**async rust**

# **What does async mean?**

Async rust is a protocol which allows programmers to run multiple IO bound processes at the same time on a single thread. As IO bound processes, they might have to wait for so IO response, so we can process more tasks at this waiting period. At the time when we are in need of that response we can simply **.await** it, such values are called futures.

# **how async of rust differs from others**

* Rust asynchronous function has to be started either by .await or some executor. Without them all you have is a future which hasn’t even started.
* In rust, u need an external library to do **asynchronous** programming. The standard library provides the executors which handles the futures. Most common executors are **async-std** and **tokio**.

# **coding example**

<https://thomashartmann.dev/blog/async-rust/?fbclid=IwAR2smLKOJnXGQEzH2E0-4gs1q4ha1X4fW650GylbeMY3GlyjJPXpilt9O8A> --- **A MINIMAL EXAMPLE**

# **For you to try**

The above given example has been given to make a nice and clean foundation for your Async programming in rust. Furthermore, you can add **serde** or try using surf’s **recv\_json<T> .** The world of Async programming awaits your approach.

# **references**

**Whole document is the summary of an article shared below.**

[**https://thomashartmann.dev/blog/async-rust/?fbclid=IwAR2smLKOJnXGQEzH2E0-4gs1q4ha1X4fW650GylbeMY3GlyjJPXpilt9O8A**](https://thomashartmann.dev/blog/async-rust/?fbclid=IwAR2smLKOJnXGQEzH2E0-4gs1q4ha1X4fW650GylbeMY3GlyjJPXpilt9O8A)