

# SRBInfoAndSupportJS

---



## Motivation

Our product installations are slowly piling up and support is becoming increasingly difficult. It is of course important for us to be able to identify errors quickly, simplify analyzes and be able to react quickly to problems.

For this reason I developed the JS Library SRBInfoAndSupport.js. This gives us the opportunity to evaluate important information quickly and easily.

### The most important information for error identification includes:

- Browser designation ( version )
- Screen size
- URL of the application ( incl. port, hostname, path )
- Device Zoom
- Version of our product (last Git commit)
- What data is in the local storage
- Screenshot of the application
- ...

The most common application of the library will probably be to display a button that shows the user the important information mentioned above. This is very easy to do with the Lib. Simply integrate it into the project and a dialog is presented to the user with the following function call:

```
SRBInfoAndSupport.showSupportDialog("My dialog title", {  
  captureScreenshot: true,  
});
```

This call results in:

The screenshot shows a dialog box with a title bar 'My dialog title'. It contains three sections: 'General', 'Local Storage', and 'Attachments'. The 'General' section lists various system and application details. The 'Local Storage' section shows a single entry. The 'Attachments' section is empty. At the bottom right, there are three buttons: 'Send via Email', 'Copy to clipboard', and 'OK'.

General	
ProductName	
Version	
CommitHash	
Hostname	cats.srb.at
Port	
Protocol	:https
FullURL	https://cats.srb.at/sap/bc/ui5_ui5/sap/zsrb_cats/?sap-client=001&sap-language=DE#/Startpage
Navigator	AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 (Windows NT 10.0; Win64; x64) 5.0
DeviceZoom	100
ScreenSize	1152x1920

Local Storage	
dnsaWhatsNewVersion	2.2.0

Attachments

Keine Daten

Send via Email Copy to clipboard OK

As can be seen in the screenshot, the users/developers now have the option of sending the data via email or copying it to the clipboard.

Furthermore, it is mandatory for legal reasons to display licenses.

Both our product licenses and licenses for the libraries we use (open source libs etc.)

This is also covered by SRBInfoAndSupport.js. Using the following function call...

```
SRBInfoAndSupport.showOverviewDialog("Title");
```

...a dialog is generated that displays all important information:

Title

My testing app

Frontend version number: 1.1.1

Commit hash: asf82oap

Licenses

MyLicense

ABCABC

OK

Important technical background:

All displayed fixed data is loaded dynamically from configuration files. This has a significant advantage when making the library dynamic. For example, in the case of licenses, a JSON file containing the licenses is loaded from the configuration. If it is necessary to display a new license due to an extension of the application, this can simply be added to the configuration.

Here's an example:

```
1 {
2   · "version": "1.0.167",
3   · "srbVersionInfo": {
4     · "productName": "SRB Workspace
5     · "latestCommitHash": "819ff36"
6     · "copyright": "© Copyright 202
7     · "licenses": {
8       · "MyNewLicense": "",
9       · "html2canvas": "Copyright {
10    · }
11  · },
```

## Getting started

**Initialize the SRBJSLogger via the NPM CDN:**

For initializing the Logger via CDN, simply copy and paste the following in the index file of the HTML5 app:

```
<script src="https://npmcdn.com/srbinfoandsupportjs/src/SRBInfoAndSupport.js">
</script>
```

Or use the minified version

```
<script
src="https://npmcdn.com/srbinfoandsupportjs/src/SRBInfoAndSupport.min.js">
</script>
```

### Initialize SRBInfoAndSupportJS via downloading the latest version:

Download the latest minified version from [here](#). Place that file into a directory of the app's folder structure, where it can access that file. For UI5 apps, using the index.html bootstrapping, instantiate the Logger like above via

```
<script src="/path/to/SRBInfoAndSupport.min.js"></script>
```

For UI5 apps, not using the index.html startup logic, paste the following in the resources section of the manifest file:

```
"resources": { "js": [ { "uri": "path/to/SRBInfoAndSupport.min.js" } ] }
```

Initialize the Logger temporarily via the browsers console. Just copy, paste and execute the following snippet in the browsers console:

```
var script = document.createElement("script");
script.type = "text/javascript";
script.src =
"https://npmcdn.com/srbinfoandsupportjs/src/SRBInfoAndSupport.min.js";
script.addEventListener("load", function () {
  SRBInfoAndSupport.init({
    appname: "My testing app",
    version: "1.1.1",
    licenses: { MyLicense: "ABCABC" },
    copyright: "My version of the copyright",
    latestCommitHash: "asf82oap",
  });
});

document.head.appendChild(script);
```

## Documentation

The documentation on this page is just an overview of the most relevant functions and features. Consider reading the [JSDoc page!](#)

### The init function

```
SRBInfoAndSupport.init({
  appname: "My testing app",
  version: "1.1.1",
  licenses: { MyLicense: "ABCABC" },
  copyright: "My version of the copyright",
  latestCommitHash: "asf82oap",
});
```

or, if using the manifest properties instead:

```
SRBInfoAndSupport.init();
```

## Parameters

`init(config[{}])`

- **config** - optional

Configuration object for the init function. If provided, the following config object has to be used:

```
{
  "appname": "",
  "version": "",
  "licenses": {
    "MyLicense": ""
  },
  "copyright": "",
  "latestCommitHash": "asf82oap"
}
```

If this configuration object is not provided, the library is searching for manifest properties of the UI5 app, that is using the library.

Example manifest.json configuration:

```
{
  "_version": "1.0.162",
  "_srbVersionInfo": {
    "productName": "My Product",
    "latestCommitHash": "3704aa5",
    "copyright": "@ Copyright 2022 SRB Consulting Team GmbH. All Rights Reserved.",
    "licenses": {
      "My Product License": "",
      "html2canvas": "Copyright (c) Microsoft Corporation. Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted. THE SOFTWARE IS PROVIDED 'AS IS' AND THE
```

```
AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE  
AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL  
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR  
PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS  
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF  
THIS SOFTWARE."
```

```
    }  
  },  
  ...  
}
```

---

## The support dialog function

```
SRBInfoAndSupport.showSupportDialog("My dialog title", {  
  captureScreenshot: true,  
});
```

### Parameters

`showSupportDialog(title[String], config[{}])`

- **title** - optional

The shown title of the support dialog

- **config** - optional

Configuration object, configuring the support dialog.

- **captureScreenshot** true|false

Indicates if a screenshot of the app shall be captured.

For this, a external library `html2canvas` is necessary. This library is not shipped with the `SRBInfoAndSupportJS`. See <https://github.com/niklasvh/html2canvas> for how to initialize the library.

Quick and dirty example that is initializing `html2canvas`. You can use it in the browser console:

```
var script = document.createElement("script");  
script.type = "text/javascript";  
script.src =  
  "https://github.com/niklasvh/html2canvas/releases/download/v1.4.1/html2canvas.min.js";  
document.head.appendChild(script);
```

Example:

```
{
  "captureScreenshot": true
}
```

## Results in

My dialog title

### General

ProductName	
Version	
CommitHash	
Hostname	cats.srb.at
Port	
Protocol	:https
FullURL	https://cats.srb.at/sap/bc/ui5_ui5/sap/zsrb_cats/?sap-client=001&sap-language=DE#/Startpage
Navigator	AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 (Windows NT 10.0; Win64; x64) 5.0
DeviceZoom	100
Screensize	1152x1920

### Local Storage

dnsaWhatsNewVersion	2.2.0
---------------------	-------

### Attachments

Keine Daten	
-------------	--

Send via Email

Copy to clipboard

OK

The overview dialog function

```
SRBInfoAndSupport.showOverviewDialog("Title");
```

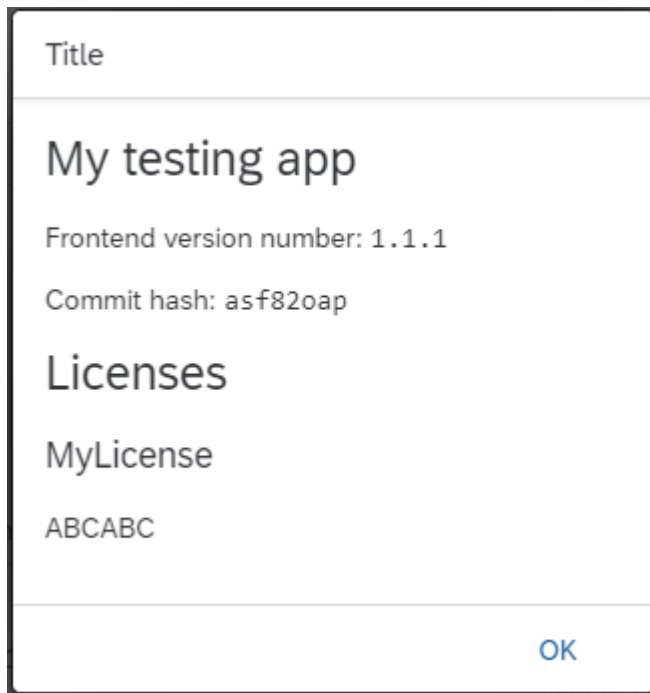
## Parameters

`showOverviewDialog(title[String])`

- **title**

The title of the info dialog

## Results in



## The initializing hidden info

This function initializes a hidden feature. The feature is open the overview dialog or the support dialog, or both. The magic thing is that the dialogs are shown only after clicking on a specific DOM element a specific number of times in a specific time range. The number of clicks is **10** in a time range of **2000** milliseconds.

The DOM element that has to be clicked can be forwarded to the function via the parameter **domElementId**. What will be showing is set via **overviewDialog** and/or **supportDialog**.

```
SRBInfoAndSupport.initHiddenInfo({
  overviewDialog: true,
  supportDialog: false,
  domElementId: "__component0---Startpage--filterSearch",
});
```

## Parameters

**initHiddenInfo(config[{}])**

- **config** - optional  
Configuration object, configuring the hidden info feature.
  - **overviewDialog** true|false - optional  
Shows the overview dialog if the hidden info is triggered by the user.
  - **supportDialog** true|false - optional  
Shows the support dialog if the hidden info is triggered by the user.
  - **domElementId** String - optional



The DOM element ID that the user has to click.

**Results in**

