

API Documentation

HJ-Lib is a library implementation of the [Habanero-Java \(HJ\) language](#). It relies on the [lambda support in Java 8](#) to provide a syntax close the HJ language.

As such, the [language constructs available in HJ](#) are available as static method calls which accept lambdas as arguments in the library implementation.

Referring to the language constructs will give a good introduction to the static methods provided in HJ-Lib.

The [API documentation for HJ-lib is available here](#). The most important classes that provide support for the static methods to use are:

- [HJ API package](#) and the [HjRuntime](#) class.
- [Module1](#) class
- [Module2](#) class

A summary of the configuration options that can be used to initialize the HJ-lib runtime is available as the different enum constants in the [HjSystemProperty](#) class.

Primers on how to use HJ-lib are also [available to view online here](#). These examples can be used to test the various constructs such as:

- [initializing](#) and [finalizing](#) the runtime
- [finish](#)
- [async](#)
- [forasync](#)
 - Simple variants: ([with int args](#), [with Iterable arg](#), [with HjRegion arg](#))
 - Chunked variants: ([with int args](#), [with HjRegion arg](#))
- [forall](#)
 - Simple variants: ([with int args](#), [with Iterable arg](#), [with HjRegion arg](#))
 - Chunked variants: ([with int args](#), [with HjRegion arg](#))
- [future](#)
- [finish-accumulators](#) ([creation](#), [finish-registration](#))
- [DataDrivenFuture](#) (DDF) ([creation](#), [async await](#))
- [phasers](#) ([creation](#), [next](#), [signal](#), and [wait](#))
- [isolated](#) (variants: [global isolation](#), [object-based isolation](#), [read-write mode isolation](#))
- [actors](#)

and their corresponding variants.

Code examples of how to interact with HJ-Lib API and the different parallel constructs are also available.