



Open Source Used In dnaspaces-iot-device-marketplace-client 7.0.2

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1168867591

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please contact us at external-opensource-requests@cisco.com.

In your requests please include the following reference number 78EE117C99-1168867591

Contents

[1.1 jasmine-core 3.2.1](#)

[1.1.1 Available under license](#)

[1.2 jasmine-core 2.8.0](#)

[1.2.1 Available under license](#)

1.1 jasmine-core 3.2.1

1.1.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

Copyright (c) 2008-2018 Pivotal Labs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*/

Found in path(s):

* /opt/cola/permits/1168892818_1622104997.21/0/jasmine-core-3-2-1-gem/data-tar-gz/lib/jasmine-core/boot.js
* /opt/cola/permits/1168892818_1622104997.21/0/jasmine-core-3-2-1-gem/data-tar-gz/lib/jasmine-core/node_boot.js
* /opt/cola/permits/1168892818_1622104997.21/0/jasmine-core-3-2-1-gem/data-tar-gz/lib/jasmine-core/jasmine.js
* /opt/cola/permits/1168892818_1622104997.21/0/jasmine-core-3-2-1-gem/data-tar-gz/lib/jasmine-core/jasmine-html.js

No license file was found, but licenses were detected in source scan.

/*

json2.js
2014-02-04

Public Domain.

NO WARRANTY EXPRESSED OR IMPLIED. USE AT YOUR OWN RISK.

See <http://www.JSON.org/js.html>

This code should be minified before deployment.

See <http://javascript.crockford.com/jsmin.html>

USE YOUR OWN COPY. IT IS EXTREMELY UNWISE TO LOAD CODE FROM SERVERS YOU DO NOT CONTROL.

This file creates a global JSON object containing two methods: stringify and parse.

JSON.stringify(value, replacer, space)

value any JavaScript value, usually an object or array.

replacer an optional parameter that determines how object values are stringified for objects. It can be a function or an array of strings.

space an optional parameter that specifies the indentation of nested structures. If it is omitted, the text will be packed without extra whitespace. If it is a number, it will specify the number of spaces to indent at each level. If it is a string (such as '\t' or ' '), it contains the characters used to indent at each level.

This method produces a JSON text from a JavaScript value.

When an object value is found, if the object contains a toJSON method, its toJSON method will be called and the result will be stringified. A toJSON method does not serialize: it returns the value represented by the name/value pair that should be serialized, or undefined if nothing should be serialized. The toJSON method will be passed the key associated with the value, and this will be bound to the value

For example, this would serialize Dates as ISO strings.

```
Date.prototype.toJSON = function (key) {
  function f(n) {
    // Format integers to have at least two digits.
    return n < 10 ? '0' + n : n;
  }

  return this.getUTCFullYear() + '-' +
    f(this.getUTCMonth() + 1) + '-' +
    f(this.getUTCDate()) + 'T' +
    f(this.getUTCHours()) + ':' +
    f(this.getUTCMinutes()) + ':' +
    f(this.getUTCSeconds()) + 'Z';
};
```

You can provide an optional replacer method. It will be passed the key and value of each member, with this bound to the containing object. The value that is returned from your method will be serialized. If your method returns undefined, then the member will be excluded from the serialization.

If the replacer parameter is an array of strings, then it will be used to select the members to be serialized. It filters the results such that only members with keys listed in the replacer array are stringified.

Values that do not have JSON representations, such as undefined or functions, will not be serialized. Such values in objects will be dropped; in arrays they will be replaced with null. You can use a replacer function to replace those with JSON values. JSON.stringify(undefined) returns undefined.

The optional space parameter produces a stringification of the value that is filled with line breaks and indentation to make it easier to read.

If the space parameter is a non-empty string, then that string will be used for indentation. If the space parameter is a number, then the indentation will be that many spaces.

Example:

```
text = JSON.stringify(['e', {pluribus: 'unum'}]);  
// text is '["e",{"pluribus":"unum"}]'
```

```
text = JSON.stringify(['e', {pluribus: 'unum'}], null, '\t');  
// text is '["\t"e",\n\t{\n\t\t"pluribus": "unum"\n\t}\n]'
```

```
text = JSON.stringify([new Date()], function (key, value) {  
    return this[key] instanceof Date ?  
        'Date(' + this[key] + ')' : value;  
});  
// text is '["Date(---current time---)"]'
```

JSON.parse(text, reviver)

This method parses a JSON text to produce an object or array.
It can throw a SyntaxError exception.

The optional reviver parameter is a function that can filter and transform the results. It receives each of the keys and values, and its return value is used instead of the original value. If it returns what it received, then the structure is not modified. If it returns undefined then the member is deleted.

Example:

```
// Parse the text. Values that look like ISO date strings will  
// be converted to Date objects.
```

```
myData = JSON.parse(text, function (key, value) {  
    var a;  
    if (typeof value === 'string') {  
        a =  
/^(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2}):(\d{2})(?:\.\d*)?Z$/.exec(value);  
        if (a) {  
            return new Date(Date.UTC(+a[1], +a[2] - 1, +a[3], +a[4],  
                +a[5], +a[6]));  
        }  
    }  
    return value;  
});
```

```
myData = JSON.parse('["Date(09/09/2001)"]', function (key, value) {  
    var d;  
    if (typeof value === 'string' &&
```

```
        value.slice(0, 5) === 'Date(' &&
        value.slice(-1) === ')') {
    d = new Date(value.slice(5, -1));
    if (d) {
        return d;
    }
}
return value;
});
```

This is a reference implementation. You are free to copy, modify, or redistribute.

*/

Found in path(s):

* /opt/cola/permits/1168892818_1622104997.21/0/jasmine-core-3-2-1-gem/data-tar-gz/lib/jasmine-core/json2.js

1.2 jasmine-core 2.8.0

1.2.1 Available under license :

Copyright (c) 2008-2017 Pivotal Labs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2021 Cisco Systems, Inc. All rights reserved.