



Emacs café

About Emacs café

# Setting up Emacs for JavaScript (part #2)

May 9, 2017 • Nicolas Petton

This is the second part of my series of articles describing how to make Emacs a great JavaScript development environment. This time we'll focus on getting good auto-completion with type inference.

If you haven't read it yet, you should jump to the [first post](#) first to get things started.

## Setting up Tern & company-mode for auto-completion

Tern is a great tool once setup correctly. It parses JavaScript files in a project and does type inference to provide meaningful completion (with type hints) and support for cross-references.

Unfortunately, cross-references with tern never reliably worked for me, that's why I have always been using [xref-js2](#) instead for that (see [part #1](#)).

For auto-completion, we'll be using company-mode with tern. Let's go ahead and install tern:

```
$ sudo npm install -g tern
```

Now let's install the Emacs packages:

```
M-x package-install RET company-tern RET
```

The Emacs configuration is straight-forward, we simply enable company-mode with the tern backend for JavaScript buffers:

```
(require 'company-mode)
(require 'company-tern)

(add-to-list 'company-backends 'company-tern)
(add-hook 'js2-mode-hook (lambda ()
                           (tern-mode)
                           (company-mode)))

;; Disable completion keybindings, as we use xref-js2 instead
(define-key tern-mode-keymap (kbd "M-.") nil)
(define-key tern-mode-keymap (kbd "M-,") nil)
```

Now, depending on your JavaScript project, you might want to setup tern to work with your project structure. If completion doesn't work out of the box using tern defaults you will have to set it up using a `.tern-project` placed in the root folder containing your JavaScript files.

Here's an example setup for a project that uses `requirejs` and `jQuery`, ignoring files from the `bower_components` directory:

```
{
  "libs": [
    "jquery"
  ],
  "loadEagerly": [
    ".*/**/*.js"
  ],
  "dontLoad": [
```

```
    "./bower_components/"
  ],
  "plugins": {
    "requirejs": {
      "baseUrl": "./"
    }
  }
}
```

Once setup, tern offers superb completion. Together with company-mode, you get great context-based completion with type inference.



When completing a function, you can hit `<F1>` to get its documentation:



```
*/
that.resolveUrl = function(aUrl) {
  if (typeof aUrl === "string") {
    aUrl = url({rawUrl: aUrl});
  }
}

that.getUrl.fr
```

freeze fn(obj) ○  
from fn(arrayLike, mapFn?, thisArg?) ○  
fromCharCode fn(code) ○  
fromCodePoint fn(point, point?) ○  
round fn(x) ○

Freezes an object: that is, prevents new properties from being added to it; prevents existing properties from being removed; and prevents existing properties, or their enumerability, configurability, or writability, from being changed.

\*company-documentation\*:1:0 All (Fundamental)

## Until next time

In the next articles I'll cover linting with Flycheck, `gulp` and `grunt` integration into Emacs, and of course how to setup and use **Indium**.

2 Comments

emacs.cafe

 Login ▾ Recommend Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS **Максим Руренко** • 3 months ago

Thank you. Setup tern for myself.

Also good example of '.tern-project' for ES6 I found here <https://atom.io/packages/at...> |  • Reply • Share ▸**Дмитрий Мазурок** • 3 months ago

Have you tried tide-mode? It originally targets typescript, but works just as fine with javascript. I haven't used it a lot yet, but I haven't seen any issues with references so far. Do you know if tern has advantages over tide? P.S. Looking forward for a setup guide on Indium!

 |  • Reply • Share ▸

## ALSO ON EMACS.CAFE

**M-x hello-world**

4 comments • 4 months ago •

**NoonianAtall** — (incf looking-forward-to-reading) ;)**Hierarchy.el**

13 comments • a month ago •

**Damien Cassou** — I have no clue what such data would look like but if it's a hierarchy that should be possible. I can help if you want to ...**How to find all unused functions in JS buffers**

2 comments • 3 months ago •


**NicolasPetton** — It returns t if the argument is nil, and nil otherwise.**Setting up Emacs for JavaScript (part #1)**

5 comments • 4 months ago •

**NicolasPetton** — Also, xref-js2 does a secondpass on result candidates and eliminates possible false positives usingjs2-mode's AST, ... Subscribe  Add Disqus to your siteAdd DisqusAdd  Privacy

## Emacs café

Emacs café  
nicolas@petton.fr

 NicolasPetton  
 NicolasPetton

A blog about Emacs, mostly focused on  
JavaScript development, by Nicolas Petton.