

WONder-ERXLocalizer

Overview

ERXLocalizer provides easy KVC (Key-Value Coding) access to localization. For a short description and a full list of all available methods please see the api (<http://wocommunity.org/documents/javadoc/wonder/latest/er/extensions/localization/ERXLocalizer.html>).

How to use

The easiest way to use localization in you WebObjects applications is to use Project Wonder's built-in localization support using the the localizer() method in the [ERXSession|WO:Programming__WebObjects-WONder-ERXSession] class. The localizer() method allows you to bind your localized language strings directly to you components. Lets say you want to localize your WebObjects application in english and german. The only thing you have to do is to create the required language resources in you application's resource directory. If you use Eclipse to develop your WebObjects applications, this is the . /Resources directory in you application root directory.

For english and german localization you need to create the following files and directories:

```
Resources
|
+-English.lproj
|   |
|   +-Localizable.strings
|
+-German.lproj
|   |
|   +-Localizable.strings
```

The Localizable.strings files contain the key-value pairs with you your localization. The key is always the *placeholder* for the translation and it is used in your WebObjects application to address the localized value. If your application is a simple *hello world* application, your localized files may look like

```
{
  "hello" = "Hello";
  "world" = "World";
}
```

for the Localizable.strings file in the English.lproj directory and

```
{
  "hello" = "Hallo";
  "world" = "Welt";
}
```

for the Localizable.strings file in the German.lproj directory.

To use the localized strings just bind the keys to an appropriate WOComponent (assuming you inherited your Session object from [ERXSession|WO: Programming__WebObjects-WONder-ERXSession] in your application).

```
<p>The application says: <webobject name="langHello"/> <webobject name="langWorld"/></p>
```

(HTML file)

```
langHello : WOString {
  value = session.localizer.hello;
}
```

```
langWorld : WOString {
    value = session.localizer.world;
}
```

(WOD file)

ERXLocalizer magically detects your system locale and puts out the the following HTML code if you use an english locale:

```
<p>The application says: Hello World</p>
```

Useful application settings

These are the default settings for ERXLocalizer:

```
er.extensions.ERXLocalizer.defaultLanguage=English
er.extensions.ERXLocalizer.fileNamesToWatch=( "Localizable.strings", "ValidationTemplate.strings" )
er.extensions.ERXLocalizer.availableLanguages=( English, German )
er.extensions.ERXLocalizer.frameworkSearchPath=( app, ERDirectToWeb, ERExtensions )
```

You can provide your own defaults in your application properties file or set the defaults somewhere in the initialization code of your application, e.g.

```
public Application() {
    NSLog.out.appendln("Welcome to " + this.name() + " !");
    /* ** put your initialization code in here ** */
    ERXLocalizer.setDefaultLanguage("English");
}
```

Unicode support

By setting this property, your Wonder application will be correctly encoded in UTF-8 throughout.

```
er.extensions.ERXApplication.DefaultEncoding=UTF-8
```

The following paragraph may no longer apply:

If you are creating UTF-8 applications, please be aware that UTF-8 support for your Localizable.strings files seems not to work. Instead of being desperate, simply convert your Localizable.strings files to UTF-16. The localized strings are shown in the correct way in your UTF-8 application. For further information see [Localizing Property Labels](#) in the Apple Developer Connection.

ERJavaMail support

By setting this property, email sent via ERJavaMail will be correctly encoded in UTF-8.

```
er.javamail.defaultEncoding = UTF-8
```