

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

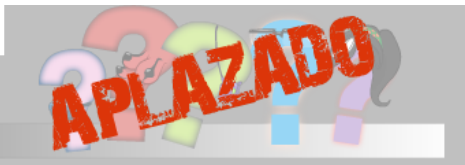
JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

AdictosAlTrabajo

Final de Terrakas
¡¡Ven al estreno!!
terrakas.com



Entra en Adictos a través de  

E-mail

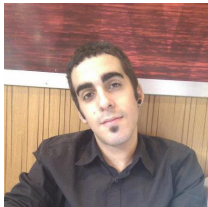
Contraseña

Entrar [Deseo registrarme](#)
[Olvidé mi contraseña](#)



[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestros libros](#) [Más](#)

» Estás en: [Inicio](#) [Tutoriales](#) [Tutorial Emberjs](#)



Rafael Macías Rodríguez

Rafael es un alumno becario en prácticas, procedente del I.E.S. Rey Fernando VI

[Ver todos los tutoriales del autor](#)

Fecha de publicación del tutorial: 2009-02-26

Tutorial visitado 2 veces [Descargar en PDF](#)

Trabajando con Emberjs

0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. ¿Qué es Emberjs?
- 4. Patrón MVC
- 5. Iniciar aplicación Ember
- 6. Creación de plantillas con Handlebars
- 7. Creación rutas y estados.
- 8. Creación del mvc.
- 9. Resultado y usos
- 10. Conclusiones

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Intel Core 2 CPU T7200 @ 2.00GHz x 2
- Sistema Operativo: Ubuntu 12.04 LTS x32
- netBeans IDE 7.3

2. Introducción

En este tutorial vamos a conocer Emberjs, veremos qué es Ember, cómo lo utilizaremos y una breve introducción a su API.

3. ¿Qué es Emberjs?

Ember es un framework para javascript, que nos permite utilizar el patrón MVC (Modelo Vista Controlador) en nuestras aplicaciones de manera más automatizada y sencilla. Podemos descargarnos esta librería desde la [página oficial de Ember](#).

Así como un paquete de iniciación donde encontraremos a parte de esta, otras librerías en las que se apoya Ember. Se apoya en jQuery, una librería javascript, que nos permite manejar nuestra aplicación en el entorno cliente, con toda la potencia de javascript pero de manera más sencilla. Si quieres conocer más sobre jQuery puedes acceder a toda la información que te proporcionan en su página oficial, así como la documentación y el API [aquí](#).

También se apoya en [Handlebars](#), una librería javascript que nos permite crear plantillas en el HTML, algo indispensable en el patrón MVC

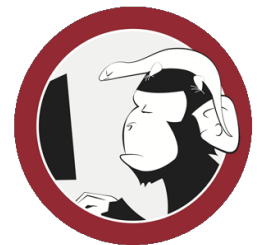
Ahora vamos a explicar los fundamentos básicos para crear vuestra aplicación con Ember. Lo más importante a tener en cuenta es tener las librerías que acabo de nombrar en el directorio de vuestro proyecto, obviamos que tenéis que incluirlas en el html.

4. Patrón MVC

Para que entendáis un poco como va el asunto, voy a explicar los fundamentos del patrón MVC (Modelo Vista controlador).

- **Vista:** es la representación de los datos al usuario.
- **Controlador:** código que obtiene esos datos dinámicamente y genera el contenido que se mostrará al usuario en las vistas. Los obtiene accediendo a los modelos.
- **Modelo:** los datos almacenados, junto con las reglas de negocio que transforman esa información (teniendo en cuenta las acciones de los usuarios).

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» Atención, APLAZADO
Estreno último capítulo de Terrakas

» Vendedor: Soy inseguro, filtra o elige por mi: si quieres que te compre.

» Comentando el libro: El arte de pensar, de Rolf Dobelli

» Ya está a la venta mi segundo libro: Planifica tu éxito, de aprendiz a empresario

» Ya esta disponible en eBook mi primer libro: Informática Profesional

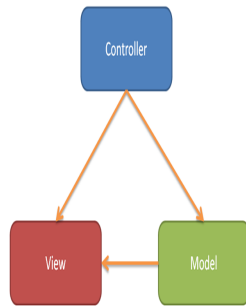
[Histórico de noticias](#)

Últimos Tutoriales

» Webs que ofrecen plantillas CSS gratuitas

» Resolver problema LockTimeoutException en Spring Web Flow

» WebSockets con Stomp y



ActiveMQ: ¿chateamos?

» Análisis de los sentimientos en twitter con el soporte de Apicultur.

» Creación de presentaciones con PowToon

Últimas ofertas de empleo

2011-09-08

Comercial - Ventas - MADRID.

2011-09-03

Comercial - Ventas - VALENCIA.

2011-08-19

Comercial - Compras - ALICANTE.

2011-07-12

Otras Sin catalogar - MADRID.

2011-07-06

Otras Sin catalogar - LUGO.

5. Iniciar aplicación Ember

Una vez visto esto, vamos a ver como creamos esto con Ember.

Lo primero de todo es inicializar nuestra aplicación en un fichero js de la siguiente manera:

```
1 | App = Ember.Application.create();
```

Con este código crearemos nuestra aplicación en el namespace "App".

6. Creación de plantillas con Handlebars

Lo siguiente que vamos a ver, es como creamos una plantilla, podríamos decir que es una forma de representar la información, no es lo que Ember trata totalmente como una vista, dado que Ember tiene una forma de crearlas, que luego veremos.

Para utilizar plantillas se apoya en Handlebars:

```
1 | <script type="text/x-handlebars">
2 |   <h2><strong>{{firstName}} {{lastName}}</strong></h2>
3 | </script>
```

Si queremos crear distintas plantillas, basta con darles el atributo data-template-name:

```
1 | <script type="text/x-handlebars" data-template-name="Index">
2 |   <h2><strong>{{firstName}} {{lastName}}</strong></h2>
3 | </script>
```

7. Creación de rutas.

Tenemos la posibilidad de crear unas rutas, con un enrutador que nos provee Ember. Este nos permite interactuar con nuestra aplicación y movernos entre diferentes estados. Basta con ejecutar el método route dentro del método map del objeto Router.

```
1 | App.Router.map( function() {
2 |   this.route( 'index', { path: "/" } ); // Takes us to "/"
3 | });
```

8. Creación del MVC

Lo siguiente es ver como crear un controlador. La sintaxis que nos pide Ember es sencilla, si queremos crear un controlador la sintaxis es la siguiente:

```
1 | App.NombreController = Ember.Controller.extend({
2 |
3 | });
```

En el caso de index:

```
1 | App.IndexController = Ember.Controller.extend({
2 |
3 | });
```

En la mayoría de las aplicaciones hechas con Ember los modelos son manejados con Ember Data, una librería construida con Ember. Ember data nos ayuda a obtener los datos de un servidor, hacer los cambios en el navegador y guardar estos cambios para enviarlos de vuelta al servidor, es algo parecido a lo que nos proporcionan los ORMs. Para esta labor se apoya en la tecnología RESTful JSON, pero también podemos crear nuestros modelos a mano.

Para crear un modelo manualmente tenemos que extender el objeto Model:

```
1 | App.Person = DS.Model.extend({
2 |   firstName: DS.attr('string'),
3 |   lastName: DS.attr('string'),
4 |   birthday: DS.attr('date'),
5 |
6 |   fullName: function() {
7 |     return this.get('firstName') + ' ' + this.get('lastName');
8 |   }.property('firstName', 'lastName')
9 | });
```

Como vemos podemos definir atributos, funciones y atributos computables. Esto lo podemos hacer también con los controladores, rutas etc.

La forma de crear vistas con Ember es ejecutando el método create del objeto View:

```
1 | var view = Ember.View.create({
2 |   name: "Bob"
3 | });
```

9. Resultado y usos

Ahora voy a enseñaros como podemos mostrar los datos de un modelo en nuestra plantilla html. Es sencillo:

```
1 <script type="text/x-handlebars" data-template-name="say-hello">
2   Hello, <b>{{App.User.name}}</b>
3 </script>
```

Uno de los potenciales de Ember es, entre otros, que cualquier propiedad de un objeto Ember tiene un sufijo Binding ligado a el para ser tratado de forma especial.

Por ejemplo tenemos la expresion valueBinding, que aplica esta regla al atributo value.

[Ejemplo de uso](#)

10. Conclusiones

Ember nos puede proporcionar las herramientas necesarias para utilizar este patrón en nuestra aplicación, y tener abstraída la parte lógica y funcional de los datos.

Para más información no dudéis en buscarla en la página oficial o internet:

- 1. [Guías oficial Ember](#)
- 2. [Guía inicial en net.tutsplus](#)
- 3. [Documentación en github](#)

Cualquier duda o sugerencia podeis comentarlo.

Saludos.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

0

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «

 Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

0 personas han traído clicks a esta página

sin clicks

+

+

+

+

+

+

+

+

powered by [karmacracy](#)