Express JS - Advantages

- Almost the standard for Node.js web middleware.
- Simple, minimalistic, flexible and scalable.
- Fast app development.
- Fully customizable.
- Low learning curve.
- Easy integration of third-party services and middleware.
- Majorly focused on browsers, making templating and rendering an almost out of the box feature.

Express JS - Disadvantages

- Organization needs to be very clear to avoid problems when maintaining the code.
- As your codebase size increases, refactoring becomes very challenging.
- A lot of manual labor is required, since you need to create all endpoints.

KOA - Advantages

- Koa improves interoperability, robustness, and makes writing middleware much more enjoyable.
- Has a large number of helpful methods but maintains a small footprint, as no middleware are bundled.
- Koa is very lightweight, with just 550 lines of code.
- Has a very good user experience.
- Better error handling through try/catch.
- Generated-based control flow.
- No more callbacks, facilitating an upstream and downstream flow of control.
- Cleaner, more readable async code.

KOA - Disadvantages

- The open source community around Koa is relatively small.
- Not compatible with Express-style middleware.
- Koa uses generators which are not compatible with any other type of Node.js framework middleware.

Hapi - Advantages

- It provides a robust plugin system that allows you to add new features and fix bugs at a fast pace.
- It enables you to build scalable APIs.
- There is a deeper control over request handling.
- It is an excellent choice for building Representational State Transfer (REST) APIs because it provides you with routing, input, output validation, and caching.
- You can build an API that serves all kinds of clients that need mobile and single-page applications.
- Detailed API reference and good support for document generation.
- You can use hapi.js with any front-end framework, like React, Angular, and Vue.js to create a single-page application.
- Configuration-based approach to some sub-middlewares(pseudo-middlewares)
- Provides the availability of caching, authentication, and input validation.
- Has a plugin-based architecture for scaling.
- Provides really good enterprise plugins such as joi, yar, catbox, boom, tv, and travelogue.

Hapi - Disadvantages

- Developers need to figure out the code structure on their own.
- “Locks” developers into using hapi-specific modules and plugins such as catbox, joi, boom, tv, good, travelogue, and yar; and which are not compatible with Express/Connect.
- Endpoints are created manually and must be tested manually.
- Refactoring is manual.

Meteor - Advantages

- **Web and Mobile** − Meteor offers a platform for developing Web, Android and IOS apps.
- **Universal Apps** − The same code for web browsers and mobile devices.
- **Packages** − Huge number of packages that are easy to install and use.
- **Meteor Galaxy** − Cloud service for Meteor app deployment.
- Developers only need JavaScript for server and client side development.
- Coding is very simple and beginner friendly.
- Meteor apps are real time by default.
- Official and community packages are huge time saver.

Meteor - Disadvantages

- Meteor isn't very suitable for large and complex applications.
- There is a lot of magic going on when working with Meteor, so developers might find themselves limited in some way.

Sails - Advantages

- Provides good code organization and blueprints
- Built-in support for WebSockets
- Supports various databases
- Data validation
- Auto-generated code for controllers, models, and routes
- Many out-of-the-box security features, e.g., CSRF and compatibility with Lusca
- Built-in file upload library
- Good documentation
- Flexible and modular architecture with hooks and plugins

Sails - Disadvantages

- Steep learning curve
- Opinionated