Cheatography

Total.js cheat sheet Cheat Sheet

by Louis Bertson via cheatography.com/183053/cs/38094/

Controllers

Controller Instance	`var controller = this;`	returns `instance`
`contro lle r.body`	Gets the parsed data from the request body.	returns `object`
`contro ller.ip`	Gets the HTTP host address	returns `string`
`contro lle r.l anguage`	Gets/Sets the controller language for the localization	returns `string`
`contro lle r.p arams`	Gets a dynamic params from URL	returns `object`
`contro lle r.query`	Gets the parsed data from the URL address query	returns `object`
`contro lle r.req`	Gets the HTTP request	returns `object`
`contro lle r.res`	Gets the HTTP response	returns `object`
`contro lle r.url`	Gets a relative lower-case URL	returns `string`
`contro lle r.user`	Gets the user instance if exists	returns `object`
`contro lle r.c all bac k([ vie w_n ame ])`	Returns data as JSON or view	returns `Function`
`contro lle r.f ile (fi lename, [callb - ack])`	Sends the file into response	returns controller `instance`
`contro lle r.f ile fs( name, id,[ca ll b ack])`	Sends a stored file in FileStorage	returns controller `instance`
`contro lle r.h ost nam e([ path])`	Gets the hostname	returns `string`
`contro lle r.l ayo ut( name)`	Sets a layout before the current view	returns controller `instance`
`contro lle r.i nva lid (error)`	Creates an ErrorBuilder instance	returns `ErrorB uilder`
`contro lle r.p lai n(text, [headers])`	Sends the plain text	returns controller `instance`
`contro lle r.p rox y(o pt/url)`	Creates a proxy for the current request	returns `object`
`contro lle r.r edi rec t(url, [perma m ent])`	Creates a redirection	returns controller `instance`
`contro lle r.s ucc ess ([s ucc ess], [ value])`	Alias for `contro lle r.j son (SU CCE SS ( true))`	returns controller `instance`
`contro lle r.t hro w40 4([ pro blem])`	Sends a `404 error - Not found` into response	returns controller `instance`
`contro lle r.v iew (na me/url, [model] )`	Sends the view	returns controller `instance` or `stri ng`

Info

Website	https://totaljs.com
Docs	https://docs.totaljs.com
Total.js Cloud	https://platform.totaljs.com
Telegram Community	https://t.me/totaljs
Blog	https://blog.totaljs.com
Tutorials	Youtube Videos

Setup

```
$ sudo npm install -g total4
$ total4 create empty myapp
```

Create an empty project from template.

$ (delegate) (cont)

To list available commands: `$ total4 help`

To list available templates: `$ total4 templates`

To create from template: `$ total4 create <te mpl ate _na - me> <pa th/ to/ emp ty/ fol der>`

Start Script

```
// index.js
var opt = {};
opt.port = 5000;
requir e(' tot al4 /de bug ')( opt);
```

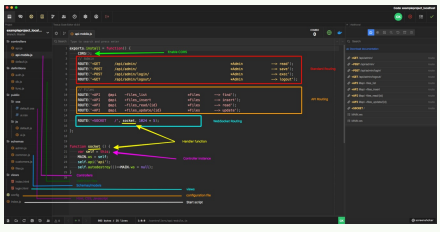
Read more `options` in docs

$ (delegate)

`$.body`	Returns body data from the request	returns `object`
`$.cont rolle r`	Contains s current instance of `Controller`	returns controller `i nstance`
`$.files`	Returns list of uploaded files	returns `Array Ht tpFile`
`$.headers`	Returns current request headers	returns `object`
`$.id`	Returns the value of the first parsed dynamic parameter	returns `string`
`$.ip`	Returns IP address form the Request	returns `string`
`$.language`	Returns parsed language from the Request	returns `String`
`$.model`	Returns the current model according to the `Schema`	returns `object`
`$.params`	Returns parsed dynamic parameters from URL	returns `object`
`$.req`	Returns the current Request instance	returns `Request`
`$.res`	Returns the current Response instance	returns `Re`
`$.user`	Returns `user` instance from the Request	returns `ob`
`$.acti on( schema, [data])`	This method is an alias fron the `CALL()` method	returns `Ca`
`$.call bac k([ error], [valu e])`	Performs a callback	returns `Fu`
`$.done ([arg])`	It wraps `$.suc c ess()` and `$ .inva lid()` method into the one function	returns `Fu sponse)`
`$.inva lid (name, [error])`	This method appends an `Err or` to `ErrorB - uilder`	returns `Er`
`$.redi rec t(url)`	Performs redirect directly from the operation	returns `Fu`
$.success([is], [value])	This method returns preddefind success object	returns `{`

Read more in docs

Project Structure (MVC)



Total.js project contains a simple directory structure, and all directories are optional. You can choose what you want to use and when.

Read more in docs

Schemas

`schema s.f ields`	Returns all fields names from the schema
`schema s.d efi ne( name, type )`	Defines the fields for the current entity
`schema s.a cti on( name, opti ons)`	Registers a new schema action
action `options`	Options for registering new schema action
`option s.name {String}`	(optional), a human readable action name
`option s.p arams {String}`	(optional), a parameter for the schema

Schemas (cont)

`option s.query {String}`	(optional), query arguments for the schema
`option s.input {String}`	(optional), input data schema
`option s.a ction {Funct ion($, mode l)}`	(required), a function that will be executed

Schemas are one of the most used parts of the Total.js framework. Schemas offer you full control over incoming data.

By **Louis Bertson**
cheatography.com/louis-bertson/

Not published yet.
Last updated 7th April, 2023.