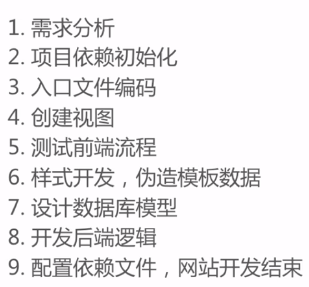
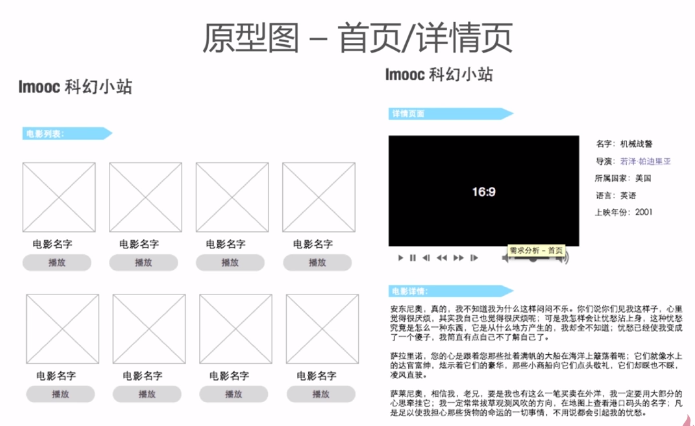
Nodsjs+mangodb建站攻略（一期）

开发步骤



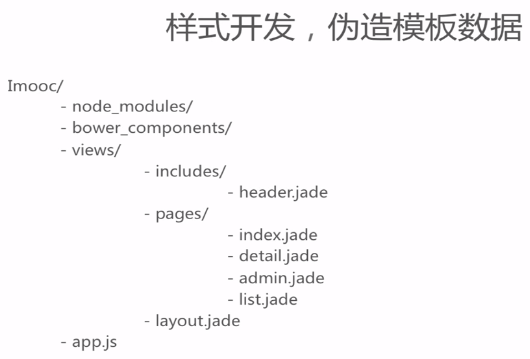




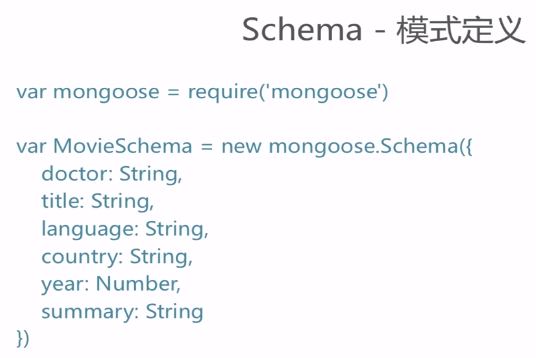


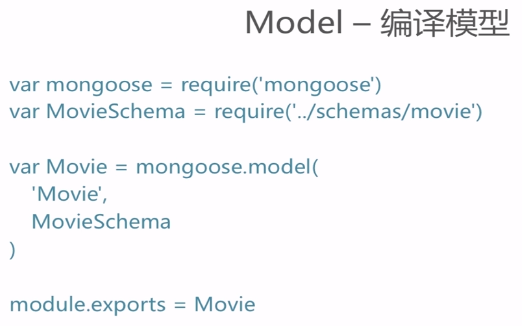


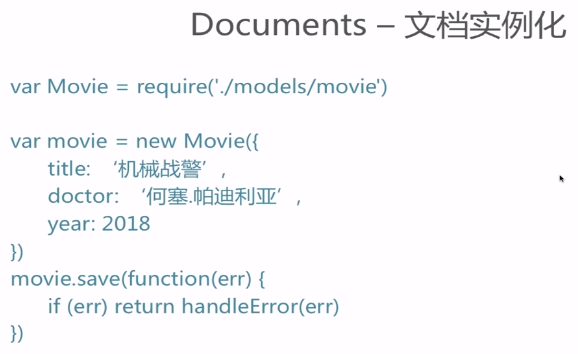


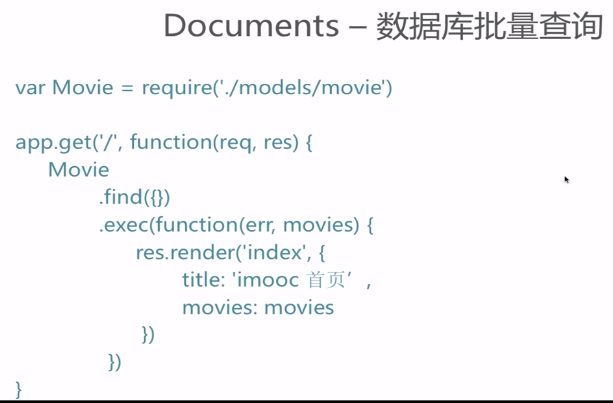




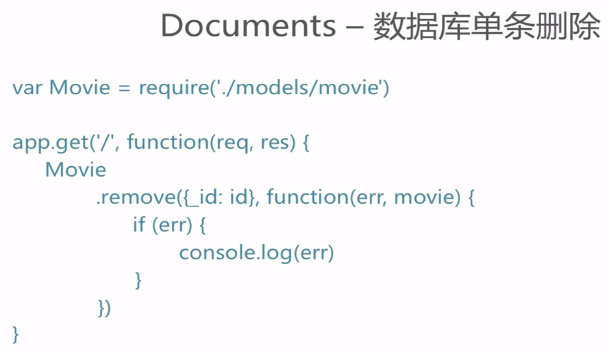


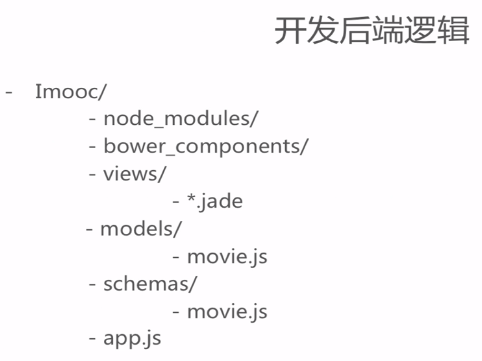












Me :

**安装**

首先假定你已经安装了 [Node.js](https://nodejs.org/)，接下来为你的应用创建一个目录，然后进入此目录并将其作为当前工作目录。

$ mkdir myapp

$ cd myapp

通过 npm init 命令为你的应用创建一个 package.json 文件。 欲了解 package.json 是如何起作用的，请参考 [Specifics of npm’s package.json handling](https://docs.npmjs.com/files/package.json)。

$ npm init

此命令将要求你输入几个参数，例如此应用的名称和版本。 你可以直接按“回车”键接受默认设置即可，下面这个除外：

entry point: (index.js)

键入 app.js 或者你所希望的名称，这是当前应用的入口文件。如果你希望采用默认的 index.js 文件名，只需按“回车”键即可。

接下来安装 Express 并将其保存到依赖列表中：

$ npm install express --save

如果只是临时安装 Express，不想将它添加到依赖列表中，只需略去 --save 参数即可：

$ npm install express

安装pug

install pug-cli –save-dev

安装moment

Moment.js是一个javascript日起处理类库，用于解析，检验，操作，以及显示日期的工具，支持多语言，网站http://momentjs.cn

**安装方法**

bower install moment --save # bower  
npm install moment --save   # npm  
Install-Package Moment.js   # NuGet  
spm install moment --save   # spm  
meteor add momentjs:moment  # meteor

**日期格式化**

moment().format('MMMM Do YYYY, h:mm:ss a'); // 二月 5日 2017, 12:09:10 中午  
moment().format('dddd');                    // 星期日  
moment().format("MMM Do YY");               // 2月 5日 17  
moment().format('YYYY [escaped] YYYY');     // 2017 escaped 2017  
moment().format();                          // 2017-02-05T12:09:10+08:00

**相对时间**

moment("20111031", "YYYYMMDD").fromNow(); // 5 年前  
moment("20120620", "YYYYMMDD").fromNow(); // 5 年前  
moment().startOf('day').fromNow();        // 12 小时前  
moment().endOf('day').fromNow();          // 12 小时内  
moment().startOf('hour').fromNow();       // 9 分钟前

**日历时间**

moment().subtract(10, 'days').calendar(); // 2017年1月26日  
moment().subtract(6, 'days').calendar();  // 本周一中午12点09  
moment().subtract(3, 'days').calendar();  // 本周四中午12点09  
moment().subtract(1, 'days').calendar();  // 昨天中午12点09分  
moment().calendar();                      // 今天中午12点09分  
moment().add(1, 'days').calendar();       // 明天中午12点09分  
moment().add(3, 'days').calendar();       // 下周三中午12点09  
moment().add(10, 'days').calendar();      // 2017年2月15日

**多语言支持**

moment().format('L');    // 2017-02-05  
moment().format('l');    // 2017-02-05  
moment().format('LL');   // 2017年2月5日  
moment().format('ll');   // 2017年2月5日  
moment().format('LLL');  // 2017年2月5日中午12点09分  
moment().format('lll');  // 2017年2月5日中午12点09分  
moment().format('LLLL'); // 2017年2月5日星期日中午12点09分  
moment().format('llll'); // 2017年2月5日星期日中午12点09分

安装body-parser

npm install body-parser –save-dev

\_id 是mongodb的默认主键。TypeError: Cannot read property '\_id' of undefined。我的出现这个错误的原因是app.use(bodyParser.urlencoded({extended:false})); 把extended:false改为extended:true。就可以了。

安装mongoose

npm install mongoose –save-dev/

**Plugging in your own Promises Library**

New in Mongoose 4.1.0

While mpromise is sufficient for basic use cases, advanced users may want to plug in their favorite [ES6-style promises library](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) like [bluebird](https://www.npmjs.com/package/bluebird), or just use native ES6 promises. Just set mongoose.Promise to your favorite ES6-style promise constructor and mongoose will use it.

Mongoose tests with ES6 native promises, [bluebird](https://www.npmjs.com/package/bluebird), and [q](https://www.npmjs.com/package/q). Any promise library that exports an ES6-style promise constructor should work in theory, but theory often differs from practice. If you find a bug, open [an issue on GitHub](https://github.com/Automattic/mongoose/issues)

// Use native promises

mongoose.Promise = global.Promise;

assert.equal(query.exec().constructor, global.Promise);

// Use bluebird

mongoose.Promise = require('bluebird');

assert.equal(query.exec().constructor, require('bluebird'));

// Use q. Note that you \*\*must\*\* use `require('q').Promise`.

mongoose.Promise = require('q').Promise;

assert.ok(query.exec() **instanceof** require('q').makePromise);

F盘创建 data/db文件夹

命令行进入F盘 ，输入 mongod启动服务

新开一个命令行窗口 直接输入mongo

use users;

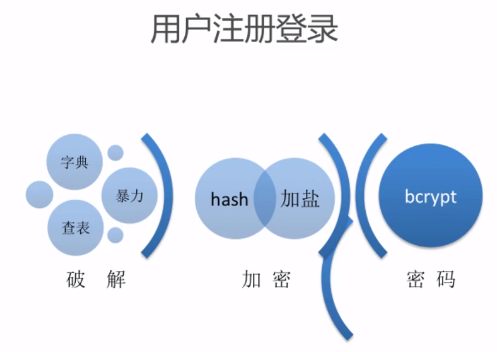
db.user.insert({})

db.user.find()

db.users.count({userid:'mldn',password:'hello'}) ;

存储在app.locals中的这些键值对一般是公共模板方法或者公共模板变量，express提供了这样的机制，便于公共数据和方法在模板中的使用，而无需每次render手动传入，这种思想值得我们设计代码框架时学习

Nodsjs+mangodb建站攻略（二期）



安装bcrypt

npm install bcrypt --save-dev

#### To hash a password:

Technique 1 (generate a salt and hash on separate function calls):

bcrypt.genSalt(saltRounds, function(err, salt) {

    bcrypt.hash(myPlaintextPassword, salt, function(err, hash) {

*// Store hash in your password DB.*

    });

});

Technique 2 (auto-gen a salt and hash):

bcrypt.hash(myPlaintextPassword, saltRounds, function(err, hash) {

*// Store hash in your password DB.*

});

#### To check a password:

*// Load hash from your password DB.*

bcrypt.compare(myPlaintextPassword, hash, function(err, res) {

*// res == true*

});

bcrypt.compare(someOtherPlaintextPassword, hash, function(err, res) {

*// res == false*

});

安装cookie-session

npm install cookie-session --save-dev

var cookieSession **=** require('cookie-session')

var express **=** require('express')

var app **=** express()

app.use(cookieSession({

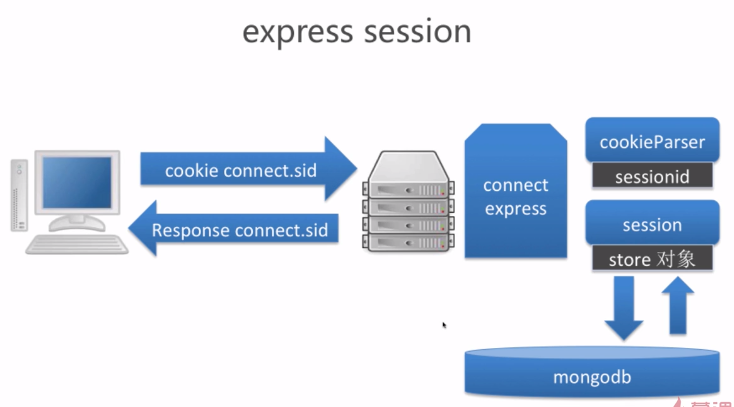
  name**:** 'session',

  keys**:** [*/\* secret keys \*/*],

*// Cookie Options*

  maxAge**:** 24 **\*** 60 **\*** 60 **\*** 1000 *// 24 hours*

}))



日志打印 安装morgan

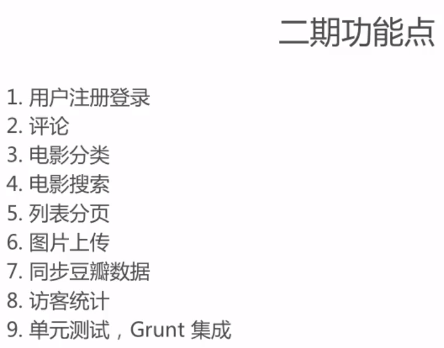
npm install morgan --save-dev





文件上传

npm install connect-multiparty --save-dev



npm install gulp --save-dev

npm install gulp-uglify --save-dev