

ITIS 5166

Network-Based Application Development

Mongoose

What is Mongoose?

- A popular Object Document Mapper for MongoDB
- Mongoose is built on top of the MongoDB native Node driver
- imposes a structure on documents through schema and schema validation
- maps MongoDB documents into objects in the application code
- performs type casting

Schema

- A Mongoose schema defines the format for all documents in a collection
- A schema defines properties that the documents can have. Each property may have
 - name, type, default values, validators
- Allowed Schema types: String, Number, Date, Buffer, Boolean, Mixed, ObjectId, Array

[Schema Documentation](#)

Example

- The Schema constructor takes two arguments
 - definition: object describing schema paths
 - options: object describing options

```
const userSchema = new Schema({  
  name: String,  
  age: Number  
});
```

Model

- A model is a compiled schema, which is used to create and access on documents stored in the corresponding collection
- Call `mongoose.model()` method to create a model
- The method takes two arguments
 - The first argument is the name of the model.
 - The second argument is the name of the schema that the model is based on.

Example

```
const userSchema = new Schema({  
  name: String,  
  age: Number  
});  
  
const User = mongoose.model('User', userSchema);
```

- Each model corresponds to a collection in the database, and the name of the collection is the plural, lowercased version of the model name
- The corresponding collection name in this example is users

[Model Documentation](#)

Constructing Documents

- A document is an instance of a model
- Create a document by passing a JavaScript object as an argument to the model
- Mongoose will
 - strip out any properties that are not defined in the schema
 - cast the values to match the types defined in the schema

Example

```
const userSchema = new Schema({
  name: String,
  age: Number
});

const User = mongoose.model('User', userSchema);

const user = new User({
  name: 'Billy Pilgrim',
  age: '64',
  rank: 'private'
});

console.log(user.name); // 'Billy Pilgrim'
console.log(user.age); // 64
console.log(user.rank); //undefined
//Mongoose strips out `rank` because it is not defined in the schema
```