# Create a simple API

## Redirecting to http://loopback.io/doc/en/lb2/Create-a-simple-API.html ...

**Prerequisite**: Install StrongLoop software as described in Installing StrongLoop.

**Recommended**: Read LoopBack core concepts.



Use the LoopBack command-line tool, `slc loopback`, to quickly create a LoopBack application, models, and data sources.

Use the LoopBack command-line tool, `slc loopback`, to create and *scaffold* applications. Scaffolding simply means generating the basic code for your application to save you time. You can then extend and modify the code as desired for your specific needs.

Before following this tutorial, make sure you have the latest version of StrongLoop with:

```
$ npm install -g strongloop
```

Make sure you run this command recently (within 24 hours) to get the latest updates. We're always improving LoopBack!

## Create new application

To create a new application, run the LoopBack application generator:

```
$ slc loopback
```

The LoopBack generator will greet you with some friendly ASCII art and prompt you for the name of the application.

Enter `loopback-getting-started`. Then the generator will prompt you for the name of the directory to contain the project; press Enter to accept the default (the same as the application name):

```
_-----_
    |       |    .--------------------------.
    |--(o)--|    |  Let's create a LoopBack |
   `---------´   |       application!       |
    ( _´U`_ )    '--------------------------'
    /___A___\
     |  ~  |
   __'.___.'__
 ´   `  |° ´ Y `
[?] What's the name of your application? loopback-getting-started
[?] Enter name of the directory to contain the project: loopback-getting-started
```

You can use a different name for the application, but if you do, be sure to substitute your name for "loopback-getting-started" throughout the rest of this tutorial.

The generator will scaffold the application including:

1. Initializing the project folder structure.
2. Creating default JSON files.
3. Creating default JavaScript files.
4. Downloading and installing dependent Node modules (as if you had manually done `npm install`).

## Create models

Now that you've scaffolded the initial project, you're going to create create a *CoffeeShop* model that will automatically have REST API endpoints.

Go into your new application directory, then run the LoopBack model generator:

```
$ cd loopback-getting-started
$ slc loopback:model
```

The generator will prompt for a model name.  Enter **CoffeeShop**:

```
[?] Enter the model name: CoffeeShop
```

It will ask if you want to attach the model to any data sources that have already been defined.

At this point, only the default in-memory data source is available.  Press **Enter** to select it:

```
...
[?] Select the data-source to attach CoffeeShop to: (Use arrow keys)
 db (memory)
```

Then the generator will prompt you for the base class to use for the model.  Since you will eventually connect this model to a persistent data source in a database, press down-arrow to choose **PersistedModel**, then press **Enter**:

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

PersistedModel is the base object for all models connected to a persistent data source such as a database.  See LoopBack core concepts for an overview of the model inheritance hierarchy.

One of the powerful advantages of LoopBack is that it automatically generates a REST API for your model.  The generator will ask whether you want to expose this REST API.

Hit **Enter** again to accept the default and expose the Person model via REST:

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

LoopBack automatically creates a REST route associated with your model using the *plural* of the model name.  By default, it pluralizes the name for you (by adding "s"), but you can specify a custom plural form if you wish.  See Exposing models over REST for all the details.

Press **Enter** to accept the default plural form (CoffeeShops):

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

Next, you'll be asked whether you want to create the model on the server only or in the `/common` directory, where it can potentially be used by both server and client LoopBack APIs.  Keep, the default, common, even though in this application you'll only be working with server-side models:

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

Every model has properties.  Right now, you're going to define one property, "name," for the CoffeeShop model.

Select **`string`** as the property type (press **Enter**, since string is the default choice):

```
Let's add some CoffeeShop 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)
```

Each property can be optional or required. Enter **y** to make `name` required:

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

Then you'll be prompted to enter a default value for the property; press Enter for no default value:

```
? Default value[leave blank for none]:
```

Then, you'll be prompted to add another property.  Follow the prompts to add a required property named "city."

```
Let's add another CoffeeShop property.
? Property name: city
? Property type: string
? Required? Yes
```

End the model creation process by pressing **Enter** when prompted for the name of the next property.

The model generator will create two files in the application's `common/models` directory that define the model: `coffee-shop.json` and `coffee-shop.js`.



The LoopBack model generator, `slc loopback:model`, automatically converts camel-case model names (for example MyModel) to lowercase dashed names (my-model).  For example, if you create a model named "FooBar" with the model generator, it creates files `foo-bar.json` and `foo-bar.js` in `common/models`.  However, the model name ("FooBar") will be preserved via the model's name property.

## Check out the project structure

For all the details of the canonical LoopBack application structure, see Project layout reference.

## Run the application

Start the application:

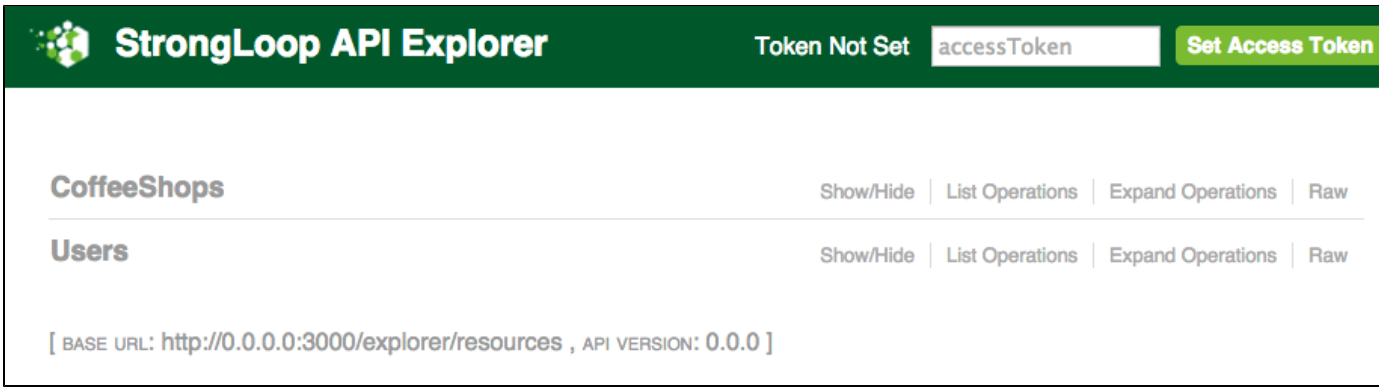
```
$ node .
...
Browse your REST API at http://0.0.0.0:3000/explorer
Web server listening at: http://0.0.0.0:3000/
```

Running your app with the `node` command is appropriate when you're developing on your local machine. Once you're ready to prepare for moving to production, you can run it with `slc start` to run it under control of StrongLoop Process Manager, that provides options for clustering, logging, monitoring, and much more. See Operating Node applications for more information on the power of the `slc` com mand-line tool.

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; for example:

```
{"started":"2014-11-20T21:59:47.155Z","uptime":42.054}
```

Now open your browser to http://0.0.0.0:3000/explorer or http://localhost:3000/explorer. You'll see the StrongLoop API Explorer:




Through a set of simple steps using LoopBack, you've created a CoffeeShop model, specified its properties and then exposed it through REST.

**Next**: In Use API Explorer, you'll explore the REST API you just created in more depth and exercise some of its operations.