**Common Mongoose Commands**

Here is a list and examples of some common Mongoose Commands you may need to use. These will be used in most Mongoose projects so make sure to practice them as much as you can.

*// Create a Schema for Users*

var UserSchema = new mongoose.Schema({

name: {type: String},

age: {type: Number}

}, {timestamps: true})

*// Store the Schema under the name 'User'*

mongoose.model('User', UserSchema);

*// Retrieve the Schema called 'User' and store it to the variable User*

var User = mongoose.model('User');

### Finding all users

*// Using the User Schema...*

*// ...retrieve all records matching {}*

User.find({}, function(err, users) {

*// Retrieve an array of users*

*// This code will run when the DB is done attempting to retrieve all matching records to {}*

})

### Finding all users based on a requirement

*// ...retrieve all records matching {name:'Jessica'}*

User.find({name:'Jessica'}, function(err, users) {

*// Retrieve an array of users matching the name. Even if 1 record is found, the result will be an array the size of 1, with 1 object inside. (Notice, if we are expecting to retrieve one record, we may want to use findOne and retrieve the object as oppose to an array the size of one.*

*// This code will run when the DB is done attempting to retrieve all matching records to {name:'Jessica'}*

})

### Finding one user

*// ...retrieve 1 record (the first record found) matching {}*

User.findOne({}, function(err, user) {

*// Retrieve 1 object*

*// This code will run when the DB is done attempting to retrieve 1 record.*

})

### Let's create a sample user

*// ...create a new instance of the User Schema and save it to the DB.*

var userInstance = new User()

userInstance.name = 'Andriana'

userInstance.age = 29

userInstance.save(function(err){

*// This code will run when Mongo has attempted to save the record.*

*// If (err) exists, the record was not saved, and (err) contains validation errors.*

*// If (err) does not exist (undefined), Mongo saved the record successfully.*

})

### Delete all users

*// ...delete all records of the User Model*

User.remove({}, function(err){

*// This code will run when the DB has attempted to remove all matching records to {}*

})

### Delete one user

*// ...delete 1 record by a certain key/vaue.*

User.remove({\_id: 'insert record unique id here'}, function(err){

*// This code will run when the DB has attempted to remove one matching record to {\_id: 'insert record unique id here'}*

})

### Update any records

*// ...update any records that match the query*

User.update({name:'Andrinnna'}, {name:'Andriana'}, function(err){

*// This code will run when the DB has attempted to update the matching record.*

})

*// another way to update a record*

User.find({name: 'Andriana'}, function(err, user){

user.name = 'Andri'

user.save(function(err){

*// if save was successful awesome!*

})

})