Node+Express+MongoDB

Review Learned

How to run server

What is nodemon

How to install require module

How to create mongodb server

Express Generator

What is Expressjs

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

What is Express Application Generator

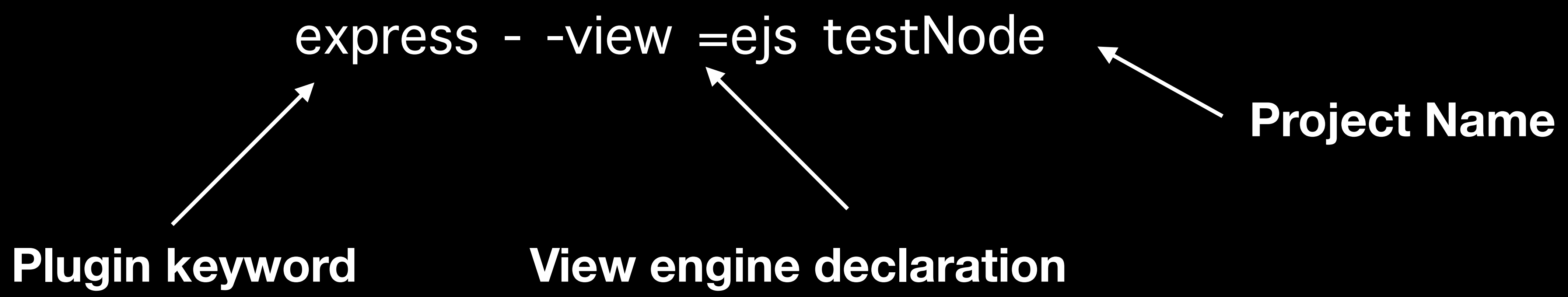
The *Express Application Generator* is not the only generator for Express applications, and the generated project is not the only viable way to structure your files and directories. The generated site does however have a modular structure that is easy to extend and understand.

How To Install

```
npm install express-generator -g
```

Express Generator

Generation Express Project



```
express - -view =ejs testNode
```

Plugin keyword

View engine declaration

Project Name

Install package & Run

Go to project path

```
cd projectName
```

Install package

```
npm install
```

Run project

```
npm start
```

Check the project

localhost:3000 at browser

Missions

- Change server local variable and check
- Add new view page
- Add new new route for view
- Add new local variable
- Add link to home page
- Add new view page name as “user-add.ejs”
- Also Add router get for this view
- Add a form at this view(name,email,password)
- Add new post router for this form
- Show result with res.end method
- Add nodemon package at package.json name as “dev” and check

Insert MongoDB in Express

Open App.js

Insert module

```
var mongoose = require(‘mongoose’);
```

Add mongodb connection

```
mongoose.connect('mongodb://127.0.0.1/librarydb');
var db = mongoose.connection;
db.on('error',console.error.bind(console,'MongoDB connection error:'))
```

Insert MongoDB in Express

Create model folder

Create User.js file under model folder and add following code

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;
 var UserSchema = new Schema({
                     name: {
                      type:String,
                      required: true
                         } ,
                     email: {
                      type:String,
                      required: true
                         } ,
                    password: {
                      type:String,
                      required: true
                          }
                                 });
  module.exports =mongoose.model('Users',UserSchema);
```

Insert MongoDB in Express

Open route file and add the module

```
var User = require(‘../models/User');
```

Can add, read, modify and Delete using Model instant

Add Data

```
var user = new User();
user.name = req.body.name;
user.email = req.body.email;
user.password= req.body.password;
user.save(function(err,rtn){
   if(err) throw err;
   console.log(rtn);
   res.redirect(‘/‘)
   });
```

Insert MongoDB in Express

Read Data

In route

```
User.find((err, rtn) => {
    if (err) throw err;
    res.render('user-list', {
      user: rtn
    });
  });
```

In view

Create user-list.ejs and table. In table body insert following code

```
<% for(var i = 0; i< user.length; i++){%>
   <tr>
     <td><%= user[i].name %></td>
     <td><%= user[i].email %></td>
     <td><%= user[i].password %></td>
   </tr>
 <%}%>
```

Insert MongoDB in Express

Update Data

```
router.get(“/update/:id“,function(req,res){

    User.findById(req.params.id,function(err,rtn){
      if(err) throw err;
      res.render(“user-update“,{user:rtn});
      });
  });

 router.post(“/update“,function(req,res){
   var update = {
     name: req.body.name,
     email: req.body.email,
     password: req.body.password
     }
     User.findByIdAndUpdate(req.body.id,{$set: update}, function(err,rtn){
       if(err) throw err;
       res.redirect(“/“);
       });
   });
```

Insert MongoDB in Express

Delete Data

```
router.get('/delete/:id', function(req, res) {
  User.findByIdAndRemove(req.params.id, function(err, rtn){
    if (err) throw err;
    console.log('rtn');
    res.redirect('/list');
  })
});
```