in/poyrazavsever

I18N MIMARISI

Uygulamanın Dilini Koddan Ayırmak

in/poyrazavsever

i18n Mimarisi Nedir?

i18n (internationalization), uygulamanın metinlerini, tarihlerini ve sayısal biçimlerini kullanıcı kültürüne göre özelleştirilebilir hale getirme sürecidir.

Amaç, tek kod tabanından çok dilli deneyim sunmak; yani metinleri koddan ayırarak sürdürülebilirlik ve ölçeklenebilirlik kazanmaktır.

in/poyrazavsever

i18n ve l10n Arasındaki Fark

Uluslararasılaştırma mı, Yerelleştirme mi?

- i18n: Yazılımın çok dil destekli çalışabilmesi için gerekli altyapının kurulması.
- l10n: Bu altyapı üzerine spesifik dil ve kültür adaptasyonunun yapılması.

Bir proje i18n olmadan l10n yapılamaz. Önce sistem, sonra içerik gelir.

Mimarinin Katmanları - UI – Core – Locale Ayrımı

- UI Katmanı: t("login.title") gibi çeviri anahtarlarını çağırır.
- Core Katmanı: Aktif dili, fallback zincirini ve depolama stratejisini yönetir.
- Locale Katmanı: JSON veya CMS kaynaklarından dil dosyalarını sağlar.

Bu katmanlar birbirine sıkı sıkı bağlı değil, bağımsız test edilebilir olmalıdır.

Anahtar Bazlı Yapı

Anahtarlar, UI ile dil dosyası arasında köprü kurar.

Metin değil, anlam taşırlar:

UI bileşenleri bu anahtarları çağırır; metin değişse bile kod sabit kalır.

```
{
  "auth.login": "Giriş Yap",
  "auth.register": "Kayıt Ol"
}
```

in/poyrazavsever

Namespace Mimarisi

Her modül kendi çeviri dosyasına sahip olabilir:

```
locales/
├─ tr/
│   ├─ auth.json
│   ├─ dashboard.json
│   └─ errors.json
└─ en/
    ├─ auth.json
    ├─ dashboard.json
    └─ errors.json
```

Bu yapı “feature-based” i18n mimarilerinde performans avantajı sağlar.

in/poyrazavsever

Fallback ve Dil Önceliği

Bir dil bulunamadığında **fallback** zinciri devreye girer:

tr-TR → tr → en

Kullanıcının **navigator.language** değeri alınır, desteklenmiyorsa varsayılan dile yönlendirilir.

Doğru **fallback** yapısı, eksik çevirilerde bozulmayı önler.

Formatlama Katmanı

Intl API veya i18next formatlama seçenekleri ile kültüre özgü veri biçimleri yönetilir:

```
new Intl.DateTimeFormat("tr-TR", { dateStyle: "long" })
  .format(new Date("2025-10-28"));
// "28 Ekim 2025"
```

Aynı tarih farklı kültürlerde farklı görünür.

Dinamik İçerik & CMS Entegrasyonu

Çeviriler sadece JSON dosyalarında değil, CMS üzerinden de yönetilebilir.

Örneğin Strapi, Contentful veya Supabase tabanlı bir sistem:

```
CMS → JSON API → i18n Loader → Frontend
```

Böylece içerik ekibi geliştiriciye ihtiyaç duymadan güncelleme yapabilir.

in/poyrazavsever

Context & Dil Geçişi - Anlık Dil Değişimi Nasıl Yönetilir?

UI içinde Context veya Store (React Context, Zustand, NgRx) kullanılabilir:

```
const { t, locale, setLocale } = useI18n();

<select onChange={(e) => setLocale(e.target.value)}>
  <option value="tr">Türkçe</option>
  <option value="en">English</option>
<select>
```

Seçim localStorage veya cookie’de saklanarak oturumlar arası korunur.

Performans & Lazy Loading

Her dil veya modül için çevirileri ayrı “chunk” olarak yükle:

```
const i18n = await import(`../locales/${locale}/auth.json`);
```

Next.js veya Angular’da route bazlı lazy-load, bundle boyutunu küçültür.

Test & Doğrulama Katmanı

CI sürecine “missing keys” ve “unused keys” kontrolü ekle:

```
npx i18n-lint --src src/locales --strict
```

Eksik anahtarlar build öncesinde tespit edilir, böylece runtime hataları azalır.

in/poyrazavsever

i18n ve SEO

- URL yapısı: /tr/about – /en/about
- <html lang="tr"> etiketi doğru tanımlanmalı
- hreflang meta etiketleri arama motorlarını yönlendirir

Next.js için i18n config’inde locales ve defaultLocale tanımı yapılmalıdır.

Sık Yapılan Hatalar

- Sabit metinleri doğrudan UI içinde kullanmak
- Aynı anahtarı farklı anlamlarda kullanmak
- Namespace’leri karıştırmak
- Fallback zincirini yanlış kurmak

Çözüm: Tek sözleşme, tek kaynak, otomatik kontrol.

in/poyrazavsever

Sürdürülebilir i18n Stratejisi

Bir i18n mimarisi sürdürülebilirse değerlidir.

Anahtar isimleri, dosya yapısı ve veri kaynakları zamanla büyüyebilir ama standart bozulmamalı.

Kural: “Her dil farklı olabilir ama yapı tek kalmalı.”

Çok Dilli Olmak, Global Düşünmenin Başlangıcıdır.

Geri bildirimlerini paylaş, birlikte daha iyi mimariler üretelim.

in/poyrazavsever