



EBook Gratis

APRENDIZAJE loopbackjs

Free unaffiliated eBook created from
Stack Overflow contributors.

#loopbackjs

Tabla de contenido

| | |
|---|----------|
| Acerca de..... | 1 |
| Capítulo 1: Empezando con loopbackjs..... | 2 |
| Observaciones..... | 2 |
| Examples..... | 2 |
| Instalación y puesta en marcha..... | 2 |
| Capítulo 2: Crear modelos..... | 4 |
| Examples..... | 4 |
| Creación de modelos con la interfaz de línea de comandos..... | 4 |
| Creditos..... | 6 |

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [loopbackjs](#)

It is an unofficial and free loopbackjs ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official loopbackjs.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con loopbackjs

Observaciones

Esta sección proporciona una descripción general de qué es loopbackjs y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de loopbackjs, y vincular a los temas relacionados. Dado que la Documentación para loopbackjs es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación y puesta en marcha

Instalación:

Para instalar la herramienta de interfaz de línea de comandos (CLI) de LoopBack:

```
npm install -g loopback-cli
```

Esto instala la herramienta de línea de comandos `lb` para el andamiaje y la modificación de aplicaciones LoopBack.

Crear nueva aplicación:

Para crear una nueva aplicación:

```
lb
```

El generador de aplicaciones LoopBack lo recibirá con un arte ASCII amigable y le pedirá el nombre de la aplicación. Simplemente responda las preguntas hechas por la herramienta para crear una aplicación.

Crear primer modelo:

Vaya a su nuevo directorio de aplicaciones, luego ejecute:

```
lb model
```

Una vez más, simplemente responda todas las preguntas hechas por la herramienta y cree su primer modelo.

Ejecute la aplicación: Desde la carpeta raíz de su proyecto, ejecute:

```
node .
```

Abra su navegador en <http://0.0.0.0:3000/> (en algunos sistemas, es posible que necesite usar <http://localhost:3000>). Verá la respuesta de la aplicación predeterminada que muestra algunos JSON con cierta información de estado.

Lea [Empezando con loopbackjs en línea](#):

<https://riptutorial.com/es/loopbackjs/topic/2752/empezando-con-loopbackjs>

Capítulo 2: Crear modelos

Examples

Creación de modelos con la interfaz de línea de comandos.

En el indicador de comandos del nodo JS, dentro de su proyecto de bucle de retorno, escriba el siguiente comando para crear un nuevo modelo.

```
slc loopback:model
```

Si ha instalado la herramienta CLI LoopBack, puede crear un modelo con:

```
lb model
```

El símbolo del sistema solicitará información sobre el modelo que se creará. En este ejemplo, se creará un modelo `Foo`.

```
Enter the model name: Foo
```

Los modelos pueden basarse en una de las clases de bucle invertido incorporadas. Los modelos persistentes son la opción más común, representan datos estructurados que deben almacenarse (= **persistir**) dentro de una base de datos.

```
[?] Select model's base class: (Use arrow keys)
  Model
   PersistedModel
    ACL
    AccessToken
    Application
    Change
    Checkpoint
```

Es posible exponer a `Foo` través de una API REST. La API generada contendrá métodos para leer, crear, actualizar y eliminar instancias del modelo `Foo`.

```
[?] Expose Foo via the REST API? (Y/n) Y
```

Para definir todas las URL de REST API para este modelo, Loopback usa la forma plural. En el siguiente mensaje a continuación, dejando el campo vacío significa que el carácter `s` se agregará al nombre del modelo `Foo`.

Por ejemplo, loopback definirá el punto final `GET /Foos`.

Pero algunos nombres no tienen una forma plural, como `Repository` convierte en `Repositories` en forma plural. En esos casos, escriba la forma plural en esos campos.

```
[?] Custom plural form (used to build REST URL):
```

TBD

```
? Common model or server only?
```

```
□ common  
server
```

```
Let's add some Foo properties now.  
Enter an empty property name when done.
```

```
[?] Property name: name
```

```
invoke loopback:property
```

```
[?] Property type: (Use arrow keys)
```

```
□ string  
number  
boolean  
object  
array  
date  
buffer  
geopoint  
(other)
```

```
[?] Required? (y/N)
```

```
node .
```

Navegue a `localhost:3000/explorer` y observe la API REST generada para `Foo`

Lea Crear modelos en línea: <https://riptutorial.com/es/loopbackjs/topic/4450/crear-modelos>

Creditos

| S. No | Capítulos | Contributors |
|-------|--------------------------|---|
| 1 | Empezando con loopbackjs | Community , Ebrahim Pasbani , mbouclas , Nilesh G |
| 2 | Crear modelos | Nilesh G , Overdrivr |