



## SETUP & INSTANCE

```
npm install fastify
import Fastify from 'fastify'
const fastify = Fastify({ logger: true
})
await fastify.listen({ port: 3000, host:
'0.0.0.0' })
fastify.log.info('Server started')
Fastest Node.js HTTP framework
(benchmarks #1 2024)
```

## ROUTES

```
fastify.get('/users', async (request,
reply) => {
return { users: [] } // auto-serialized
})
fastify.post('/users', {
schema: { body: userSchema, response: {
201: userResponse } }
}, async (req, reply) => {
reply.status(201).send(newUser)
})
```

## SCHEMA VALIDATION

```
const schema = {
body: {
type: 'object',
required: ['name','email'],
properties: {
name: { type: 'string', minLength: 2 },
email: { type: 'string', format: 'email'
}
}
}
}
JSON Schema via Ajv; also boosts
serialization 2x
```

## PLUGINS

```
import fp from 'fastify-plugin'
const myPlugin = fp(async (fastify,
opts) => {
fastify.decorate('db', new DB(opts))
})
fastify.register(myPlugin, { uri:
process.env.DB_URI })
fastify.register(require('@fastify/cors'))
fastify.register(require('@fastify/jwt'),
{ secret: '...' })
fp() makes plugin available across
encapsulation boundaries
```

## DECORATORS

```
fastify.decorate('config', { port: 3000
})
fastify.decorateRequest('user', null)
fastify.decorateReply('sendError',
function(code, msg) {
this.status(code).send({ error: msg })
})
// In handler:
request.user / reply.sendError(401,
'Unauthorized')
```

## REQUEST & REPLY

```
request.body / request.params /
request.query
request.headers / request.ip /
request.id
reply.send(data) // infers Content-Type
reply.status(404).send({ error: 'Not
found' })
reply.type('text/html').send('<h1>Hi</h1>')
reply.redirect('/new-url')
reply.header('X-Custom', 'value')
```

## LOGGING

```
const fastify = Fastify({
logger: {
level: 'info',
transport: { target: 'pino-pretty' }
}
})
request.log.info({ userId: id },
'Fetching user')
fastify.log.error({ err }, 'Fatal
error')
Built on Pino very fast structured
logging
```

## COMMON PLUGINS

```
@fastify/jwt JWT auth
@fastify/cors CORS headers
@fastify/static static files
@fastify/swagger OpenAPI docs
@fastify/rate-limit rate limiting
@fastify/compress gzip/deflate
@fastify/sensible httpErrors helpers
```

## COMMON GOTCHAS

Schema must use  
additionalProperties:false or extra  
props pass  
Decorated properties must be  
initialized in decorateRequest/Reply  
Plugin encapsulation: fp() needed to  
share across contexts  
reply.send() after async handler =  
duplicate response error  
JSON schema 'format' requires  
ajv-formats plugin