

# Request types

## Contents

### 1. [Requests](#)

---

## Requests

By default the request parameters, such as those from a form POST, will be automatically parsed and set as a `:params` key on the request.

However, if you're passing in parameters as a request body then you will need to enable the appropriate middleware to handle them. Luminus uses [ring-middlewares-format](#) for encoding and decoding the body parameters.

The request parameters will be available under the `:params` key of the request. Note that the middleware will also handle encoding the response bodies when you set the appropriate MIME type on the response. Please see the [response types](#) section for more information on generating responses.

Below is a list of valid formats:

- `:json` - JSON with string keys in `:params` and `:body-params`
- `:json-kw` - JSON with keywordized keys in `:params` and `:body-params`
- `:edn` - native Clojure format.
- `:yaml` - YAML format
- `:yaml-kw` - YAML format with keywordized keys in `:params` and `:body-params`
- `:yaml-in-html` - yaml in a html page
- `:transit-json` Transit over JSON format
- `:transit-msgpack` Transit over Msgpack format

Build Tool:

## Topics

- Your First Application
- REPL Driven Developme
- Application Profiles
- HTML Templating
- Static Assets
- ClojureScript
- Routing
- RESTful Services
- **Request types**
  - Response types
  - Websockets
  - Middleware
  - Sessions and Cookies
  - Input Validation
  - Security
  - Component Lifecycle
  - Database Access
  - Database Migrations
  - Logging
  - Internationalization
  - Testing
  - Server Tuning
  - Environment Variables
  - Deployment
  - Useful Libraries
  - Sample Applications
  - Upgrading
  - Clojure Resources

## Books

Web Development with  
**Clojure**  
2nd Edition  
Build Bulletproof Web Apps  
with Less Code



---

Luminus framework is released under the [MIT License](#) - Copyright © 2019