



Curso de NodeJS

Unidad Didáctica 07: App Completa



Ayuntamiento
de Vitoria-Gasteiz
Vitoria-Gasteizko
Udala

Índice de contenidos

- Introducción
- Controladores
- Vistas
- Formularios
- Navegación
- Conclusiones



Introducción

Hasta ahora hemos visto cómo gestionar el modelo y la gestión de peticiones. Es hora de que veamos cómo se puede integrar con un sistema de vista.



Controladores

Los controladores son aquellos que nos van a relacionar las funcionalidades con las peticiones entrantes



Controladores

Primero debemos definir las funciones y el nombre en los controladores

```
// Display book create form on GET
```

```
exports.book_create_get = function(req, res, next) {  
    res.send('NOT IMPLEMENTED: Book create GET');  
};
```



Controladores

Después debemos usar esas definiciones en la gestión de las peticiones

```
/* GET request for creating a Book. NOTE This must  
come before routes that display Book (uses id) */
```

```
router.get('/book/create',  
book_controller.book_create_get);
```



Controladores

Una vez creada y cargada la ruta y asociada a la funcionalidad ya podríamos probarla



Vistas

Node JS dispone de distintos motores de vistas

Pug (antiguamente llamado Jade) es uno de los principales



Vistas

Se puede configurar con express de la siguiente manera

```
// view engine setup
```

```
app.set('views', path.join(__dirname, 'views'));
```

```
app.set('view engine', 'pug');
```



Vistas

Esto nos permitirá cargar las vistas desde el directorio views usándolo como base de los enlaces internos a las plantillas



Vistas

Los ficheros mínimos de funcionamiento deberían ser:

error.pug página de error

index.pug página principal

layout.pug layout principal



Vistas

Desde los controladores se cargan las vistas con la función render

```
// Display list of all Books

exports.book_list = function(req, res, next) {

  Book.find({}, 'title author ')

    .populate('author')

    .exec(function (err, list_books) {

      if (err) { return next(err); }

      //Successful, so render

      res.render('book_list', { title: 'Book List', book_list: list_books });

    });

};
```

<http://cursosdedesarrollo.com/>



Vistas

Las variables de vista luego se pueden usar desde la vista invocada

```
extends layout
```

```
block content
```

```
h1= title
```

```
ul
```

```
each book in book_list
```

```
li
```

```
  a(href=book.url) #{book.title}
```

```
  | (#{book.author.name})
```

```
else
```

```
li There are no books.
```

<http://cursosdedesarrollo.com/>



Vistas

La extensión o herencia nos va a permitir gestionar inclusiones en los layout



Vistas

```
//- layout.pug
```

```
html
```

```
  head
```

```
    title My Site - #{title}
```

```
    block scripts
```

```
      script(src='/jquery.js')
```

```
  body
```

```
    block content
```

```
    block foot
```

```
      #footer
```

```
        p some footer content
```

<http://cursosdedesarrollo.com/>



Vistas

```
//- page-a.pug
```

```
extends layout.pug
```

```
block scripts
```

```
  script(src='/jquery.js')
```

```
  script(src='/pets.js')
```

```
block content
```

```
  h1= title
```

```
  - var pets = ['cat', 'dog']
```

```
  each petName in pets
```

```
    include pet.pug
```

<http://cursosdedesarrollo.com/>



Vistas

También permiten la gestión de la herencia de layouts

<https://pugjs.org/language/inheritance.html>

<http://cursosdedesarrollo.com/>



Vistas

```
//- sub-layout.pug
```

```
extends layout.pug
```

```
block content
```

```
.sidebar
```

```
block sidebar
```

```
p nothing
```

```
.primary
```

```
block primary
```

```
p nothing
```

<http://cursosdedesarrollo.com/>



Vistas

```
//- page-b.pug
```

```
extends sub-layout.pug
```

```
block content
```

```
.sidebar
```

```
block sidebar
```

```
p nothing
```

```
.primary
```

```
block primary
```

```
p nothing
```

<http://cursosdedesarrollo.com/>



Vistas

La sintaxis es básica la indentación va a marcar la anidación de los elementos

```
doctype html
```

```
html(lang="en")
```

```
  head
```

```
    title= title
```

```
    script(type='text/javascript').
```

```
  body
```

```
    h1= title
```

<http://cursosdedesarrollo.com/>



Vistas

Los atributos de las etiquetas se definen entre ()

`script(type='text/javascript')`

`link(rel='stylesheet', href='/stylesheets/style.css')`



Vistas

Las variables pasadas a la vista se pueden imprimir

```
h1= title
```

```
p= 'Evaluated and <em>escaped expression</em>:'  
+ title
```



Vistas

Se puede insertar texto escapado o sin escapar

p This is a line with #[em some emphasis] and
#[strong strong text] markup.

p This line has an un-escaped string: !{' is
emphasised'}, an escaped string: #{' is
not emphasised'}, and escaped variables:
#{title}.

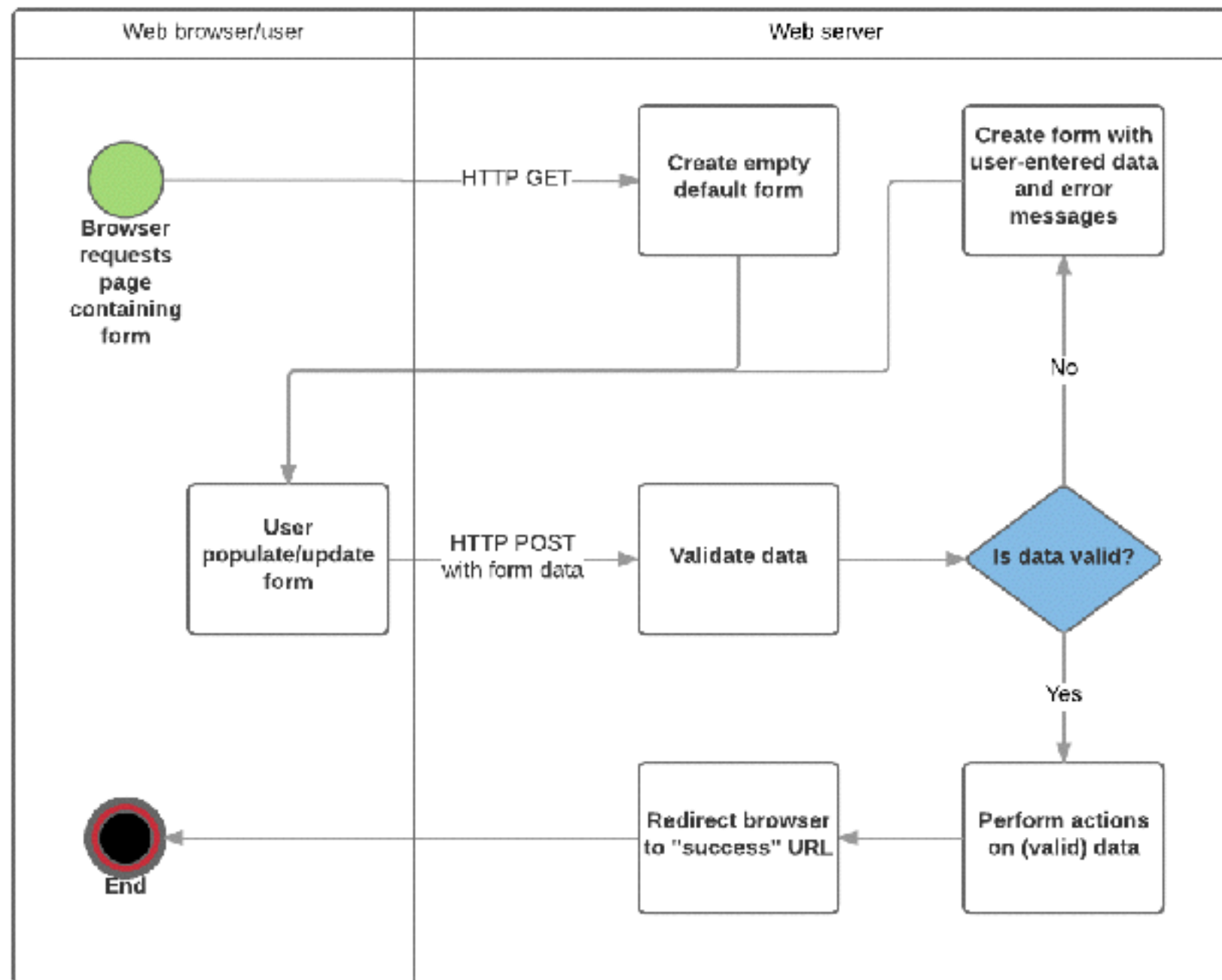


Formularios

Los formularios es una parte esencial de la gestión de los datos, a continuación veremos cómo maneja express los formularios



Formularios



Formularios

Para la validación del formulario necesitaremos usar

<https://www.npmjs.com/package/express-validator>



Formularios

Por lo que instalaremos el módulo

```
npm install express-validator —save
```



Formularios

En la importación deberemos incluirlo

```
var expressValidator = require('express-validator');
```

```
app.use(expressValidator()); // Add this after the  
bodyParser middlewares!
```



Formularios

En la recogida de datos del formulario podremos gestionar las validaciones

```
req.checkBody('name', 'Invalid name').isAlpha();
```

```
req.checkBody('age', 'Invalid age').notEmpty().isInt();
```



Formularios

También podremos limpiar los campos

```
req.sanitizeBody('name').escape();
```



Formularios

Después ya podremos preguntar si los campos son válidos:

```
var errors = req.validationErrors();

if (errors) {

    // Render the form using error information

}

else {

    // There are no errors so perform action with valid data (e.g. save record).

}
```



Formularios

Los posibles errores nos vendrán en un array de objetos con los siguientes campos:

```
{param: 'name', msg: 'Invalid name', value:  
  '<received input>'}
```



Formularios

Ejemplo de vista de formulario

```
extends layout
```

```
block content
```

```
h1 #{title}
```

```
form(method='POST' action='')
```

```
div.form-group
```

```
label(for='name') Genre:
```

```
input#name.form-control(type='text', placeholder='Fantasy, Poetry etc.' name='name' value=(undefined===genre ? '' :  
genre.name))
```

```
button.btn.btn-primary(type='submit') Submit
```

```
if errors
```

```
ul
```

```
for error in errors
```

```
li!= error.msg
```

<http://cursosdedesarrollo.com/>



Formularios

Ejemplo completo de la utilización de formularios

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/forms

<http://cursosdedesarrollo.com/>



Navegación

La navegación es uno de los apartados más importantes ya que nos permite llegar a todas las páginas del sitio web



Navegación

```
ul(class='sidebar-nav')
```

```
  li
```

```
    a(href='/catalog') Home
```

```
  li
```

```
    a(href='/catalog/books') All books
```

```
  li
```

```
    a(href='/catalog/authors') All authors
```

```
  li
```

```
    a(href='/catalog/genres') All genres
```

```
  li
```

```
    a(href='/catalog/bookinstances') All book-instances
```

<http://cursosdedesarrollo.com/>



Navegación

Las rutas deben coincidir con aquellas rutas definidas en el router de los componentes



Navegación

Los includes nos van a permitir componentizar la vista



Navegación

```
//- index.pug
```

```
doctype html
```

```
html
```

```
include includes/head.pug
```

```
body
```

```
h1 My Site
```

```
p Welcome to my super lame site.
```

```
include includes/foot.pug
```

<http://cursosdedesarrollo.com/>



Navegación

```
//- includes/head.pug
```

```
head
```

```
  title My Site
```

```
  script(src='/javascripts/jquery.js')
```

```
  script(src='/javascripts/app.js')
```



Navegación

```
//- includes/head.pug
```

```
head
```

```
  title My Site
```

```
  script(src='/javascripts/jquery.js')
```

```
  script(src='/javascripts/app.js')
```



Navegación

```
//- includes/foot.pug
```

```
footer#footer
```

```
p Copyright (c) foobar
```



Conclusiones

Hemos visto cómo realizar la integración con el motor de vista para dar al usuario una manera de interactuar con la aplicación



Datos de Contacto

<http://www.cursosdedesarrollo.com>
david@cursosdedesarrollo.com

<http://cursosdedesarrollo.com/>



Licencia



David Vaquero Santiago

Esta obra está bajo una
Licencia Creative Commons
Atribución-NoComercial-
CompartirIgual 4.0 Internacional

