RustLogs (RLG) is a powerful and flexible logging library for Rust that provides application-level logging with a simple and readable output format. It offers a wide range of features and customization options to streamline your logging work flow and enhance your Rust applications.

Overview

RustLogs (RLG) simplifies the process of adding logging functionality to your Ru st applications. With its intuitive APIs and helpful macros, you can easily inte grate logging into your codebase and gain valuable insights into your applicatio n's behavior.

divider

Features

Multiple Log LevelsRustLogs (RLG) supports a comprehensive set of log levels, in cluding

,

,

,

,

,

,

,

,

, and

, allowing you to control the granularity of your logs.

Structured Log FormatsThe library provides structured log formats that are easy to parse and filter, enabling efficient analysis and troubleshooting of your app lication's logs.

Extensive Output Format CompatibilityRustLogs (RLG) supports a wide range of out
put formats, including

Common Event Format (CEF) - A standardized log format developed by ArcSight for
the exchange of event information between security devices and applications.Exte
nded Log Format (ELF) - A log format that extends the Common Log Format (CLF) wi
th additional fields to provide more detailed information about web server reque
sts.Graylog Extended Log Format (GELF) - A log format designed for use with the
Graylog log management system, which allows for the transmission of log messages
over a network.JavaScript Object Notation (JSON) - A lightweight data interchang
e format that is easy for humans to read and write and easy for machines to pars
e and generate.NCSA Common Log Format (CLF) - A standardized log format used by
web servers to log client requests, including information such as the client IP
address, request method, and response status code.W3C Extended Log File Format (
W3C) - A log format developed by the World Wide Web Consortium (W3C) that extend
s the Common Log Format (CLF) with additional fields for more detailed logging o
f web server requests.Syslog Format - A standardized log format used by various
network devices and applications to send event messages to a centralized logging
server.Apache Access Log Format - A log format used by the Apache web server to
log client requests, including information such as the client IP address, reques
t method, and response status code.Logstash Format - A log format used by the Lo
gstash data processing pipeline, which allows for the collection, parsing, and f
orwarding of log data from various sources.Log4j XML Format - A log format used
by the Log4j logging framework, which allows for the structured logging of event
s in an XML format.NDJSON (Newline Delimited JSON) - A log format where each log

message is represented as a JSON object on a separate line, making it easy to pa
rse and process log data in a streaming fashion.Flexible ConfigurationCustomize
the behavior of RustLogs (RLG) to suit your application's needs. You can easily
configure the log file path, log levels, and output formats to match your requir
ements.
Asynchronous LoggingRustLogs (RLG) supports asynchronous logging, allowing you t
o log messages without blocking your application's execution. This ensures optim
al performance and responsiveness.
Helpful MacrosThe library provides a set of convenient macros that simplify comm
on logging tasks, such as creating log entries, printing logs to stdout, logging
to files, and conditionally logging based on feature flags or predicates.
Certainly! Here's a CTA (Call to Action) that you can add at the end of the home
page to encourage users to get started with RustLogs (RLG):
divider
Get Started with RustLogs (RLG) Today
Ready to take your Rust application logging to the next level? Get started with
RustLogs (RLG) today and experience the power and flexibility of advanced loggin
g!
Easy Integration
Integrating RustLogs (RLG) into your Rust project is a breeze. With just a few l
ines of code, you can start logging messages and gaining valuable insights into
your application's behaviour.
Comprehensive Documentation
Dive into our comprehensive documentation to learn more about RustLogs (RLG)'s f
eatures, configuration options, and usage examples. Our docs will guide you ever

y step of the way.

Community Support

Join our vibrant community of Rust developers and logging enthusiasts. Get help, share your experiences, and contribute to the future of RustLogs (RLG). Together , let's make logging better!

Get Started Now

divider