



**FREE eBook**

# LEARNING loopbackjs

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#loopbackjs**

# Table of Contents

<b>About</b> .....	<b>1</b>
<b>Chapter 1: Getting started with loopbackjs</b> .....	<b>2</b>
Remarks.....	2
Examples.....	2
Installation & Getting started.....	2
<b>Chapter 2: Create models</b> .....	<b>4</b>
Examples.....	4
Model creation with the command line interface.....	4
<b>Credits</b> .....	<b>6</b>

---

# About

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](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with loopbackjs

## Remarks

This section provides an overview of what loopbackjs is, and why a developer might want to use it.

It should also mention any large subjects within loopbackjs, and link out to the related topics. Since the Documentation for loopbackjs is new, you may need to create initial versions of those related topics.

## Examples

### Installation & Getting started

#### Installation:

To install the LoopBack command-line interface (CLI) tool:

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

This installs the `lb` command-line tool for scaffolding and modifying LoopBack applications.

#### Create new application:

To create a new application:

```
lb
```

The LoopBack application generator will greet you with some friendly ASCII art and prompt you for the name of the application. Simply answer questions asked by tool to create an application.

#### Create first model:

Go into your new application directory, then run:

```
lb model
```

Again, simply answer all questions asked by tool and create your first model.

**Run the application:** From your project's root folder, run:

```
node .
```

Open your browser to <http://0.0.0.0:3000/> (on some systems, you may need to use <http://localhost:3000> instead). You'll see the default application response that displays some JSON with some status information.

Read **Getting started with loopbackjs** online: <https://riptutorial.com/loopbackjs/topic/2752/getting-started-with-loopbackjs>

# Chapter 2: Create models

## Examples

### Model creation with the command line interface

In the node JS command prompt, inside your loopback project, type the following command to create a new model.

```
slc loopback:model
```

If you have installed LoopBack CLI tool, you can create model with:

```
lb model
```

The command prompt will request informations about the model to create. In this example, a model `Foo` will be created.

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

Models can be based on one of the built-in loopback classes. Persisted models are the most common choice, representing structured data that should be stored (= **persisted**) inside a database.

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

It is possible to expose `Foo` through a REST API. The generated API will contain methods for reading, creating, updating and deleting instances of the `Foo` model.

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

To define all REST API urls for this model, Loopback uses the plural form. In the next prompt below, leaving the field empty means the `s` character will be appended to the model's name `Foo`.

For instance, loopback will define the endpoint `GET /Foos`.

But some names do not have such a plural form, such as `Repository` becomes `Repositories` in plural form. In those cases, type the plural form in those fields.

```
[?] 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  
  geoint  
  (other)
```

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

```
node .
```

Navigate to `localhost:3000/explorer` and observe the REST API generated for `Foo`

Read Create models online: <https://riptutorial.com/loopbackjs/topic/4450/create-models>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with loopbackjs	<a href="#">Community</a> , <a href="#">Ebrahim Pasbani</a> , <a href="#">mbouclas</a> , <a href="#">Nilesh G</a>
2	Create models	<a href="#">Nilesh G</a> , <a href="#">Overdriv</a>