**ROCKET:**

Rocket is web framework for Rust that makes it simple to write fast, secure web applications without sacrificing flexibility, usability or type safety.  Its major strong point is that it [doesn’t make you sacrifice speed for a productive environment and vice-versa](https://medium.com/@lholznagel/comparing-nodejs-and-rust-http-frameworks-response-times-5738dfa1843d) like many frameworks do. It supports JSON out of the box so you don’t need to have another dependency for that. Form handling is easy and simple. It doesn’t let bad form requests through so your code doesn’t break. It gives you complete control over all aspects of your application.

**ACTIX:**

Actix web is a small, pragmatic, and extremely fast rust web framework. It’s well suited for writing services with hard logic and components. You get to also use these services in an async way. It also provides a lot of features (like logging, http/2, etc.) out of the box. Depending on your preference, It is a great framework with proper documentation.

**CONCLUSION**:

If you want to be on the bleeding edge and design web applications in compliance with emerging standards, **Rocket** is a great choice. That’s not saying the framework falters in other scenarios, but its current reliance on nightly Rust does raise stability concerns. Because Rocket isn’t asynchronous, it can only handle so many connections at any given time. Consequently, it may be better suited to smaller projects, or at least to web applications where active requests are minimized. However, Rocket is quite adept at handling large quantities of data. As the framework matures, it may very well power some large, high-profile projects. Rocket takes much of the guesswork out of the equation for newer users.

**Actix** big advantages lie in two areas: actor management and speed. If you’re building out a web application based around accounts (email, web services with SOAP endpoints), the framework truly shines. This actor support is available in other frameworks like Rocket, though Actix implementation is more elegant. API builds reliant on the WebSockets protocol will mesh harmoniously with Actix. If you’re trying to squeeze as much performance out of your web servers as possible, the protocol’s low overhead will facilitate that.