

[readthedocs.org](http://readthedocs.org)

---

# Flask-Babel 简介

---

最近用 [Flask](#) 给公司做了个小 **web** 应用，做的时候用英文了，现在要求翻译成中文。查看官方文档，发现有些内容讲得不甚详细，尤其缺乏一个完整的 **walkthrough**。于是我把自己的翻译流程大致总结一下，贴在这里供大家分享。由于我的 **app** 比较简单，本文也不会涉及到获取 **locale** 以及显示时间格式等话题，详细信息还是请多读官方文档和实际案例，使用 [google code](#) 搜索 `flaskext.babel` 可以看到很多案例应用。

## 安装 Flask-Babel

[Flask-Babel](#) 是 **Flask** 的翻译扩展工具。安装命令如下：

安装它的时候会顺便安装 [Babel](#)、[pytz](#)、[speakeasy](#) 这三个包，其中 **Babel** 是 **Python** 的一个国际化工具包。[pytz](#) 是处理时区的工具包，[speakeasy](#) 相当于是 **Babel** 的一个辅助工具，我们这里集中在翻译流程上，这几个工具就供以后进一步了解吧。

## Hello, World

接下来我们做一个简单的 **Hello World** 程序，新建一个叫 **hello** 的文件夹，在其中创建一个叫 **hello.py** 的文件，内容如下：

```
from flask import Flask, render_template</code>

app = Flask(__name__)

@app.route('/')
def hello():
    day = "Saturday"
    return render_template('index.html', day=day)

if __name__ == '__main__':
    app.debug = True
    app.run()
```

然后在 **hello.py** 的同一级目录下创建一个叫 **templates** 的文件夹，在其中写一个 **index.html**，内容如下：

```
<p>Hello, world!</p>
<p>It's {{ day }} today.</p>
```

很简单的 **Hello World** 程序，接下来我们要做的是让这个站变成中文站。

## 更新程序和模板¶

再接下来就是翻译了。翻译需要用到 **flask-babel** 这个 **flask** 扩展。首先我们将这个 **app** “国际化”，为模板和 **.py** 文件中的每一个字符串添加一个 **gettext** 函数，由于 **gettext** 函数被引用的次数太多了，为了方便手写，就将其 **import** 为 “\_”：

```
from flask import Flask, render_template</code>

from flaskext.babel import Babel, gettext as _

app = Flask(__name__)
app.config['BABEL_DEFAULT_LOCALE'] = 'zh_CN'
babel = Babel(app)

@app.route('/')
def hello():
    s = _("Saturday")
    return render_template('index.html', day=day)

if __name__ == '__main__':
    app.debug = True
    app.run()
```

然后修改模板：

```
<p>{{ _("Hello, world!") }}</p>
<p>{{ _("It's %(day)s today", day=day) }}</p>
```

你可以注意到我们对 **app** 的 **locale** 做了配置，然后用 **babel** 扩展将 **app** 再次初始化，并且将 **.py** 和 **.html** 中的字符串做了配置，让它们都使用 **gettext** 这个函数。其中值得注意的是 **gettext** 的格式化字符串的参数。如果直接用类似 **"It's %s today" % day** 是不行的。

这么一来，**app** 的语言其实是被写死成中文了。其实你可以在 **flask** 程序中让用户选择自己喜好的语言，或者依据浏览器设置用户优先显示的语言，详细做法可以参考官方文档中提到 **localeselector** 的部分。

## 设置 Babel¶

接下来我们要做的是 **babel** 的配置。在 **hello.py** 的同级目录创建一个叫 **babel.cfg** 的文件，内容如下：

```
[python: *.py]
```

```
[jinja2: **/templates/**/*.html]
extensions=jinja2.ext.autoescape,jinja2.ext.with_
```

## 生成翻译模板

这样 **babel** 就知道要从哪些位置搜索要翻译的字符串了。然后我们用 **pybabel** 生成要翻译的 **PO** 模板文件，这个命令是 **babel** 这个工具包带来的，生成翻译模板命令如下：

```
$ pybabel extract -F babel.cfg -o messages.pot .
```

注意结尾的点“.”，这个点表示当前目录，目录是 **pybabel** 必须的参数，官方文档中缺少这个点，所以命令是无法执行成功的。**messages.pot** 就是我们生成的翻译模板文件，内容大致如下：

```
# Translations template for PROJECT.
# Copyright (C) 2011 ORGANIZATION
# This file is distributed under the same license as the PROJECT project.
# FIRST AUTHOR , 2011.
#
#, fuzzy
msgid ""
msgstr ""

"Project-Id-Version: PROJECT VERSION\n"
"Report-Msgid-Bugs-To: EMAIL@ADDRESS\n"
"POT-Creation-Date: 2011-07-26 15:39+0800\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME \n"
"Language-Team: LANGUAGE \n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=utf-8\n"
"Content-Transfer-Encoding: 8bit\n"
"Generated-By: Babel 0.9.6\n"</code>

#: hello.py:11
msgid "Saturday"
msgstr ""

#: templates/index.html:1
msgid "Hello, world!"
msgstr ""

#: templates/index.html:2
#, python-format
msgid "It's %(day)s today"
msgstr ""
```

你可以修改里边头文件的信息，把个人和项目相关的资料加进去。

## 翻译

接下来我们创建中文翻译：

```
$ pybabel init -i messages.pot -d translations -l zh_CN
```

这句命令会在 **hello** 文件夹中生成一个 **translations** 文件夹，要确保 **flask** 能找到翻译内容，**translations**文件夹要和 **templates** 文件夹在同一个目录中。接下来我们就可以进行翻译了，修改 *translations/zh\_CN/LC\_MESSAGES/messages.po* 文件，将其中的内容翻译过来：

```
# Chinese (China) translations for PROJECT.
# Copyright (C) 2011 ORGANIZATION
# This file is distributed under the same license as the PROJECT project.
# FIRST AUTHOR , 2011.
#
msgid ""
msgstr ""
"Project-Id-Version: PROJECT VERSION\n"
"Report-Msgid-Bugs-To: EMAIL@ADDRESS\n"
"POT-Creation-Date: 2011-07-26 15:39+0800\n"
"PO-Revision-Date: 2011-07-26 09:07+0800\n"
"Last-Translator: FULL NAME \n"
"Language-Team: zh_CN \n"
"Plural-Forms: nplurals=1; plural=0\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=utf-8\n"
"Content-Transfer-Encoding: 8bit\n"
"Generated-By: Babel 0.9.6\n"</code>

#: hello.py:11
msgid "Saturday"
msgstr "星期六"

#: templates/index.html:1
msgid "Hello, world!"
msgstr "哈罗，世界！"

#: templates/index.html:2
#, fuzzy, python-format
msgid "It's %(day)s today"
msgstr "今天是%(tag)s"
```

PO文件的翻译其实可以用专门的工具来编辑，比如 [Poedit](#)，不过小文件直接手译就可以了。

## 编译翻译结果¶

翻译完后执行下面的命令，为其编译出 **message.mo** 文件：

```
$ pybabel compile -d translations
```

如果上述命令无法生成 **messages.mo** 文件，那你需要将 **message.po** 中的 `#, fuzzy` 删除。

然后就算基本完成了。这时执行 **python hello.py** 就会看到翻译的中文页面了。

## 更新翻译¶

有时我们需要对程序和模板做修改，翻译也要随之更新。更新后需要用前面的命令重新生成 **messages.pot** 文件，然后使用下面的命令将更新的内容 **merge** 到原来的翻译中：

```
$ pybabel update -i messages.pot -d translations
```

最后再到对应 **locale** 的文件夹下更新翻译并 **compile** 即可。

---

### Original URL:

<http://readthedocs.org/docs/translations/en/latest/flask-babel.html>