# Localization in flutter

quá trình chuẩn bị ứng dụng để hỗ trợ nhiều ngôn ngữ và vùng lãnh thổ khác nhau mà **không cần thay đổi mã nguồn.**

**Theory**

**L10n** là quá trình điều chỉnh một ứng dụng để phù hợp với ngôn ngữ, văn hóa, và tiêu chuẩn của một khu vực hoặc một quốc gia cụ thể.

This is an important step after initiated **i18n**

# Tầm quan trọng của L10n trong ứng dụng đa ngôn ngữ

* ***Tăng khả năng tiếp cận người dùng:*** cung cấp giao diện và nội dung bằng ngôn ngữ mẹ đẻ của người dùng
* ***Cải thiện trải nghiệm người dùng:*** Người dùng sẽ có trải nghiệm tốt hơn khi họ tương tác với một ứng dụng sử dụng ngôn ngữ và định dạng văn hóa quen thuộc.
* ***Tối ưu hóa thị trường:*** hỗ trợ nhiều ngôn ngữ giúp ứng dụng dễ dàng mở rộng sang các thị trường mới, tăng tính cạnh tranh và cơ hội thành công.

# Gói Intl của FLUTTEr

**Gói intl** được sử dụng để hỗ trợ việc **định dạng dành riêng cho ngôn ngữ**, bao gồm ngày tháng, số, tiền tệ, và quản lý bản dịch.

**CÁC TÍNH NĂNG CHÍNH:**

* **Định dạng Date:** vài nơi có định dạng MM/dd/yyyy; trong khi Việt Nam là dd/MM/yyyy.
* **Định dạng số và tiền tệ:** định dạng số và tiền tệ theo cách hiển thị phù hợp với văn hóa của từng quốc gia *(ví dụ: sử dụng dấu phẩy làm phân cách hàng nghìn ở một số nước và dấu chấm ở những nước khác).*
* **Quản lý bản dịch:** cung cấp cơ chế để xử lý bản dịch của chuỗi văn bản trong ứng dụng. Điều này thường kết hợp với các tệp tin ARB (Application Resource Bundle), nơi chứa các bản dịch cho mỗi ngôn ngữ.

**Để quản lý bản dịch:**

Tạo các tệp **.arb** chứa các chuỗi văn bản cần dịch. Kết hợp với công cụ flutter\_localizations để tạo bản dịch tự động.

**Biện pháp thực hành tốt nhất cho quản lý bản dịch**

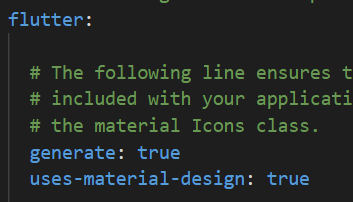
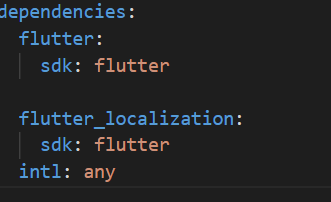
1. **Sử dụng tệp .arb**: Đây là định dạng JSON được khuyến nghị cho việc quản lý các bản dịch trong Flutter
2. **Tránh hard-code văn bản trong mã nguồn**
3. **Kiểm tra kỹ các bản dịch**

# CONCLUSION

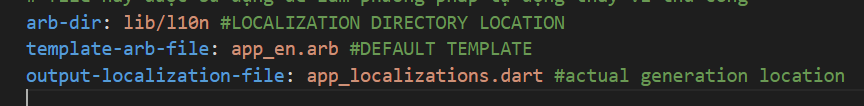
Sử dụng gói intl trong Flutter là một cách hiệu quả để quản lý các bản dịch và định dạng dữ liệu dành riêng cho ngôn ngữ, giúp ứng dụng có khả năng hỗ trợ người dùng trên toàn cầu. Việc áp dụng L10n không chỉ nâng cao trải nghiệm người dùng mà còn mở rộng khả năng phát triển thị trường cho ứng dụng.

**Cách bắt đầu một app i18n va l10nád**

**STEP 1:**Tạo project mới và thêm **flutter\_localization và generate** trong **pubspec.yaml** như trong hình

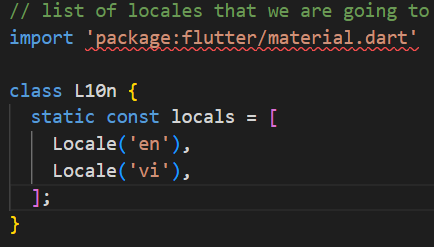
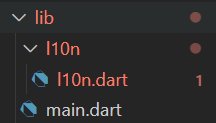
****

**STEP 2:** Tạo một file tên l10n.yaml (***file này được sử dụng để làm phương pháp tự động thay vì thủ công)***

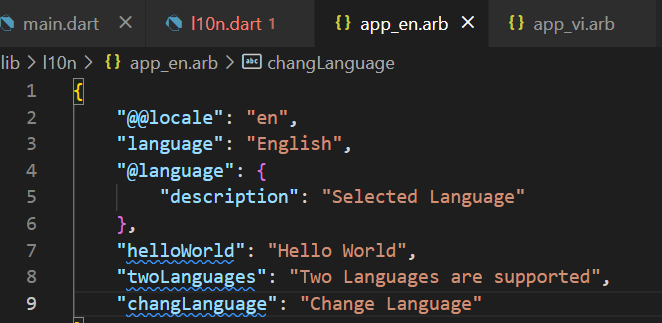
******

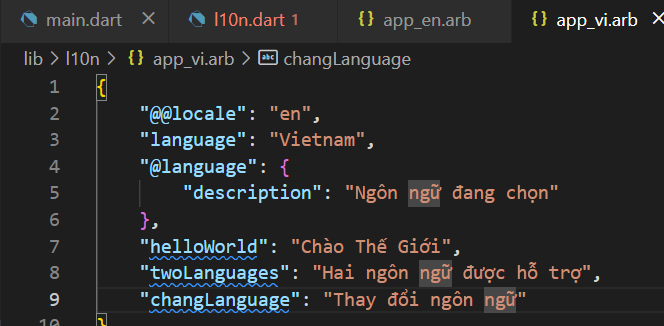
**STEP 3:** Tạo một Folder tên l10n và file con trùng với tên folder. Trong file l10n.dart sẽ chứa danh sách những địa phương ngôn ngữ mình sử dụng

Điều này đơn giản chỉ là một danh sách các Locale mà ứng dụng hỗ trợ, nhưng không trực tiếp liên quan đến việc xử lý bản dịch (translation). Lớp L10n chỉ liệt kê các ngôn ngữ được hỗ trợ mà không tự động tạo ra các bản dịch hay cung cấp cách truy xuất bản dịch.



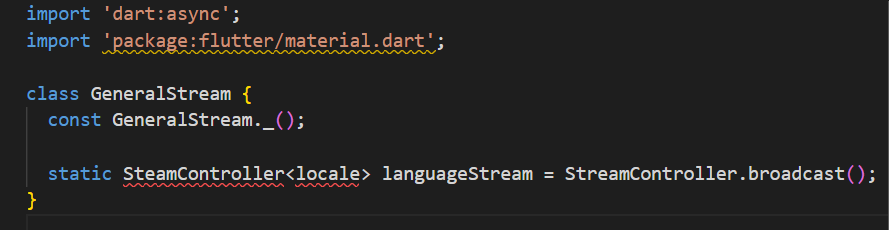
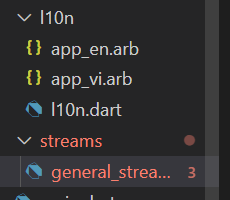
**STEP 4:** Tạo file cho các ngôn ngữ





**STEP 5: Sử dụng StreamController để quản lý thay đổi ngôn ngữ (trong general\_stream.dart)**

ngôn ngữ được quản lý thông qua Stream. Khi người dùng chọn thay đổi ngôn ngữ, sự kiện này sẽ được truyền qua StreamController, và các widget trong ứng dụng sẽ theo dõi sự thay đổi này để cập nhật ngôn ngữ hiển thị.

****

Cách này giúp truyền ngôn ngữ giữa các phần khác nhau trong ứng dụng một cách linh hoạt hơn thông qua **Stream**

**STEP 6: make dealing with localization translation easier**