```
/**
 * Copyright JS Foundation and other contributors, http://js.foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 **/

// The `https` setting requires the `fs` module. Uncomment the following
// to make it available:
//var fs = require("fs");

module.exports = {
    // the tcp port that the Node-RED web server is listening on
    uiPort: process.env.PORT || 1880,

    // By default, the Node-RED UI accepts connections on all IPv4 interfaces.
    // To listen on all IPv6 addresses, set uiHost to "::",
    // The following property can be used to listen on a specific interface. For
    // example, the following would only allow connections from the local
machine.
    //uiHost: "127.0.0.1",

    // Retry time in milliseconds for MQTT connections
    mqttReconnectTime: 15000,

    // Retry time in milliseconds for Serial port connections
    serialReconnectTime: 15000,

    // Retry time in milliseconds for TCP socket connections
    //socketReconnectTime: 10000,

    // Timeout in milliseconds for TCP server socket connections
    //  defaults to no timeout
    //socketTimeout: 120000,

    // Maximum number of messages to wait in queue while attempting to connect
to TCP socket
    //  defaults to 1000
    //tcpMsgQueueSize: 2000,

    // Timeout in milliseconds for HTTP request connections
    //  defaults to 120 seconds
    //httpRequestTimeout: 120000,

    // The maximum length, in characters, of any message sent to the debug
```

```
sidebar tab
    debugMaxLength: 1000,

    // The maximum number of messages nodes will buffer internally as part of
their
    // operation. This applies across a range of nodes that operate on message
sequences.
    //  defaults to no limit. A value of 0 also means no limit is applied.
    //nodeMaxMessageBufferLength: 0,

    // To disable the option for using local files for storing keys and
certificates in the TLS configuration
    //  node, set this to true
    //tlsConfigDisableLocalFiles: true,

    // Colourise the console output of the debug node
    //debugUseColors: true,

    // The file containing the flows. If not set, it defaults to
flows_<hostname>.json
    //flowFile: 'flows.json',

    // To enabled pretty-printing of the flow within the flow file, set the
following
    //  property to true:
    //flowFilePretty: true,

    // By default, credentials are encrypted in storage using a generated key.
To
    // specify your own secret, set the following property.
    // If you want to disable encryption of credentials, set this property to
false.
    // Note: once you set this property, do not change it - doing so will
prevent
    // node-red from being able to decrypt your existing credentials and they
will be
    // lost.
    //credentialSecret: "a-secret-key",

    // By default, all user data is stored in a directory called `.node-red`
under
    // the user's home directory. To use a different location, the following
    // property can be used
    //userDir: '/home/nol/.node-red/',

    // Node-RED scans the `nodes` directory in the userDir to find local node
files.
    // The following property can be used to specify an additional directory to
scan.
    //nodesDir: '/home/nol/.node-red/nodes',

    // By default, the Node-RED UI is available at http://localhost:1880/
    // The following property can be used to specify a different root path.
    // If set to false, this is disabled.
```

```
    //httpAdminRoot: '/admin',

    // Some nodes, such as HTTP In, can be used to listen for incoming http
requests.
    // By default, these are served relative to '/'. The following property
    // can be used to specifiy a different root path. If set to false, this is
    // disabled.
    //httpNodeRoot: '/red-nodes',

    // The following property can be used in place of 'httpAdminRoot' and
'httpNodeRoot',
    // to apply the same root to both parts.
    //httpRoot: '/red',

    // When httpAdminRoot is used to move the UI to a different root path, the
    // following property can be used to identify a directory of static content
    // that should be served at http://localhost:1880/.
    //httpStatic: '/home/nol/node-red-static/',

    // The maximum size of HTTP request that will be accepted by the runtime
api.
    // Default: 5mb
    //apiMaxLength: '5mb',

    // If you installed the optional node-red-dashboard you can set it's path
    // relative to httpRoot
    //ui: { path: "ui" },

    // Securing Node-RED
    // -----------------
    // To password protect the Node-RED editor and admin API, the following
    // property can be used. See http://nodered.org/docs/security.html for
details.
    //adminAuth: {
    //    type: "credentials",
    //    users: [{
    //        username: "admin",
    //        password:
"$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN.",
    //        permissions: "*"
    //    }]
    //},

    // To password protect the node-defined HTTP endpoints (httpNodeRoot), or
    // the static content (httpStatic), the following properties can be used.
    // The pass field is a bcrypt hash of the password.
    // See http://nodered.org/docs/security.html#generating-the-password-hash
    //httpNodeAuth:
{user:"user",pass:"$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN."
},
    //httpStaticAuth:
{user:"user",pass:"$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN."
},
```

```
    // The following property can be used to enable HTTPS
    // See
http://nodejs.org/api/https.html#https_https_createserver_options_requestlistene
r
    // for details on its contents.
    // See the comment at the top of this file on how to load the `fs` module
used by
    // this setting.
    //
    //https: {
    //    key: fs.readFileSync('privatekey.pem'),
    //    cert: fs.readFileSync('certificate.pem')
    //},

    // The following property can be used to cause insecure HTTP connections to
    // be redirected to HTTPS.
    //requireHttps: true,

    // The following property can be used to disable the editor. The admin API
    // is not affected by this option. To disable both the editor and the admin
    // API, use either the httpRoot or httpAdminRoot properties
    //disableEditor: false,

    // The following property can be used to configure cross-origin resource
sharing
    // in the HTTP nodes.
    // See https://github.com/troygoode/node-cors#configuration-options for
    // details on its contents. The following is a basic permissive set of
options:
    //httpNodeCors: {
    //    origin: "*",
    //    methods: "GET,PUT,POST,DELETE"
    //},

    // If you need to set an http proxy please set an environment variable
    // called http_proxy (or HTTP_PROXY) outside of Node-RED in the operating
system.
    // For example - http_proxy=http://myproxy.com:8080
    // (Setting it here will have no effect)
    // You may also specify no_proxy (or NO_PROXY) to supply a comma separated
    // list of domains to not proxy, eg - no_proxy=.acme.co,.acme.co.uk

    // The following property can be used to add a custom middleware function
    // in front of all http in nodes. This allows custom authentication to be
    // applied to all http in nodes, or any other sort of common request
processing.
    //httpNodeMiddleware: function(req,res,next) {
    //    // Handle/reject the request, or pass it on to the http in node by
calling next();
    //    // Optionally skip our rawBodyParser by setting this to true;
    //    //req.skipRawBodyParser = true;
    //    next();
    //},
```

```
    // The following property can be used to pass custom options to the
Express.js
    // server used by Node-RED. For a full list of available options, refer
    // to http://expressjs.com/en/api.html#app.settings.table
    //httpServerOptions: { },

    // The following property can be used to verify websocket connection
attempts.
    // This allows, for example, the HTTP request headers to be checked to
ensure
    // they include valid authentication information.
    //webSocketNodeVerifyClient: function(info) {
    //    // 'info' has three properties:
    //    //   - origin : the value in the Origin header
    //    //   - req : the HTTP request
    //    //   - secure : true if req.connection.authorized or
req.connection.encrypted is set
    //    //
    //    // The function should return true if the connection should be
accepted, false otherwise.
    //    //
    //    // Alternatively, if this function is defined to accept a second
argument, callback,
    //    // it can be used to verify the client asynchronously.
    //    // The callback takes three arguments:
    //    //   - result : boolean, whether to accept the connection or not
    //    //   - code : if result is false, the HTTP error status to return
    //    //   - reason: if result is false, the HTTP reason string to return
    //},

    // The following property can be used to seed Global Context with predefined
    // values. This allows extra node modules to be made available with the
    // Function node.
    // For example,
    //    functionGlobalContext: { os:require('os') }
    // can be accessed in a function block as:
    //    global.get("os")
    functionGlobalContext: {
        // os:require('os'),
        // jfive:require("johnny-five"),
        // j5board:require("johnny-five").Board({repl:false})
        Enocean:require('enocean-js')
    },
    // `global.keys()` returns a list of all properties set in global context.
    // This allows them to be displayed in the Context Sidebar within the
editor.
    // In some circumstances it is not desirable to expose them to the editor.
The
    // following property can be used to hide any property set in
`functionGlobalContext`
    // from being list by `global.keys()`.
    // By default, the property is set to false to avoid accidental exposure of
    // their values. Setting this to true will cause the keys to be listed.
    exportGlobalContextKeys: false,
```

```
    // Context Storage
    // The following property can be used to enable context storage. The
configuration
    // provided here will enable file-based context that flushes to disk every
30 seconds.
    // Refer to the documentation for further options:
https://nodered.org/docs/api/context/
    //
    contextStorage: {
        default: {
            module:"localfilesystem"
        },
    },

    // The following property can be used to order the categories in the editor
    // palette. If a node's category is not in the list, the category will get
    // added to the end of the palette.
    // If not set, the following default order is used:
    //paletteCategories: ['subflows', 'input', 'output', 'function', 'social',
'mobile', 'storage', 'analysis', 'advanced'],

    // Configure the logging output
    logging: {
        // Only console logging is currently supported
        console: {
            // Level of logging to be recorded. Options are:
            // fatal - only those errors which make the application unusable
should be recorded
            // error - record errors which are deemed fatal for a particular
request + fatal errors
            // warn - record problems which are non fatal + errors + fatal
errors
            // info - record information about the general running of the
application + warn + error + fatal errors
            // debug - record information which is more verbose than info + info
+ warn + error + fatal errors
            // trace - record very detailed logging + debug + info + warn +
error + fatal errors
            // off - turn off all logging (doesn't affect metrics or audit)
            level: "info",
            // Whether or not to include metric events in the log output
            metrics: false,
            // Whether or not to include audit events in the log output
            audit: false
        }
    },

    // Customising the editor
    editorTheme: {
        projects: {
            // To enable the Projects feature, set this value to true
            enabled: false
```

```
            }
        }
    }
```