



# II Websites Team

Robert Mayr (robyduck)



Cos'è il Websites Team?

Di cosa si occupa?

La struttura del sistema e il suo build

Come posso contribuire?





fedora<sup>f</sup><sup>TM</sup>

**Cos'è il Websites Team?**

...un gruppo di appassionati  
che sanno utilizzare...

- Html – Css
- Php
- Python
- Genshi

... e che hanno l'obiettivo di...

- Consolidare i siti del Fedoraproject in uno schema uniforme
- Mantenere i contenuti che non rientrano in alcun Sotto-Progetto
- Rendere i siti gradevoli ed eccitanti come il progetto che rappresentano

fedora 

Ambass

Bugzap

Docs

Package

Epel

Marketing

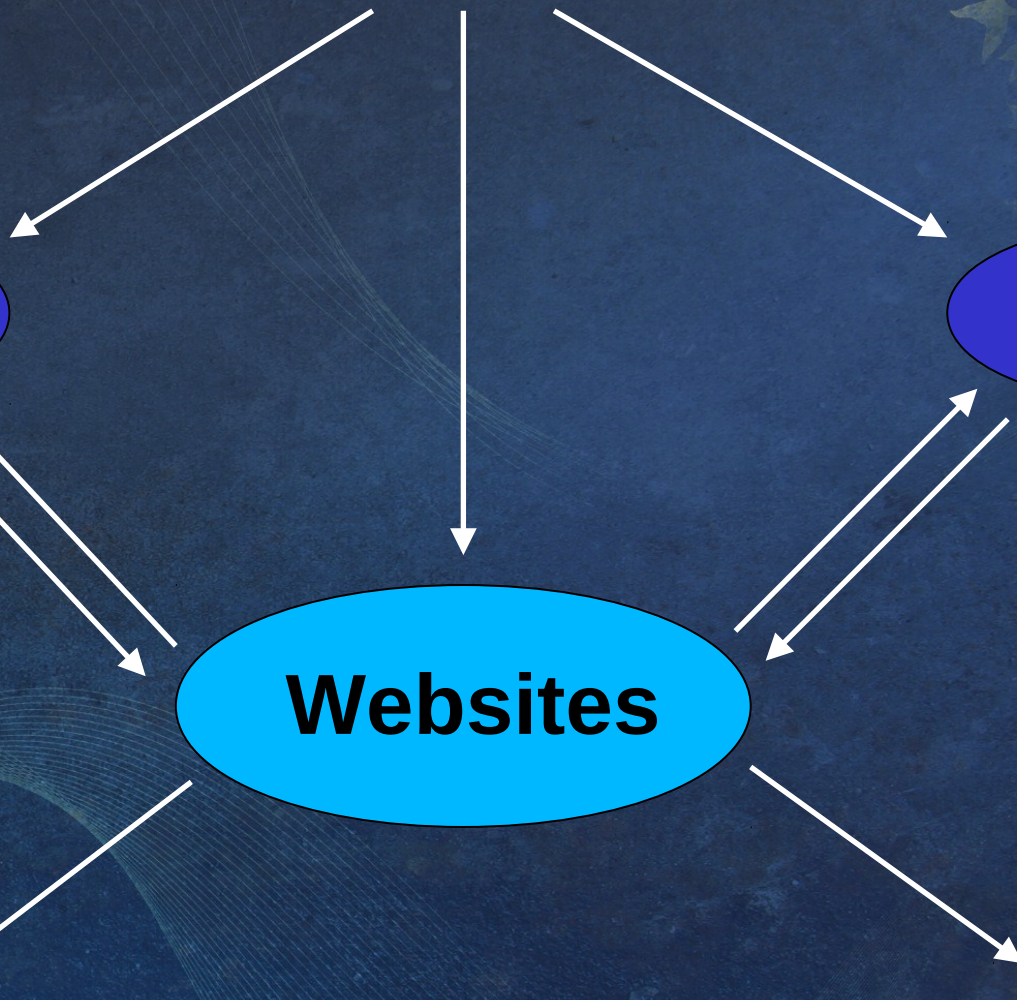
Infrastructure

Design

Websites


Localization

Intern.tion



Di cosa si occupa  
il  
Websites Team?



- Manutenzione di tutti i siti con dominio fedoraproject.org
- Collaborazione stretta con il Team Infra 



# [ Come funziona? ]

Architettura e build dei siti mantenuti



In ogni singolo subdomain l'albero delle directory attualmente è fatto nel seguente modo: ▶

```
.  
|-- boot.fedoraproject.org  
|   |-- build  
|   |-- ChangeLog  
|   |-- data  
|   |-- httpd  
|   |-- Makefile  
|   |-- po  
|   `-- static  
`-- TODO
```

# i18n

Utilizziamo GENSHI per l'internazionalizzazione. Alcuni markup di esempio sono i seguenti:

1. `${_('String')}`

2. `${Markup(_('String with &gt; html code'))}`

3. `${Markup(_('String with a &lt;a href="%s"&gt;link&lt;/a&gt;') %  
'http://fedoraproject.org')}`

4. `${_('%(size)s, DVD ISO disc image for %(arch)s-bit PC') % {'size':'3.1 GB', 'arch':'32'}}`

1. E' l'esempio più semplice;

2. Un esempio con un markup che può essere utilizzato dappertutto, con codice html!

3. Esempio con una variabile: comodo per i traduttori per non dover tradurre tutte le stringhe, ma soltanto una variabile.

4. La struttura più complessa in cui si lavora con più di una variabile. Il traduttore può cambiare l'ordine delle variabili utilizzando il loro nome.

Per il test si utilizza "make en test" per costruire la pagina in lingua inglese (evitando così errori dei traduttori)

Se corretto si costruisce il nuovo template file (POT) per i traduttori con:

**\$ make pot**

# l10n

Tutti I file PO vengono scaricati ogni ora.

Ogni volta che cambia il contenuto HTML è necessario aggiornare il template POT, che deve essere caricato nella piattaforma dei traduttori, transifex.com.

## **\$ make pushpot**

Per aggiungere una nuova lingua bisogna aggiornare il file LINGUAS, il file translation.py e il language.conf file. Vediamo l'albero:

```
.
|-- boot.fedoraproject.org
|   |-- build
|       |-- translations.py
|   |-- httpd
|       |-- conf
|           |-- languages.conf.in
|-- po
|-- LINGUAS
-- l10n_update.sh
```

Lo script "l10N\_update.sh" è utilizzato per creare un nuovo file LINGUAS, per caricare un nuovo POT e per fare un backup dei POT.

Questo processo attualmente è in fase di aggiornamento per ottenere un unico translation.py file e un languages.conf.in file.

La nuova architettura invece avrà un albero  
leggermente diverso: (ancora in fase di definizione)

```
.  
|-- boot.fedoraproject.org  
    |-- build  
    |-- data  
    |-- httpd  
    |-- Makefile  
    |-- po  
    \-- static
```

```
...  
|-- build  
|-- Makefile  
|-- static  
|-- templates  
 \-- TODO
```

# Build

- Tutti i siti vengono ricostruiti ogni ora con uno script `syncStatic`
- Solo il gruppo Infra può cambiare lo script e decidere con quale ramo di sviluppo costruire i siti
- Vengono costruiti anche siti in produzione, come `stg.fedoraproject.org` e `spins.stg.fedoraproject.org` ▶ ▶
- La ricostruzione di tutti i siti impiega circa 30-35 minuti

fedora<sup>™</sup> 

**Facciamo pratica**



# Prepariamo il sistema

- 1) Configuriamo il sistema e cloniamo il GIT repository del Team Websites

```
su -c 'yum shell'
```

```
> install git gettext python-genshi python-setuptools babel python-feedparser
```

```
> groupinstall 'Web Server'
```

```
> run
```

```
$ git config --global user.name "Robert Mayr"
```

```
$ git config --global user.email robyduck@fedoraproject.org
```

```
$ git config --global color.ui auto
```

```
$ cd $HOME
```

```
$ git clone git://git.fedorahosted.org/git/fedora-web.git
```



# Creazione di un ramo

- 1) Impostiamo un ramo di sviluppo locale e verificiamo di aver sincronizzato tutto

```
$ cd fedora-web/fedoraproject.org  
$ git checkout -b ramosviluppo
```

- 2) Si può anche lavorare su un ramo esistente, senza dover sovrascrivere nulla, per esempio:

```
$ git checkout -b ramosviluppo --track origin/f18-beta
```

```
$ git remote show origin  
$ git branch -a  
$ git status
```

# Test delle modifiche

- 1) Fatte le modifiche vediamo se la costruzione del sito va a buon fine:

```
$ cd fedora-web/fedoraproject.org  
$ make  
$ make test  
$ make stoptest
```

- 2) Se tutto procede senza errori si è pronti per l'invio delle modifiche:

```
$ git add data/content/file1.html data/content/file2.html ...  
$ git commit -am 'la mia prima modifica'
```

# Invio delle modifiche

- 1) Per i primi invii dobbiamo creare una patch:

```
$ cd fedora-web/fedoraproject.org  
$ git format-patch origin/master
```

La patch può essere inviata ai website-ticket, da dove gli amministratori la applicheranno.

- 2) Ricevuti i permessi di scrittura sui server invece inviamo direttamente:

```
$ git push
```

Come contribuire al

fedora<sup>f</sup>  
Websites Team



# Primi passi

- Iscriverti alla Mailing List del Websites Team e inviare una presentazione di se stessi
- Creazione di un account FAS con relativa pagina Wiki
- Richiedere l'iscrizione al gruppo Websites
- Seguire il canale IRC #fedora-websites su freenode

# Cominciamo a contribuire

- Utilizzare i ticket Easyfix per i primi lavori
- Inviare le proprie patch
- Informare il team della patch linkandola sulla ML
  
- Richiedere la sponsorizzazione per il gruppo gitfedora-web