

Guia Completo - Sistema de traduções Next-intl

Sistema de Traduções - Guia Rápido

Traduções internacionalizadas usando `next-intl` com estrutura modular

Estrutura

```
1 messages/locales/
2   └─ pt/
3       ├── common.json      # Labels comuns (ações, status)
4       ├── auth.json        # Login/autenticação
5       ├── users.json       # Usuários
6       ├── units.json       # Unidades
7       ├── suppliers.json   # Fornecedores
8       ├── defensives.json  # Itens agrícolas
9       ├── smart-caldas.json # Smart Caldas
10      ├── maintenance.json  # Manutenção
11      ├── audit.json        # Auditoria
12      └─ layout.json        # Sidebar/Footer
13   └─ en/ (mesma estrutura)
14   └─ es/ (mesma estrutura)
```

Como Usar

Importar traduções em componentes:

```
1 import { useTypedTranslations } from "@app/hooks/useTypedTranslations"
2 import { AppMessagesEnum } from "@app/utils/enum/app-messages.enum"
3
4 export function MeuComponente() {
5   const t = useTypedTranslations(AppMessagesEnum.Users)
6
7   return (
8     <div>
9       <h1>{t("GerenciamentoUsuarios.titulo")}</h1>
10      <label>{t("GerenciamentoUsuarios.lbNome")}</label>
11    </div>
12  )
13 }
```

Usar traduções comuns:

```
1 const tCommon = useTypedTranslations(AppMessagesEnum.Common)
2
3 <button>{tCommon("acaoCreate")}</button> { /* "Criação" */ }
4 <button>{tCommon("acaoUpdate")}</button> { /* "Atualização" */ }
5 <button>{tCommon("acaoDelete")}</button> { /* "Exclusão" */ }
```

Adicionar Novas Traduções

1. Adicionar chave em módulo existente:

Edite os 3 arquivos (pt, en, es):

```
1 // messages/locales/pt/users.json
```

```

2 {
3   "GerenciamentoUsuarios": {
4     "titulo": "Gerenciamento de Usuários",
5     "lbNome": "Nome",
6     "lbTelefone": "Telefone" // ← Nova chave
7   }
8 }

```

2. Criar novo módulo:

a) Crie os arquivos JSON:

- messages/locales/pt/novo-modulo.json
- messages/locales/en/novo-modulo.json
- messages/locales/es/novo-modulo.json

b) Adicione ao app/i18n/config.ts :

```

1 const MODULE_MAPPINGS = {
2   pt: {
3     // ... módulos existentes
4     NovoModulo: () => import("@/messages/locales/pt/novo-modulo.json"),
5   },
6   en: {
7     // ... módulos existentes
8     NovoModulo: () => import("@/messages/locales/en/novo-modulo.json"),
9   },
10  es: {
11    // ... módulos existentes
12    NovoModulo: () => import("@/messages/locales/es/novo-modulo.json"),
13  },
14 }

```

c) Adicione a tipagem em app/types/messages.ts :

```

1 export interface AppMessages extends Messages {
2   // ... outros módulos
3   NovoModulo: {
4     SubModulo: {
5       chave: string
6     }
7   }
8 }

```

d) Adicione ao enum em app/utils/enum/app-messages.enum.ts :

```

1 export enum AppMessagesEnum {
2   // ... outros
3   NovoModulo = "NovoModulo",
4 }

```

 Validar Traduções

```

1 # Verificar consistência entre idiomas
2 npx tsx scripts/validate-translations.ts
3
4 # Verificar tipagem TypeScript
5 npx tsc --noEmit

```

⚠️ Regras Importantes

✅ FAÇA:

- Use objetos aninhados: `{ "GerenciamentoUsuarios": { "lbNome": "Nome" } }`
- Adicione chaves nos 3 idiomas (pt, en, es) simultaneamente
- Use `Common` para labels reutilizáveis (ações, status, etc)
- Valide antes de commitar

❌ NÃO FAÇA:

- Não use strings com pontos: `"GerenciamentoUsuarios.lbNome": "Nome"`
- Não deixe valores vazios: `"titulo": ""`
- Não adicione chaves em apenas 1 idioma
- Não duplique chaves entre módulos

🔧 Troubleshooting

Erro: "Cannot find module"

Solução: Verifique se adicionou o módulo em `MODULE_MAPPINGS` no `config.ts`

Erro: "MISSING_MESSAGE"

Solução: Verifique se o namespace está em `AppMessagesEnum` e `messages.ts`

Erro: TypeScript - Key não existe

Solução: Adicione a chave ao JSON e atualize a interface em `app/types/messages.ts`

Avisos de validação

Solução: Execute `npx tsx scripts/validate-translations.ts` e adicione as chaves faltantes

📖 Módulos Disponíveis

Enum	Namespace	Uso
<code>AppMessagesEnum.Common</code>	<code>Common</code>	Ações, status, filtros comuns
<code>AppMessagesEnum.Auth</code>	<code>Auth</code>	Login, recuperação de senha
<code>AppMessagesEnum.Users</code>	<code>Users</code>	Gerenciamento de usuários

AppMessagesEnum.Units	Units	Gerenciamento de unidades
AppMessagesEnum.Suppliers	Suppliers	Gerenciamento de fornecedores
AppMessagesEnum.Defensivos	Defensivos	Itens agrícolas
AppMessagesEnum.SmartCaldas	SmartCaldas	Smart Caldas
AppMessagesEnum.Maintenance	Maintenance	Manutenção
AppMessagesEnum.Audit	Audit	Auditoria
AppMessagesEnum.Layout	Layout	Sidebar, footer, navegação

Última atualização: 14/01/2025

Dúvidas? Consulte a equipe de frontend