

DOS

The developer needs to be familiar with Laravel API structure and use this mental model:

“Client → route (versioned, throttled, auth) → controller → Form Request (validation + policy) → domain logic (services/models/transactions) → API Resource (shape output) → JSON with correct status codes and error format.”

The developer needs to understand the specs and create a plan of action.

Routing (versioned)

- Add routes to routes/api.php, usually under /v1 and auth middleware.
- Keep names consistent (api.v1.posts.favorite.store, etc.).

Data shape

- New tables? Write migrations and factories.
- Add relationships on models.
- Indexes/uniques/foreign keys.

Validation & Authorization

- Validate request content
- Use Policies (or Gates) for permissions.

Controller logic

- Keep controllers thin; push logic into model/service.
- Use transactions when touching multiple tables.
- Return proper status codes (201, 204, 400, 401, 403, 404, 422).

Resources (transformers)

- Create an API Resource to shape responses and hide internal fields.
- Add pagination using Resource::collection(\$query->paginate()).

Tests (Feature)

- Assert JSON shape, status codes, DB state, and policy enforcement.

Volopa uses custom authorization processes via middlewares for OAuth2 type access tokens or WSSE credentials.

DONTS

- Don't use a class or a method that doesn't exist in the current repository or was not defined in the added code.
- Don't add methods that already exists in the current repository
- Don't return raw Eloquent models
- Don't use session/redirect patterns in APIs.
- Don't return 200 for errors
- Don't expose stack traces or sensitive error details.
- Don't disable mass-assignment protection or trust client-owned fields.
- Don't build query filters directly from user input.
- Don't create N+1 queries (especially in Resources).
- Don't return unbounded lists.
- Don't forget transactions for multi-write operations.
- Don't hardcode timestamps/timezones.
- Don't ignore caching.
- Don't let file uploads bloat the API process.
- Don't respond with inconsistent shapes/casing.
- Don't leak environment or config.
- Don't forget observability.