as:

- Integer
- Double
- Timestamp
- Binary data
- Object ID
- Null

Null is data type which used to represent the absence of data



Using MongoDB with ExpressJS

Using MongoDB with ExpressJS

As ExpressJS is not opinionated, you can use MongoDB database with an quite easily

To use MongoDB with Express or any NodeJS application you can use one of many npm packages. Two of the most popular are the native MongoDB driver and the Mongoose package

Native MongoDB Driver

Native MongoDB Driver

The Native MongoDB driver is an NPM package that allows you to directly interact with your MongoDB server

It is created by the MongoDB team to allow applications to interact with the MongoDB server, very much like using the Mongo Shell

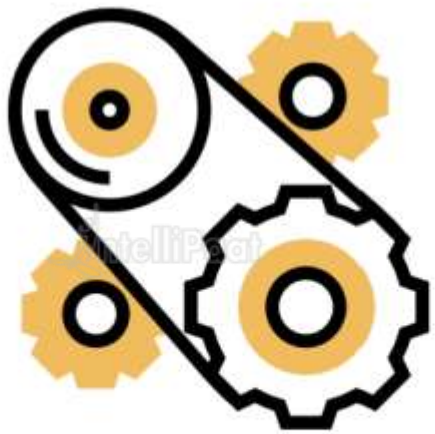
Why use Native MongoDB Driver?

Why use Native MongoDB Driver?

There are several benefits in using Native MongoDB Driver

- Fast
- Easy
- No Schema

As it is a very thin abstraction, it provides really good performance

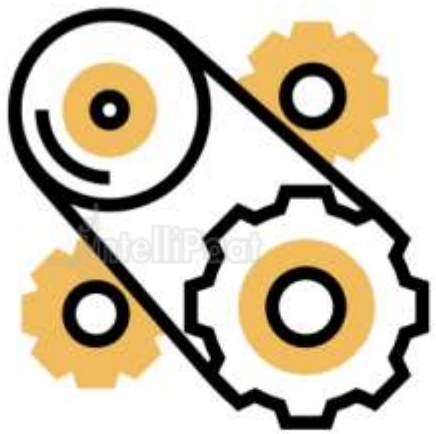


Why use Native MongoDB Driver?

There are several benefits in using Native MongoDB Driver

- Fast
- Easy
- No Schema

Using native driver is very easy as it make use of the same syntax that is used in the mongo shell

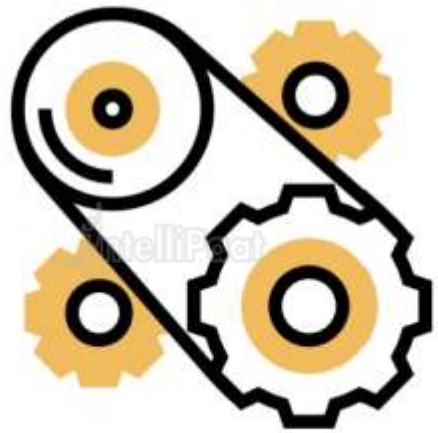


Why use Native MongoDB Driver?

There are several benefits in using Native MongoDB Driver

- Fast
- Easy
- No Schema

You can issue queries without using defining schema for your collections and documents



Mongoose

Mongoose

The Mongoose package is an ODM (Object Data Mapper)

It allows you to define schemas and models and instead of interacting with the database directly, you can just interact with the model instances and mongoose will issue the queries

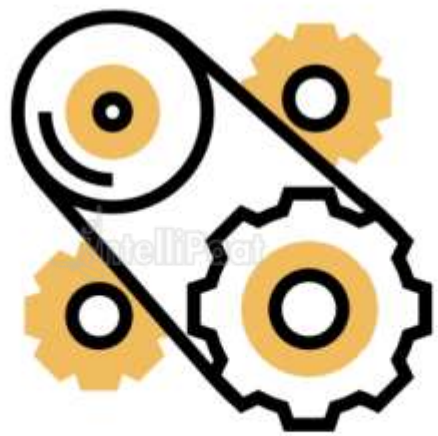
Why use Mongoose?

Why use Mongoose?

There are several benefits in using Native MongoDB Driver

- Speed
- Schemas
- Validation
- Utilities
- Reduced Effort

Even with all the additional benefits mongoose still provides very good data access speed

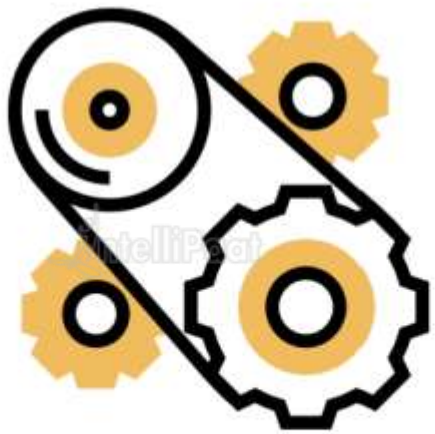


Why use Mongoose?

There are several benefits in using Native MongoDB Driver

- Speed
- Schemas
- Validation
- Utilities
- Reduced Effort

You can enforce strict schema validations to keep your data in a normalized state

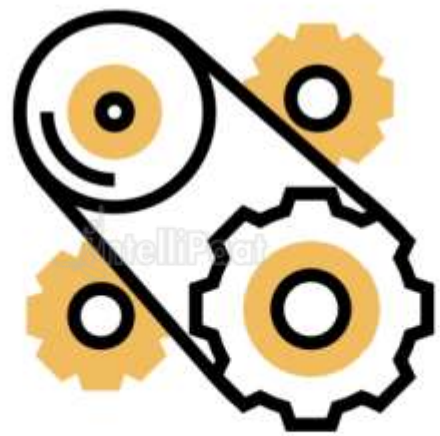


Why use Mongoose?

There are several benefits in using Native MongoDB Driver

- Speed
- Schemas
- Validation
- Utilities
- Reduced Effort

Mongoose takes validation of rules defined in schema definitions

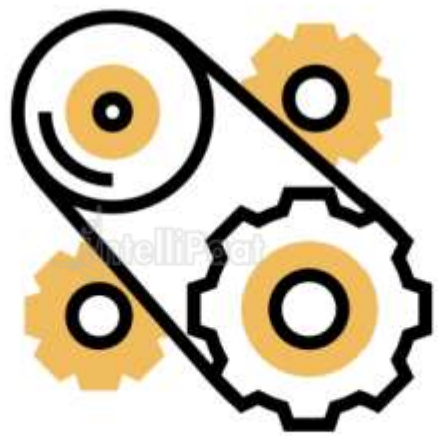


Why use Mongoose?

There are several benefits in using Native MongoDB Driver

- Speed
- Schemas
- Validation
- Utilities
- Reduced Effort

Mongoose has several built in methods that allow you to run custom code on successful query execution

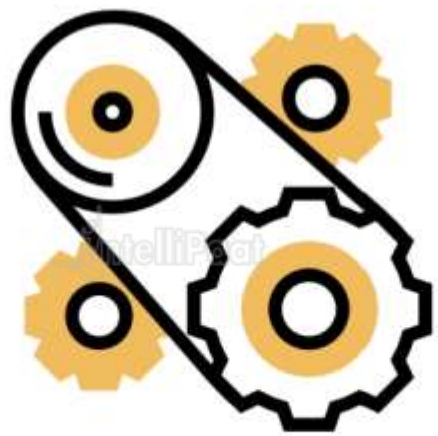


Why use Mongoose?

There are several benefits in using Native MongoDB Driver

- Speed
- Schemas
- Validation
- Utilities
- Reduced Effort

Mongoose greatly reduces the time it takes developer to build applications



Quiz

Quiz

1. Which method is used to read file synchronously?

A	readFileSynchronously
B	readFileSync
C	readFS
D	readFile

Quiz

1. Which method is used to read file synchronously?

A	readFileSynchronously
B	readFileSync
C	readFS
D	readFile

Quiz

2. Which method is used to read file asynchronously?

A	readFileAsynchronously
B	readFileAsync
C	readFAS
D	readFile

Quiz

2. Which method is used to read file asynchronously?

A	readFileAsynchronously
B	readFileAsync
C	readFAS
D	readFile

Quiz

3. Which constant is NodeJS is used to refer to absolute path of the current file?

A	__filename
B	__dirname
C	__absoule_file_path
D	__abs_filename

Quiz

3. Which constant is NodeJS is used to refer to absolute path of the current file?

A	__filename
B	__dirname
C	__absoule_file_path
D	__abs_filename

Quiz

4. Which constant is NodeJS is used to refer to absolute path of the current directory?

A	__filename
B	__dirname
C	__absoule_dir_path
D	__abs_directory

Quiz

4. Which constant is NodeJS is used to refer to absolute path of the current directory?

A	__filename
B	__dirname
C	__absoule_dir_path
D	__abs_directory

Quiz

5. Which property of the filestats object can be used to determine the time the file was last accessed?

A	access_timestamp
B	birthname
C	atime
D	uid

Quiz

5. Which property of the filestats object can be used to determine the time the file was last accessed?

A	access_timestamp
B	birthname
C	atime
D	uid

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor