

# Index

## Welcome to HiveMind

This wiki contains documentation for the [Hivemind](#) project.

HiveMind is an services and configuration microkernel. Its features are also referred to as Inversion of Control (IoC) Container or Lightweight Container.

The adoption of HiveMind in an application ensures the use of certain design principles which improve encapsulation, modularization, testability and reusability.

- **Services:** HiveMind services are POJOs (Plain Old Java Objects) that can be easily accessed and combined. Each service ideally defines a Java interface it implements (this is not mandatory). HiveMind takes care of the life cycle of services. It instantiates and finalizes services and configures each service just as necessary. HiveMind lets services collaborate with each other via dependency injection, so that the service code itself is released from the task of looking up dependencies.
- **Configuration:** HiveMind allows you to provide complex configuration data to your services in a format you define. HiveMind will integrate the contributions of such data from multiple modules and convert it all into data objects for you. HiveMind configurations allow for powerful, data-driven solutions which combine seamlessly with the service architecture.

## HiveMind 2

HiveMind 2 is a complete refactoring of HiveMind 1.

The Refactoring was mainly done to support alternative ways of defining a registry and its services and configuration points.

Main features of version 2:

- [Java Registry Definition API](#)
- [Annotation Based Definition API \(Annotated modules\)](#)
- [Extensible Autowiring Service](#)
- Arbitrary Configuration Types

## Status



### Alpha

HiveMind 2 is currently in alpha stage. The first alpha version will be released shortly.  
The documentation is unfinished and will get an overhaul during the next weeks.  
The new definition APIs are still subject to change.

If you are interested in using a stable version of HiveMind please go [here](#).