

Initialize NLS
In store.js
syncTranslationWithStore : set getTranslation's function to get data from store
loadTranslations : load translations which get from AEM server into the store
In Header.js
setLocale : set Locale when componentDidMount and switchLang

Store.js

```
const mainEndpoint =CDN.getStaticCMSPath('/apps/nett/dictionaries/main.json',false);
const accountsEndpoint =CDN.getStaticCMSPath('/apps/nett/dictionaries/ai.json',false);

axios.all([axios.get(mainEndpoint), axios.get(accountsEndpoint)])
.then(axios.spread(function (mains, accounts) {
  // Register translations
  let mainsTranslations = i18nUtil.unflattenTranslations(mains.data);
  let accountsTranslations = i18nUtil.unflattenTranslations(accounts.data);
  syncTranslationWithStore(stores);
  stores.dispatch(loadTranslations(_.merge({}, I18n._translations, mainsTranslations, accountsTranslations)));
})))
```

React-redux-i18n

index.js

syncTranslationWithStore

action.js

loadTranslations

setLocale

Header.js

componentDidMount

this.props.dispatch(setLocale(this.props.user.item.userLanguage));

switchLang(lang)

this.props.dispatch(setLocale(lang));

Get NLS in project

I18n.t('nav.alerts', {arg0: alertCount})

React-redux-i18n

action.js

React-i18nify

I18n.js

t

Package

JS

function

Source code

React-redux-i18n

action.js

```
LOAD_TRANSLATIONS = '@@i18n/LOAD_TRANSLATIONS'
SET_LOCALE = '@@i18n/SET_LOCALE';
function setLocale(locale) {
  return function (dispatch) {
    dispatch({
      type: SET_LOCALE,
      locale: locale
    });
  }
}
function loadTranslations(translations) {
  return function (dispatch) {
    dispatch({
      type: LOAD_TRANSLATIONS,
      translations: translations
    });
  };
};
```

Switch Locale
loadTranslations

reducer.js

```
switch (action.type) {
  case _actions.LOAD_TRANSLATIONS:
    return _extends({}, state, {
      translations: action.translations
    });
  case _actions.SET_LOCALE:
    return _extends({}, state, {
      locale: action.locale
    });
  default:
    return state;
}
```

index.js

```
var _reactI18nify = require('react-i18nify');
exports.I18n = _reactI18nify.I18n;
exports.Translate = _reactI18nify.Translate;
exports.Localize = _reactI18nify.Localize;
function syncTranslationWithStore(store) {
  _reactI18nify.I18n.setTranslationsGetter(function () {
    return store.getState().i18n.translations;
  });
  _reactI18nify.I18n.setLocaleGetter(function () {
    return store.getState().i18n.locale;
  });
}
```

React-i18nify

I18n.js

Translate.js

Localize.js

setTranslationsGetter: function setTranslationsGetter(fn) {
 this._getTranslations = fn;
},

setLocaleGetter: function setLocaleGetter(fn) {
 this._getLocale = fn;
},

t: function t(key) {
 return this._translate(key, replacements);
},

_translate: function _translate(key) {
 var translation = '';
 var translationLocale = this._translations[this._locale] ? this._locale : this._locale.split('-')[0];
 translation = this._fetchTranslation(this._translations, translationLocale + '.' + key, replacements.count);
 return this._replace(translation, replacements);
},

_fetchTranslation: function _fetchTranslation(translations, key) {
 if (translations[key]) {
 return translations[key];
 }
},

get_translations() {
 return this._getTranslations ? this._getTranslations() : this._translationsObject;
},

set_translations(translations) {
 this._translationsObject = translations;
},

get_locale() {
 return this._getLocale ? this._getLocale() : this._localeKey;
},

set_locale(locale) {
 this._localeKey = locale;
},