

Quy ước đặt tên biến và hàm cho dự án Vue + Laravel

Dưới đây là các quy ước đặt tên biến và hàm được khuyến nghị cho dự án kết hợp Vue.js và Laravel:

Laravel (Backend)

Tên bảng trong cơ sở dữ liệu

- Sử dụng **snake_case** và số nhiều: users, blog_posts, product_categories

Models

- Sử dụng **PascalCase** (viết hoa chữ cái đầu) và số ít: User, BlogPost, ProductCategory
- Tên model tương ứng với bảng, nhưng ở dạng số ít

Controllers

- Sử dụng **PascalCase** kèm hậu tố "Controller": UserController, BlogPostController, ProductCategoryController
- Phương thức tuân theo RESTful conventions:
 - index(): Hiển thị danh sách tài nguyên
 - create(): Hiển thị form tạo mới
 - store(): Lưu tài nguyên mới
 - show(): Hiển thị tài nguyên cụ thể
 - edit(): Hiển thị form chỉnh sửa
 - update(): Cập nhật tài nguyên
 - destroy(): Xóa tài nguyên

Routes

- Sử dụng **kebab-case** cho URL: /blog-posts, /product-categories
- Sử dụng **camelCase** cho tên route: blogPosts.index, productCategories.show

Functions và Methods

- Sử dụng **camelCase**: getUserData(), calculateTotal()
- Tên hàm nên bắt đầu bằng động từ: get, post, update, delete, calculate, process

Biến

- Sử dụng **camelCase**: \$userName, \$totalAmount, \$isActive

- Biến boolean nên bắt đầu bằng is, has, should: \$isAdmin, \$hasAccess

Constants

- Sử dụng **UPPER_SNAKE_CASE**: APP_NAME, MAX_LOGIN_ATTEMPTS

Vue.js (Frontend)

Component Names

- Sử dụng **PascalCase** trong khai báo: AppHeader, UserProfile
- Sử dụng **kebab-case** trong templates HTML: <app-header>, <user-profile>
- Nên sử dụng tên đầy đủ, tránh viết tắt

Props

- Sử dụng **camelCase** khi khai báo trong script: userData, itemList
- Sử dụng **kebab-case** khi truyền trong HTML: :user-data, :item-list

Events

- Sử dụng **kebab-case**: click, submit, user-logged-in
- Custom events nên theo quy ước: update:modelValue cho v-model

Data, Computed, Methods

- Sử dụng **camelCase**: userData, calculateTotal()
- Method nên bắt đầu bằng động từ: fetchUsers(), submitForm()

Vuex Store

- **camelCase** cho state và getters: isLoggedIn, currentUser
- **UPPER_SNAKE_CASE** cho mutations: SET_USER, CLEAR_ERRORS
- **camelCase** cho actions: fetchUser, updateProfile

Lưu ý cho cả 2 phía

- Tránh sử dụng tên biến/hàm quá ngắn hoặc không rõ nghĩa: a, temp, x
- Sử dụng tên có ý nghĩa thể hiện mục đích của biến/hàm
- Nhất quán với quy tắc đặt tên trong toàn bộ dự án
- Tránh sử dụng tiếng Việt có dấu trong tên biến và tên hàm để tránh các vấn đề về encoding

Những quy ước này sẽ giúp code của bạn dễ đọc, dễ bảo trì và tuân theo các tiêu chuẩn tốt nhất của cả Laravel và Vue.js.