

## .add(key, value)

Add a new schema to the schemas object. As described in the constructor definition, `key` is a link prefix (`skype:`, for example), and `value` is a String to alias to another schema, or an Object with `validate` and optionally `normalize` definitions. To disable an existing rule, use `.add(key, null)`.

## .set(options)

Override default options. Missed properties will not be changed.

## License

[MIT](#)

# linkify-it



*Links recognition library with FULL unicode support. Focused on high quality link patterns detection in plain text.*

## [Demo](#)

Why it's awesome:

Full unicode support, *with astral characters!*

International domains support.

Allows rules extension & custom normalizers.

## Install

Browserification is also supported.

## Usage examples

### Example 1

```
import linkifyit from 'linkify-it';
const linkify = linkifyIt();
```

```
    .tlds(require('tlds')) // Reload with full tlds list
```

```
        true) // Add  
        .onion` domain  
        unofficial` .onion` domain
```

```
// Add 'git://')
```

```
console.log(linkify.test('Site  
github.com!')); // true
```

```
github.com!`))]; // [ {
```

11

.tlds(list, keepOld)

Checks if a match exists at the start of the string. Returns Match (see docs for `match(text)`) or null if no URL is at the start. Doesn't work with fuzzy links.

## .matchAtStart(text)

- |   |
|---|
| • <b>raw</b> - matched text                             |
| • <b>lastindex</b> - index of next char after match end |
| • <b>index</b> - offset of matched text                 |
| • <b>text</b> - normalized text                         |
| • <b>url</b> - link, generated from matched text        |

### .test(text)

Searches linkifiable pattern and returns `true` on success or `false` on fail.

### .pretest(text)

Quick check if link MAY BE can exist. Can be used to optimize more expensive `.test()` calls. Return `false` if link can not be found, `true` - if `.test()` call needed to know exactly.

### .testSchemaAt(text, name, offset)

Similar to `.test()` but checks only specific protocol tail exactly at given position. Returns length of found pattern (0 on fail).

### .match(text)

Returns Array of found link matches or null if nothing found.

Each match has:

- **schema** - link schema, can be empty for fuzzy links, or `//` for protocol-neutral links.

```
index: 5,  
lastIndex: 15,  
raw: "github.com",  
text: "github.com",  
url: "http://github.com",
```

### Example 2. Add twitter mentions handler

```
linkify.add('@', {  
  validate: function (text, pos, self) {  
    const tail = text.slice(pos);  
  
    if (!self.re.twitter) {  
      self.re.twitter = new RegExp(  
        '^([a-zA-Z0-9_])\\{1,15}(?!_)\\(?  
        =\$|' + self.re.src_ZPCc + ')'  
    );  
    }  
    if (self.re.twitter.test(tail)) {  
      // Linkifier allows punctuation  
      // chars before prefix,  
      // but we additionally disable `@`  
      // ("@@mention" is invalid)  
      if (pos >= 2 && tail[pos - 2] ===  
          '@') {  
        return false;  
      }  
    }  
  }  
});
```

<b>Protocol/rule:</b>	<b>schemas</b> is an object, where each key/value describes
<b>key</b> - link prefix (usually, protocol name with : at the end,	<b>key</b> - link prefix (for example). Linkify - it makes sure that prefix is not preceded with alphanumeric char.
<b>value</b> - rule to check tail after link prefix	<b>value</b> - rule to check tail after link prefix
<b>String</b> - just alias to existing rule	<b>String</b> - just alias to existing rule
<b>Object</b>	<b>Object</b>
<b>validate</b> - either a RegExp (start with ^, and don't include the link prefix itself), or a validator function which, given arguments	used to access the linkify object to cache data.
<b>normalize</b> - optional function to normalize text & url of matched result (for example, for twitter mentions).	matched result (for example, for twitter mentions).
<b>fuzzyLink</b>	<b>fuzzyLink</b> - recognize URLs without http(s):// head.
<b>options:</b>	<b>options:</b>
<b>fuzzyIP</b> - allow IPs in fuzzy links above. Can conflict with some texts like version numbers. Default false.	<b>fuzzyIP</b> - allow IPs in fuzzy links above. Can conflict with some texts like version numbers. Default false.
<b>fuzzyEmail</b> - recognize emails without mailto: prefix.	<b>fuzzyEmail</b> - recognize emails without mailto: prefix.
<b>Default true.</b>	<b>Default true.</b>
<b>longDash</b> .	<b>longDash</b> .

```
return tail.match(select(re.twitter))  
    .normalize(function(match) {  
        var url = match.url.replace(/\@/, '_');  
        + match.url.replace(/\@/, '_');  
    })  
});
```