



Node for Max API

Version 9.1.1-rev.0

Cycling '74

cycling74.com

Contents

Node for Max API	3
function addHandler	5
function addHandlers	6
type Anything	7
function getDict	8
type JSONArray	9
type JSONObject	10
type JSONPrimitive	11
type JSONValue	12
enum MAX_ENV	13
type MaxFunctionHandler	14
type MaxFunctionSelector	15
enum MESSAGE_TYPES	16
function outlet	17
function outletBang	18
enum POST_LEVELS	19
function post	20
function removeHandler	21
function removeHandlers	22
function setDict	23
function updateDict	24
Credits	25

Node for Max API

Enums	3
Functions	3
Type Aliases	4

Reference for the API exposed as "max-api" when running NodeJS applications using the `[node.script]` object. This API can be required using

```
const maxAPI = require("max-api");
```

Enums

Item	Description
<code>MAX_ENV</code>	Enum options for the value Node For Max will set on <code>process.env.MAX_ENV</code>
<code>MESSAGE_TYPES</code>	Predefined generic MaxFunctionSelector types
<code>POST_LEVELS</code>	Log Levels used in <code>maxAPI.post</code>

Functions

Item	Description
<code>addHandler</code>	Register a single handler

addHandlers	Register handlers
getDict	Get the value of a dict object
outlet	Outlet any values
outletBang	Outlet a Bang
post	Post to the Max console. Setting the last argument to a value of maxAPI.POST_LEVELS allows control of the log level
removeHandler	Remove a single handler
removeHandlers	Remove handlers
setDict	Set the value of a dict object
updateDict	Partially update the value of a dict object at a given path

Type Aliases

Item	Description
Anything	
JSONArray	
JSONObject	
JSONPrimitive	
JsonValue	
MaxFunctionHandler	
MaxFunctionSelector	

function addHandler

Register a single handler

```
export function addHandler(selector: MaxFunctionSelector, handler:  
MaxFunctionHandler): void;
```

Name	Type	Description
selector	MaxFunctionSelector	
handler	MaxFunctionHandler	

function addHandlers

Register handlers

```
export function addHandlers(handlers: Record<MaxFunctionSelector,  
MaxFunctionHandler>): void;
```

Name	Type	Description
handlers	Record< MaxFunctionSelector , MaxFunctionHandler >	

type Anything

```
export type Anything = string | number | Array<string | number> |
JSONObject | JSONArray;
```

function getDict

Get the value of a dict object

```
export function getDict(id: string): Promise<JSONObject>;
```

Name	Type	Description
id	string	
Return Value	Promise<JSONObject>	

type JSONArray

```
export type JSONArray = Array<JSONValue>;
```

type JSONObject

```
export type JSONObject = { [key: string]: JSONValue | undefined };
```

type JSONPrimitive

```
export type JSONPrimitive = string | number | boolean | null;
```

type JSONValue

```
export type JSONValue = JSONPrimitive | JSONArray | JSONObject;
```

enum MAX_ENV

Enum options for the value Node For Max will set on process.env.MAX_ENV

Members

Member	Value	Description
MAX	"max"	node.script running from within Max
MAX_FOR_LIVE	"maxforlive"	node.script running from within Max For Live
STANDALONE	"max:standalone"	node.script running from within a standalone application

type MaxFunctionHandler

```
export type MaxFunctionHandler = (...args: any[]) => any;
```

type MaxFunctionSelector

```
export type MaxFunctionSelector = MESSAGE_TYPES | string;
```

enum MESSAGE_TYPES

Predefined generic MaxFunctionSelector types

Members

Member	Value	Description
ALL	"all"	Generic Type for *all* kinds of messages
BANG	"bang"	Bang message type
DICT	"dict"	Dictionary message type
LIST	"list"	List message type
NUMBER	"number"	Number message type

function outlet

Outlet any values

```
export function outlet(...args: JSONValue[]): Promise<void>;
```

Name	Type	Description
args	JSONValue[]	
Return Value	Promise<void>	

function outletBang

Outlet a Bang

```
export function outletBang(): Promise<void>;
```

Name	Type	Description
Return Value	Promise<void>	

enum POST_LEVELS

Log Levels used in maxAPI.post

Members

Member	Value	Description
ERROR	"error"	error level messages
INFO	"info"	info level messages
WARN	"warn"	warn level messages

function post

Post to the Max console. Setting the last argument to a value of maxAPI.POST_LEVELS allows control of the log level

```
export function post(...args: Array<Anything | POST_LEVELS>):  
Promise<void>;
```

Name	Type	Description
args	Array<Anything POST_LEVELS>	
Return Value	Promise<void>	

function removeHandler

Remove a single handler

```
export function removeHandler(selector: MaxFunctionSelector, handler:  
MaxFunctionHandler): void;
```

Name	Type	Description
selector	MaxFunctionSelector	
handler	MaxFunctionHandler	

function removeHandlers

Remove handlers

```
export function removeHandlers(selector: MaxFunctionSelector): void;
```

Name	Type	Description
selector	MaxFunctionSelector	

function setDict

Set the value of a dict object

```
export function setDict(id: string, dict: JSONObject): Promise<JSONObject>;
```

Name	Type	Description
id	string	
dict	JSONObject	
Return Value	Promise<JSONObject>	

function updateDict

Partially update the value of a dict object at a given path

```
export function updateDict(id: string, updatePath: string, updateValue: JSONValue): Promise<JSONObject>;
```

Name	Type	Description
id	string	
updatePath	string	
updateValue	JSONValue	
Return Value	Promise<JSONObject>	

Credits

Cycling '74
440 N Barranca Ave #4074
Covina, CA 91723 USA

Copyright 2025 Cycling '74. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Cycling '74. Every effort has been made to ensure that the information in this document is accurate. Cycling '74 assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Except as permitted by such license, no part of this publication may be reproduced, edited, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of Cycling '74.

Contact Support: <https://cycling74.com/support/contact>