

MAIBORNWOLFF

CodeCharta Manual

CodeCharta is a visualization tool designed to assist in the analysis and comprehension of complex software projects. It utilizes interactive 3D maps to visualize architectural metrics derived from tools such as SonarQube.

This manual provides guidance on exporting metadata from your project for visualization and analysis.

You can find examples of the output file in our showcase:

<https://maibornwolff.github.io/codecharta/showcase/>

Prerequisites

Before you begin, make sure that:

- Your Code is managed with Git.
- SonarQube instance is running and utilized for code analysis for your project.
- You have access to the SonarQube project.

Installation

To operate CodeCharta, you must install several tools on your machine to run the CodeCharta Shell (ccsh).

Start by installing:

- Java (11 or higher), *we recommend installing the latest OpenJDK LTS version (21 or higher, Temurin build <https://adoptium.net/installation/>)*
- Node.js (18 or higher) and npm (included with Node.js), *we recommend installing the latest NodeJS LTS version (20+ <https://nodejs.org/en/download/package-manager>)*



After installing Java, Node.js and npm, you can install CodeCharta by running the following command in your terminal

```
$ npm i -g codecharta-analysis
```

This command installs CodeCharta and the ccsch command globally on your system.

Exporting Metadata from SonarQube

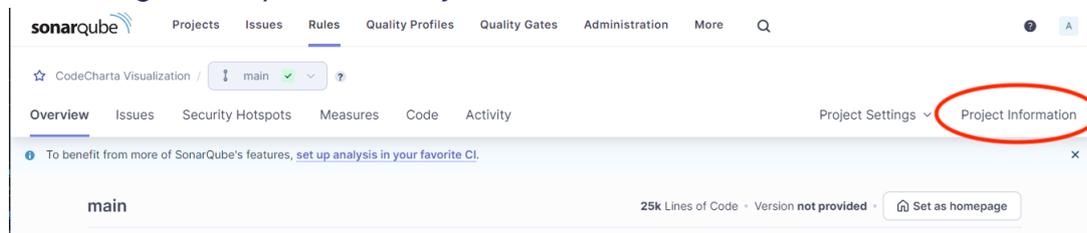
To export metadata from SonarQube, you will need:

- Project key
- User token
- SonarQube URL (where your SonarQube is located)

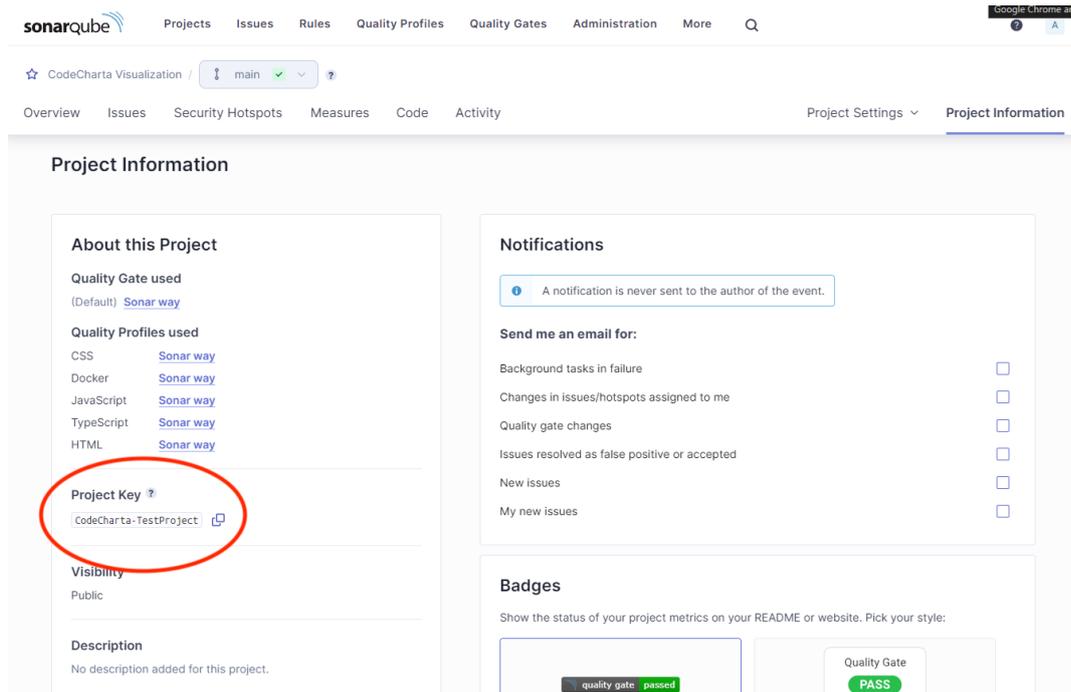
Getting your project key

Your project key is a unique identifier for your project in SonarQube. You have to login to your SonarQube instance and select "Projects" at the top. Find your project and open it.

On the right side you find "Project Information".



Here you can find your project key. Write it down, as we will need it later.

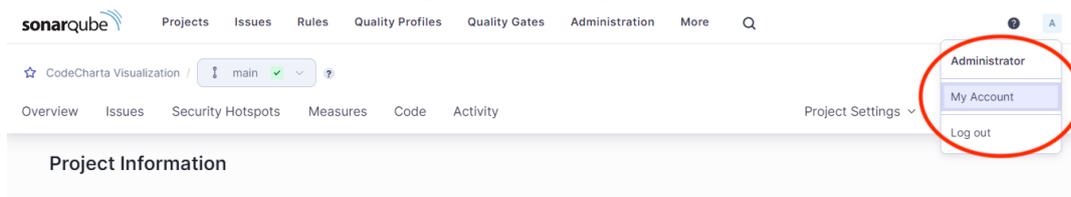


The screenshot shows the SonarQube interface for a project named 'CodeCharta Visualization'. The 'Project Information' section is active. Under 'About this Project', the 'Project Key' is listed as 'CodeCharta-TestProject' and is circled in red. Other sections include 'Quality Gate used' (Sonar way), 'Quality Profiles used' (CSS, Docker, JavaScript, TypeScript, HTML), 'Notifications', and 'Badges' (Quality Gate: PASS).

Getting a user token

For CodeCharta to export metadata from your SonarQube instance it needs to log in. To authorize it you must create a user token. This can be deleted after exporting.

Go to your “My Account”-Page and navigate to security.



The screenshot shows the SonarQube user menu. The user is logged in as 'Administrator'. The 'My Account' option is circled in red, indicating the next step in the process.



A Administrator

Profile Security Notifications Projects

Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name Type Expires in

Name	Type	Project	Last use	Created	Expiration	
Analyze "CodeCharta TestProject"	Project	CodeCharta Visu...	< 1 hour ago	August 6, 2024	September 5, 2024	<input type="button" value="Revoke"/>

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

At "Generate Token" you give the token a name and select "User Token". The name does not matter.

Generate Tokens

Name Type Expires in

Press "Generate" and copy your token.

New token "CodeCharta Importer" has been created. Make sure you copy it now, you won't be able to see it again!

Export

Now that you have all the necessary components, you can proceed to export and parse your metadata. We exclusively export metadata, which encompasses metrics and filenames; no actual code is included in the export.



Export from SonarQube

Open a new terminal and navigate to your project folder.

Run the following command and replace everything in between the <> brackets.

```
$ ccsh sonarimport "<SonarQubeURL>" "<ProjectKey>" "--user-token=<UserToken>" "--output-file=<ProjectName>.sonar" "--merge-modules=false"
```

This will create a file named <Project Name>.sonar.cc.json.gz in your project folder. It includes all the metadata that was exportable from your SonarQube project.

An explanation of the whole command can be found under

<https://maibornwolff.github.io/codecharta/docs/sonar-importer>

Export from Git

To get insights into a couple of metrics, for example number of commits or number of authors. You can parse your gitlog file. Run the following command and replace everything in between the <> brackets.

```
$ ccsh gitlogparser "repo-scan" "--output-file=<ProjectName>.git" "--silent=false" "--add-author=false" "--repo-path=."
```

To ensure privacy, we use the option "--add-author=false" to exclude the names of any authors from the export.

This will generate a file <ProjectName>.git.cc.json.gz which includes the parsed results.

An explanation of the whole command can be found under

<https://maibornwolff.github.io/codecharta/docs/git-log-parser>



Merge

Afterwards both files must be merged. As before, run the following command and replace everything in between the <> brackets:

```
$ cssh merge <ProjectName>.git.cc.json.gz  
<ProjectName>.sonar.cc.json.gz -o <ProjectName>.merged
```

The process merges the data from both sources, resulting in a file named <ProjectName>.merged.cc.json.gz, which will include all results from SonarQube and Git.

This is the file we need 😊

