Jscrambler 3.x to 5.1

Migration Manual

Jscrambler Development Team

(+351) 227 663 702 contact@jscrambler.com jscrambler.com

Table of Contents

Why should you change to 5.1?

Jscrambler's newest version offers a series of new and improved features to help you achieve the best degree of protection while keeping your code performant and compliant with every Javascript framework.

New & Improved Transformations

We have worked hard on keeping our obfuscation and protection technology up-to-date and in providing more and more powerful transformations:

Control Flow Flattening

A very powerful transformation which obfuscates the control flow of your code, making it really hard to debug and understand.

Improved Self Defending (Anti-debugging & Anti-tampering)

With harder anti-debugging capabilities and the possibility of tolerating (benign) native functions poisoning and transformations applied by other minifiers.

Improved Locks

- Browser and OS locks are now decoupled
- Possibility of locking the domain to IPs or file:/// (via Domain Lock)
- Possibility of adding a start date on the Date Lock

Increased and better string transformations

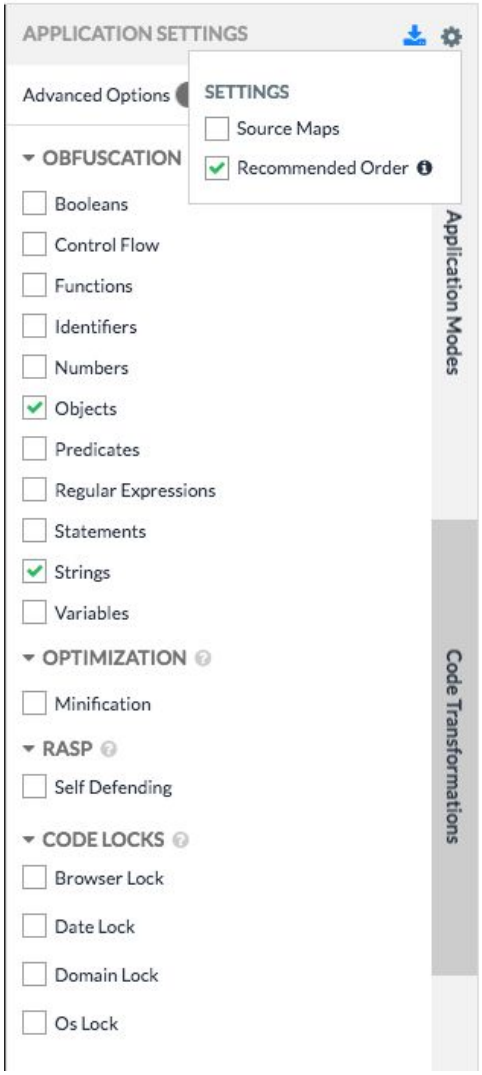
We have more string transformations at your disposal:
String Encoding: Converts your strings into an encoded representation making them harder to read.
String Concealing: Conceals all the strings in your code so that they're no longer visible.

Synergies across transformations

Some transformations now work together to provide a higher degree of obfuscation and protection to your code.

Targets

We've also introduced the concept of targets. Instead of selecting which transformations you want to apply, you can choose which type of Javascript primitive types/expressions you want to target. If your code is heavily string and object reliant, choose the Strings and Objects targets and we will choose suitable transformations.



Fresh Improvements

Randomization Seed

Gives you the ability of completely reproducing the random behaviours of a previous transformation (to be used in debugging/benchmarking).

Source Maps

Gives you the possibility of debugging obfuscated code in its original form.

More Flexible Code Annotations

3.x only offered the possibility of ignoring the protection on certain code-blocks. On 5.1 and above we take on a whitelisting approach instead of blacklisting and have implemented a standard syntax for annotations: JSDoc. Now you can choose specific transformations with custom options to only apply protection where you most need it and:

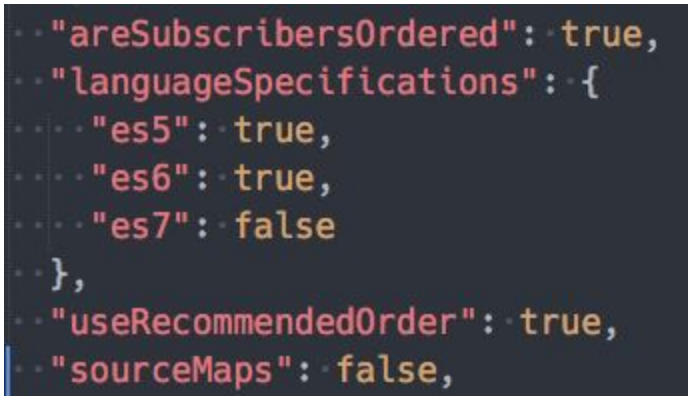
- enable/disable transformations globally or in specific statements/blocks
- define transformation aliases with your custom settings
- change the transformations order and apply them more than once
- Enable transformations without knowing their names by selecting well known JavaScript primitive types/expressions you want to target (e.g. functions, objects, strings)

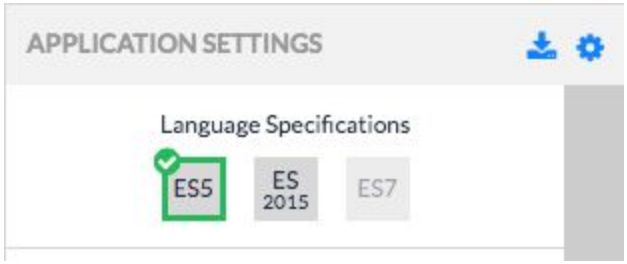
Choose Order

By using our "areSubscribersOrdered" option (only available via API), you can choose the order in which transformations are applied to take full advantage of synergies between transformations (e.g., using String Splitting and Control-Flow Flattening). If this option is not set, the order will be random. You can also enable the "recommendedOrder" option (available either in the UI or via API) to make use of our specially crafted order, which is adequate for most use-cases.

ES2015 (aka ES6)

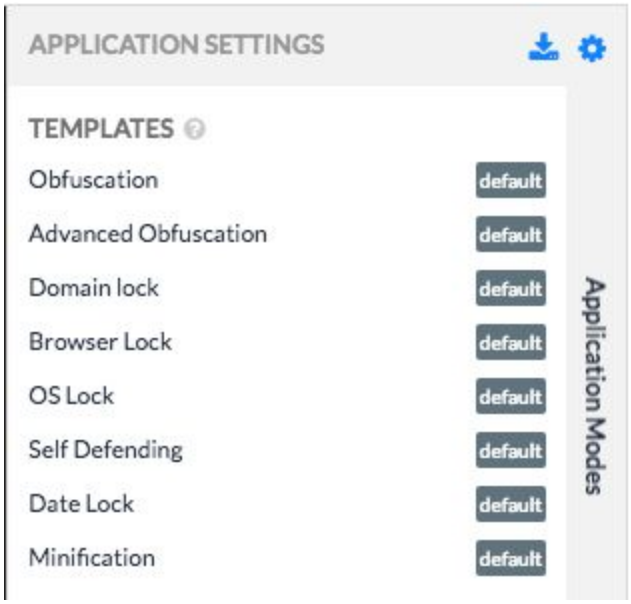
In version 5.1 Jscrambler is compliant with the ES2015 spec. You can select it either via UI or enabling the option on your JSON configuration.





Better Default templates

We've revised our default templates and made them more suitable to common use-cases. These are balanced and effective *cocktails* to protect your code out-of-the-box.



Enhanced UI & CLI

Node CLI

We have a new and improved Node.js CLI. It allows for customization of the behavior of interactions with Jscrambler's API, the most notable option being the --werror flag. This option which will consider warnings thrown while protecting as errors, ensuring that you'll only protect your code when it is 100% warning-free.

Refreshed UI & App Dashboard

Jscrambler 5.1 introduces the concept of Apps. Apps are containers of each of the applications you need to protect with Jscrambler. Each app typically has a specific source code and a particular set of transformations have to be chosen to protect it. On our revamped Dashboard it is now possible to have an overview of all of your applications.

You will also find the Demo and Playground apps, which are default apps that will always be present. On the Demo app you will be able to walk through the steps of successfully uploading your code, protecting it and downloading the result. Give it a

shot if you never used our most recent interface. The Playground app is a sandbox where you can try out most of Jscrambler's transformations (even the ones not available on your plan) on a pre-defined input file.

5.1 also comes with a refreshed UI that offers a lot more capabilities in selecting your desired protection while being able to view the protection result.

You can find more info about how to use our new UI in our getting started guide.

Enterprise Made Better

Our Enterprise solution also received major improvements in terms of performance and functionality. Our remote support was completely updated to provide a quick and effective support experience. We've also made some major improvements in terms of how you can manage your Enterprise Jscrambler solution. If you're already an Enterprise customer get in touch with your account manager for more information. If you want more information on our Enterprise plan let us know by filling the form on the bottom of the Enterprise page or sending an email to security@jscrambler.com

Migrating from 3.x to 5.1

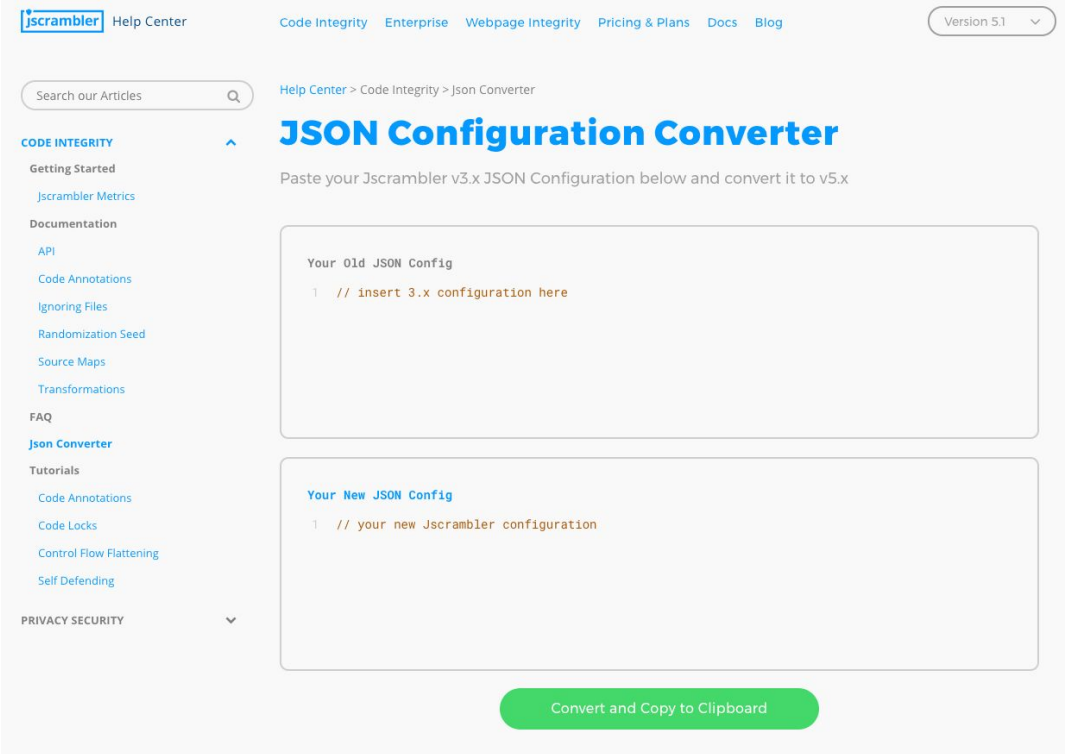
Converting your JSON configuration

What will you need?

- Jscrambler 3.x JSON config file (Appendix 1)
- The ID from an application on Jscrambler 5.1

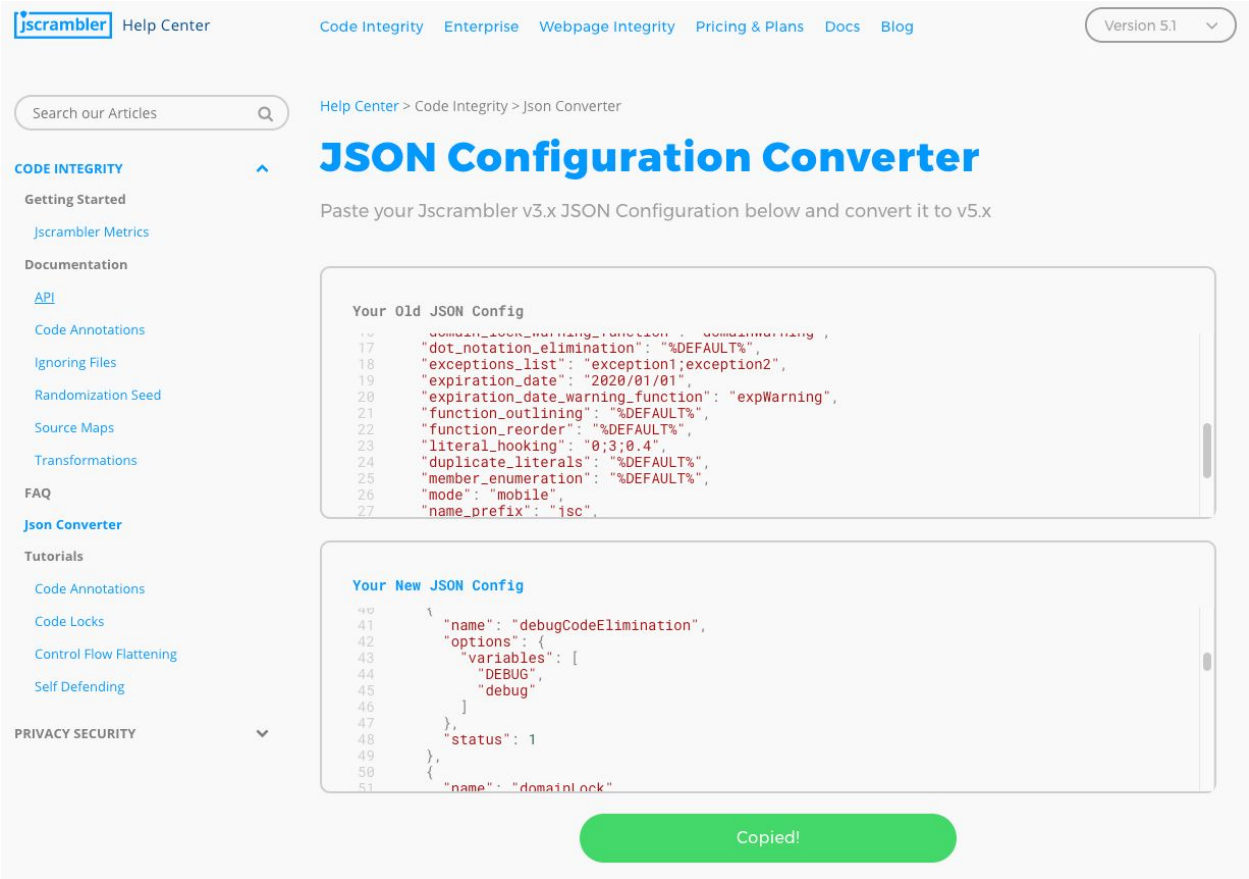
API conversion

- Follow this link and you will be presented with a configuration conversion utility
- Place your Jscrambler 3.x *JSON* config on the upper textarea



(+351) 227 663 702 contact@jscrambler.com jscrambler.com

- Hit the convert button
- The generated configuration was copied to your clipboard, now paste it on a text editor of your choice



- Replace the applicationId, <APPID>, by the ID of your application

The final config should have a similar structure to the one on Appendix 2.
Use this config to protect with Jscrambler 5.1. Please notice that to do so you also need to install one of our new clients, please follow to the API Client's Documentation for more details on this.

Migrating to the new Web Interface

If you intend to use the new Web Interface please consider using one of our default templates, they were revamped to comply with the new features providing improved protection levels.

Alternatively, considering you were using the old Web Interface, if you want to keep a similar config to the one you were using there are some ways to do so.

- Create a config with the old parameters and convert it using the provided tool. In this case, you will need to copy the output to the UI manually.
- Follow the mapping we made available on the current document and select the equivalent set of transformations on the new version.

If you've created your own custom templates on version 3.x don't worry because they have been automatically migrated to the new Jscrambler 5.1 already.

Transformations conversion mapping

Currently there are no mappings to the transformations *Dictionary Compression* and *Member Enumeration* being ignored by the conversion tool. The mapping provided with our tool was tested against some real world samples and tweaked to provide an equivalent or superior protection level.
The transformations options available on Jscrambler 5.1 are not necessarily the same. For further details on the options for specific transformations please follow the docs.
The Appendix 3 displays the current transformation mapping.

Getting Started

Once you've converted all you need you should be able to start using Jscrambler 5.1. You can find more info about how to use our new UI in our getting started guide. If you're more inclined into using our API, you can find more info about our customers and their usage here.

Appendices

Appendix 1 - Jscrambler 3.x sample configuration

```
{
  "keys": {
    "accessKey": "ACCESS_KEY",
    "secretKey": "SECRET_KEY"
  },
  "filesDest": "./obf",
  "params": {
    "asserts_elimination": "assert;equals",
    "browser_os_lock": "firefox",
    "constant_folding": "%DEFAULT%",
    "dead_code": "%DEFAULT%",
    "dead_code_elimination": "%DEFAULT%",
    "debugging_code_elimination": "DEBUG;debug",
    "dictionary_compression": "%DEFAULT%",
    "domain_lock": "example.com;www.jscrambler.com",
    "domain_lock_warning_function": "domainWarning",
    "dot_notation_elimination": "%DEFAULT%",
    "exceptions_list": "exception1;exception2",
    "expiration_date": "2020/01/01",
    "expiration_date_warning_function": "expWarning",
    "function_outlining": "%DEFAULT%",
    "function_reorder": "%DEFAULT%",
    "ignore_files": "file1;file2",
    "literal_hooking": "0;3;0.4",
    "duplicate_literals": "%DEFAULT%",
    "member_enumeration": "%DEFAULT%",
    "mode": "mobile",
    "name_prefix": "jsc",
    "rename_all": "%DEFAULT%",
    "rename_local": "%DEFAULT%",
    "self_defending": "%DEFAULT%",
    "string_splitting": "0.3;0.4",
    "whitespace": "%DEFAULT%"
  }
}
```

Appendix 2 - Jscrambler 5.1 sample configuration

```
{
 "keys": {
   "accessKey": "ACCESS_KEY",
   "secretKey": "SECRET_KEY"
 },
 "filesDest": "./obf",
 "applicationId": "<APP_ID>",
 "params": [{
     "name": "identifiersRenaming",
     "options": {
       "mode": "UNSAFE",
       "excludeList": [
         "exception1",
         "exception2"
       ],
       "namePrefix": "jsc"
     },
     "status": 1
   },
   {
     "name": "whitespaceRemoval",
     "status": 1
   }
 ],
 "areSubscribersOrdered": true,
 "applicationTypes": {
   "webBrowserApp": false,
   "desktopApp": false,
   "serverApp": false,
   "hybridMobileApp": true,
   "javascriptNativeApp": false,
   "html5GameApp": false
 },
 "languageSpecifications": {
   "es5": true,
   "es6": false,
   "es7": false
 },
 "useRecommendedOrder": false
}
```

Appendix 3 - Transformation mappings

Transformation	3.x notation	5.x notation	Notes
Asserts Elimination	asserts_elimination	assertionsRemoval	
Browser Lock	browser_os_lock	browserLock	On 3.x the option selected dictates if it is browser or OS related
Os Lock	browser_os_lock	osLock	
Constant Folding	constant_folding	constantFolding	
Dead Code Injection	dead_code	deadCodeInjection	
Dead Code Elimination	dead_code_elimination	deadCodeElimination	
Debug Code Elimination	debugging_code_elimination	debugCodeElimination	
Dot To Bracket Notation + Duplicate Literals Removal	dot_notation_elimination	dotToBracketNotation + duplicateLiteralsRemoval	Combining those two transformations we get a similar output
Date Lock	expiration_date	dateLock	
Function Outlining	function_outlining	functionOutlining	
Function Reorder	function_reorder	functionReordering	
Boolean To Anything + Number To String + Char To Ternary Operator	literal_hooking	booleanToAnything + numberToString + charToTernaryOperator	Combining those three transformations we get a similar output
Duplicate Literals Removal	duplicate_literals	duplicateLiteralsRemoval	

Identifiers Renaming	`rename_all`	`identifiersRenaming`	`rename_all` and `rename_local` were merged into one being the mode SAFEST equivalent to `rename_local`
Identifiers Renaming	`rename_local`	`identifiersRenaming`	
Self Defending	`self_defending`	`selfDefending`	
String Splitting	`string_splitting`	`stringSplitting`	
Whitespace Removal	`whitespace`	`whitespaceRemoval`	