#HerHackathon Challenge

Green Code Evaluator
Mercedes-Benz AG

Important Links

Slack Channel

GitHub Repo

Goal

We ask you to create a solution that evaluates code according to one or more of five metrics that indicate sustainable principles. You will receive material regarding those metrics, including a short description of each and how they can be identified. We will furthermore provide you with some code samples that you can work with. You can choose whether you want to analyze one programming language or compare statically-typed and dynamically-typed languages to one another, whether you focus on reporting or recommendations and how you define sustainability in this context. For us at Mercedes-Benz, sustainability is a priority. As a world-leading car manufacturer, running environment is a key factor to considerate when designing code. Our developers work in very different environments, ranging from car IT and cloud environments to lean production software and mobile apps. Therefore, you should also try to cover the different running environments in your evaluation. Your solution should evaluate written code and generate a report (as a simple text file), that highlights issues concerning sustainability. Preferable, it also delivers a sustainability index to indicate the overall performance. Join our challenge and help us to produce greener and more sustainable code!

Material

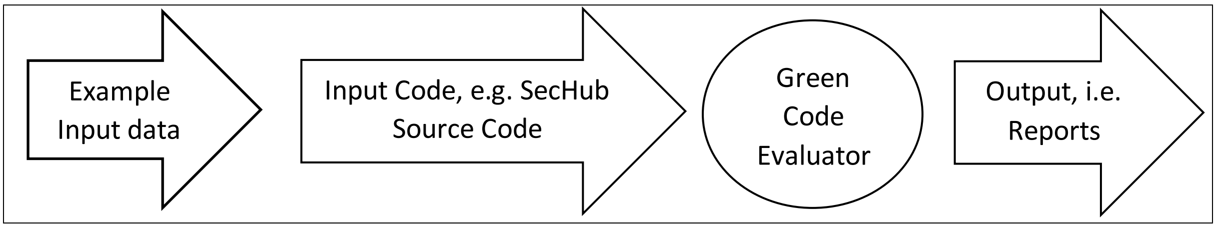
Metrics –[link to slides in repo]

> These slides describe five dimensions that are related to sustainable code: energy consumption, utilization, hardware requirements, accessibility and data ownership. They also give you an idea about how each dimension can be expressed in numbers.
>
> You can pick one or more dimension and implement software to measure them. Of course, if you have an idea what other metrics are useful to describe the sustainability of code, we would be thrilled to learn about your approach.

Code to evaluate - SecHub

> You need to feed some code to your evaluator. We would suggest that you have a look at the source code of SecHub, an open source tool developed by us which provides one API to scan code for security issues.However, you may also use any other free and open source code to demonstrate your evaluator’s abilities.

Impulses



The above schematic shows an abstract representation of the information flow. You can use it to further detail the processes and interfaces or to map the skills of your team members to particular phases.

Other characteristics to look into:

- unused imports of libraries
- unnecessary loops
- algorithmic complexity O(n) / scalability – variation of input data
- data duplication
- comparison of the same functionality among different languages
- code sections with particularly high energy costs
- obsolete network calls
- anti-patterns
- different languages and frameworks vary in consumption

You may want to look at standard profiling software or code analysis tools to see what characteristics are commonly measure. Depending on their license, you may also chose a curating approach.

Checklist

This checklist is meant to help you getting started. By all means, feel free to structure your work according to your own preferences.

☐ Meet Teammates and find a team name

☐ Fork the repo, organize access for everyone

☐ Agree upon documentation practices and collaboration tools: we recommend GitHub projects for task organization and Slack for communication purposes. You might also want to define a space to collect your open questions.

☐ Get to know each other: what skills do you possess? What interests do you share?

☐ Make a rough concept: what could your minimal viable product (MVP) look like? What should it measure, how would it process the data, how are the results presented?

☐ Divide into smaller tasks, prioritize and assign to team members

☐ Set a time at which you want to regroup and sync. Decide how each of you can reach out for help in between the syncs. If you want to, you can also define Milestones to track your progress.

☐ Make sure to reserve some time to prepare for the presentation of your work

☐ Make a pull request to have your code featured in the original repo as a dedicated subfolder (main branch)

Important! Please note that in order for your PULL-request to be accepted, all team members have to sign our Contributor License Agreement and then submit it to Daimler TSS CLA team so that we can accept your contribution.

We look forward to see your ideas come to life! If you need help, feel free to reach out to