fabric_colors Documentation

Release 0.9.44

Calvin Cheng

June 18, 2013

CONTENTS

Contents:

CHAPTER

ONE

DB PACKAGE

1.1 db Package

fabric_colors.db.**sphinx_sanity_check**()
This is where our database fabric functions exist.

1.2 mongodb Module

fabric_colors.db.mongodb.**initializer**
Enable initializer for mongodb

fabric_colors.db.mongodb.**install**
Install mongodb and start it if it isn't running.

fabric_colors.db.mongodb.**installed**
Check if mongodb is installed.

fabric_colors.db.mongodb.**restart**
Restart mongodb

fabric_colors.db.mongodb.**start**
Start mongodb

fabric_colors.db.mongodb.**status**
Is mongodb already running?

fabric_colors.db.mongodb.**stop**
Stop mongodb

1.3 postgis Module

fabric_colors.db.postgis.**create_template**

fabric_colors.db.postgis.**install**
Distro-agnostic way of installing postgis2.

fabric_colors.db.postgis.**installed**
Distro-agnostic way of checking if postgis2 is installed.

1.4 postgresql Module

fabric_colors.db.postgresql.**chk_data_dir**
Distro-agnostic checking postgresql's data directory

fabric_colors.db.postgresql.**chk_db**
Checks if such a database exist.

fabric_colors.db.postgresql.**create_db**

fabric_colors.db.postgresql.**enable_postgis**(**args*, ***kwargs*)

fabric_colors.db.postgresql.**initializer**
Distro-agnostic: set up initializations script for postgresql

fabric_colors.db.postgresql.**install**
Distro-agnostic way of installing postgresql.

fabric_colors.db.postgresql.**installed**
Distro-agnostic way of checking if postgresql is installed.

fabric_colors.db.postgresql.**pg_hba_conf**(**args*, ***kwargs*)
Set our pg_hba.conf to our preferred defaults.

fabric_colors.db.postgresql.**postgresql_conf**(**args*, ***kwargs*)
In postgresq.conf, toggle listen to * or specific IP or commented out

fabric_colors.db.postgresql.**restart**
Distro-agnostic: restart postgresql

fabric_colors.db.postgresql.**setup_data_dir**
Distro-agnostic: set up postgresql data directory

fabric_colors.db.postgresql.**start**
Distro-agnostic: start postgresql

fabric_colors.db.postgresql.**status**
Distro-agnostic checking postgresql status

fabric_colors.db.postgresql.**stop**
Distro-agnostic: stop postgresql

CHAPTER

TWO

DEPLOY PACKAGE

2.1 deploy Package

fabric_colors.deploy.**deploy**
Usage: *fab -R all deploy* or fab -H mysite.com deploy‘. Execute a deployment to the given groups of hosts or host

fabric_colors.deploy.**pip_package_preinstall**(**args*, ***kwargs*)
Preinstall packages listed in package_exceptions if they are found in requirements.txt! Packages listed in package_exceptions need to have their setup.py executed beforehand.

fabric_colors.deploy.**pip_requirements**
Usage: *fab -R all deploy.pip_requirements*. Install the required python packages from the requirements.txt file using pip

fabric_colors.deploy.**prepare_deploy_env**(*target*)
Make sure that we have the release directories before executing deployment.

fabric_colors.deploy.**test_host_check**
Usage: *fab -R all deploy.test_host_check* or *fab -H mysite.com deploy.test_host_check*

2.2 git Module

fabric_colors.deploy.git.**git_archive_and_upload_tar**
Create an archive from the current git branch and upload it to target machine.

fabric_colors.deploy.git.**git_branch_check**
Usage: *fab -R all deploy.git_branch_check*. Check that we are on master branch before permitting deploy

2.3 releases Module

fabric_colors.deploy.releases.**cleanup**
Usage: *fab -R all releases.cleanup*. Ensure that target node only has *n* most recent deployed directories, where *n* by default is 10 but can be overridden by that node's settings in fabsettings.

fabric_colors.deploy.releases.**manage_release**(*description=''*, *disable_newrelic=False*)
Helper function to annotate env.release and env.release_notes; and optionally update env.newrelic and register event on newrelic.

fabric_colors.deploy.releases.**showlist**
Usage: *fab -R all releases.showlist*. Returns a list of deploy directories.

fabric_colors.deploy.releases.**symlink_check**
Usage: *fab -R all releases.symlink_check*. Check the hash pointing to the current symlink.

fabric_colors.deploy.releases.**symlink_current**
Usage: *fab -R all releases.symlink_current*. Symlink our current release.

CHAPTER

THREE

ARCH PACKAGE

3.1 `arch` Package

3.2 `postgis` Module

3.3 `postgresql` Module

CHAPTER

FOUR

MAC PACKAGE

4.1 `postgis` Module

4.2 `postgresql` Module

CHAPTER

FIVE

DISTRO PACKAGE

5.1 distro Package

fabric_colors.distro.**distro**
Usage: *fab -R all distro*. Determine the distro of given target host(s).

5.2 server Module

fabric_colors.distro.server.**chk_exists**
Usage: *fab -R dev server.chk_exists:path*. Returns 1 or empty string.

fabric_colors.distro.server.**get_ownership**
Usage: *fab -R all server.get_ownership*. Given a path, and the target node, return owner and group for that directory

fabric_colors.distro.server.**groupadd**
Usage: *fab -R dev server.groupadd:sudo*. Adds the given group.

fabric_colors.distro.server.**groups**
Usage: *fab -R dev server.groups*. Returns available groups.

fabric_colors.distro.server.**initializers**
Shows us which are the services that will be initialized automatically after a reboot.

fabric_colors.distro.server.**setup**
Usage: *fab -R all server.setup*. Runs all scripts as given username, in sudo mode, on the target server.

fabric_colors.distro.server.**setup_base**
Usage: *fab -R dev server.setup_base*. Installs the base requirements.

fabric_colors.distro.server.**setup_python**
Usage: *fab -R all server.setup_python*. Sets up the python environment

fabric_colors.distro.server.**show_sudo_users_and_groups**(*ug*, *nopasswd*)
Helper function that prints out users and groups with sudo (or no passwd sudo) rights.

fabric_colors.distro.server.**ssh_copy_id**
Usage: *fab -R all server.ssh_copy_id*. Add local machine's public key to target's username

fabric_colors.distro.server.**sudo_users_and_groups**
Usage: *fab -R dev server.sudo_users_and_groups:nopasswd*. nopasswd(optional)=True/False.

fabric_colors.distro.server.**useradd**
Usage: *fab -R all server.useradd:web*. Create user given the user name & target machine.

fabric_colors.distro.server.**userchk**
Usage: *fab -R all server.userchk:web*. Check if the given user name is available on the given target host(s).

fabric_colors.distro.server.**userdel**
Usage: *fab -R all server.userdel:web*. Delete user & its home directory given the user name & target machine.

fabric_colors.distro.server.**usersudo**
Usage: *fab -R dev server.usersudo:user,sudogroup*. Adds deploying user to the given sudo group.

fabric_colors.distro.server.**visudo**
Usage: *fab -R all server.visudo:sudogroup*. Enable the given sudo group in /etc/sudoers programmatically.

This is the same as typing *visudo* and manually commenting out a commented line for the group so it has ALL=(ALL) ALL access rights, i.e. sudo rights.

5.3 Subpackages

5.3.1 arch Package

arch Package

postgis Module

postgresql Module

5.3.2 mac Package

postgis Module

postgresql Module

CHAPTER

SIX

ENVIRONMENT PACKAGE

6.1 environment Package

fabric_colors.environment.**set_target_env**(*f*)
decorator function that dynamically sets the current host's env variables using _env_set(target)

Usage on a fabric function:

```
@set_target_env
def host_type():
    run('uname -s')
```

CHAPTER

SEVEN

MONITOR PACKAGE

7.1 newrelic Module

fabric_colors.monitor.newrelic.**newrelic**
Check if newrelic is sending data to rpm.newrelic.com's data collector

fabric_colors.monitor.newrelic.**record_deploy**
values is a dictionary containing key and value. Equivalent of this example (please use the correct API key of course) curl -H "x-api-key:f55da53a178959f9130381396b2172ac5e26cfa2383503c" -d "deployment[application_id]=2142298" -d "deployment[host]=localhost" -d "deployment[description]=This deployment was sent using curl" -d "deployment[revision]=1242" -d "deployment[changelog]=many hands make light work" -d "deployment[user]=Calvin Cheng" https://rpm.newrelic.com/deployments.xml

CHAPTER

EIGHT

FABRIC_COLORS PACKAGE

8.1 fabric_colors Package

8.2 _version Module

8.3 main Module

fabric_colors.main.**main**()
Main command-line execution loop.

8.4 quickstart Module

fabric_colors.quickstart.**create_fabfile**(*project_root*)
Creates a fabfile.py in the given project_root if it does not exist.

fabric_colors.quickstart.**quickstart**()
Run *fabc-quickstart* in your console to get started with fabric-colors!

fabric_colors.quickstart.**specify_nodes**(*name*, *nodes*)
Accepts the name of a new node and a dictionary of existing nodes. Returns the new node with user-provided domain name and ip address and a dictionary of all nodes.

8.5 Subpackages

8.5.1 db Package

db Package

fabric_colors.db.**sphinx_sanity_check**()
This is where our database fabric functions exist.

mongodb Module

fabric_colors.db.mongodb.**initializer**
Enable initializer for mongodb

fabric_colors.db.mongodb.**install**
Install mongodb and start it if it isn't running.

fabric_colors.db.mongodb.**installed**
Check if mongodb is installed.

fabric_colors.db.mongodb.**restart**
Restart mongodb

fabric_colors.db.mongodb.**start**
Start mongodb

fabric_colors.db.mongodb.**status**
Is mongodb already running?

fabric_colors.db.mongodb.**stop**
Stop mongodb

postgis Module

fabric_colors.db.postgis.**create_template**

fabric_colors.db.postgis.**install**
Distro-agnostic way of installing postgis2.

fabric_colors.db.postgis.**installed**
Distro-agnostic way of checking if postgis2 is installed.

postgresql Module

fabric_colors.db.postgresql.**chk_data_dir**
Distro-agnostic checking postgresql's data directory

fabric_colors.db.postgresql.**chk_db**
Checks if such a database exist.

fabric_colors.db.postgresql.**create_db**

fabric_colors.db.postgresql.**enable_postgis**(**args*, ***kwargs*)

fabric_colors.db.postgresql.**initializer**
Distro-agnostic: set up initializations script for postgresql

fabric_colors.db.postgresql.**install**
Distro-agnostic way of installing postgresql.

fabric_colors.db.postgresql.**installed**
Distro-agnostic way of checking if postgresql is installed.

fabric_colors.db.postgresql.**pg_hba_conf**(**args*, ***kwargs*)
Set our pg_hba.conf to our preferred defaults.

fabric_colors.db.postgresql.**postgresql_conf**(**args*, ***kwargs*)
In postgresq.conf, toggle listen to * or specific IP or commented out

fabric_colors.db.postgresql.**restart**
Distro-agnostic: restart postgresql

fabric_colors.db.postgresql.**setup_data_dir**
Distro-agnostic: set up postgresql data directory

fabric_colors.db.postgresql.**start**
Distro-agnostic: start postgresql

fabric_colors.db.postgresql.**status**
Distro-agnostic checking postgresql status

fabric_colors.db.postgresql.**stop**
Distro-agnostic: stop postgresql

8.5.2 deploy Package

deploy Package

fabric_colors.deploy.**deploy**
Usage: *fab -R all deploy* or fab -H mysite.com deploy‘. Execute a deployment to the given groups of hosts or host

fabric_colors.deploy.**pip_package_preinstall**(**args*, ***kwargs*)
Preinstall packages listed in package_exceptions if they are found in requirements.txt! Packages listed in package_exceptions need to have their setup.py executed beforehand.

fabric_colors.deploy.**pip_requirements**
Usage: *fab -R all deploy.pip_requirements*. Install the required python packages from the requirements.txt file using pip

fabric_colors.deploy.**prepare_deploy_env**(*target*)
Make sure that we have the release directories before executing deployment.

fabric_colors.deploy.**test_host_check**
Usage: *fab -R all deploy.test_host_check* or *fab -H mysite.com deploy.test_host_check*

git Module

fabric_colors.deploy.git.**git_archive_and_upload_tar**
Create an archive from the current git branch and upload it to target machine.

fabric_colors.deploy.git.**git_branch_check**
Usage: *fab -R all deploy.git_branch_check*. Check that we are on master branch before permitting deploy

releases Module

fabric_colors.deploy.releases.**cleanup**
Usage: *fab -R all releases.cleanup*. Ensure that target node only has *n* most recent deployed directories, where *n* by default is 10 but can be overridden by that node's settings in fabsettings.

fabric_colors.deploy.releases.**manage_release**(*description=''*, *disable_newrelic=False*)
Helper function to annotate env.release and env.release_notes; and optionally update env.newrelic and register event on newrelic.

fabric_colors.deploy.releases.**showlist**
Usage: *fab -R all releases.showlist*. Returns a list of deploy directories.

fabric_colors.deploy.releases.**symlink_check**
Usage: *fab -R all releases.symlink_check*. Check the hash pointing to the current symlink.

fabric_colors.deploy.releases.**symlink_current**
Usage: *fab -R all releases.symlink_current*. Symlink our current release.

8.5.3 distro Package

distro Package

fabric_colors.distro.**distro**
Usage: *fab -R all distro*. Determine the distro of given target host(s).

server Module

fabric_colors.distro.server.**chk_exists**
Usage: *fab -R dev server.chk_exists:path*. Returns 1 or empty string.

fabric_colors.distro.server.**get_ownership**
Usage: *fab -R all server.get_ownership*. Given a path, and the target node, return owner and group for that directory

fabric_colors.distro.server.**groupadd**
Usage: *fab -R dev server.groupadd:sudo*. Adds the given group.

fabric_colors.distro.server.**groups**
Usage: *fab -R dev server.groups*. Returns available groups.

fabric_colors.distro.server.**initializers**
Shows us which are the services that will be initialized automatically after a reboot.

fabric_colors.distro.server.**setup**
Usage: *fab -R all server.setup*. Runs all scripts as given username, in sudo mode, on the target server.

fabric_colors.distro.server.**setup_base**
Usage: *fab -R dev server.setup_base*. Installs the base requirements.

fabric_colors.distro.server.**setup_python**
Usage: *fab -R all server.setup_python*. Sets up the python environment

fabric_colors.distro.server.**show_sudo_users_and_groups**(*ug*, *nopasswd*)
Helper function that prints out users and groups with sudo (or no passwd sudo) rights.

fabric_colors.distro.server.**ssh_copy_id**
Usage: *fab -R all server.ssh_copy_id*. Add local machine's public key to target's username

fabric_colors.distro.server.**sudo_users_and_groups**
Usage: *fab -R dev server.sudo_users_and_groups:nopasswd*. nopasswd(optional)=True/False.

fabric_colors.distro.server.**useradd**
Usage: *fab -R all server.useradd:web*. Create user given the user name & target machine.

fabric_colors.distro.server.**userchk**
Usage: *fab -R all server.userchk:web*. Check if the given user name is available on the given target host(s).

fabric_colors.distro.server.**userdel**
Usage: *fab -R all server.userdel:web*. Delete user & its home directory given the user name & target machine.

fabric_colors.distro.server.**usersudo**
Usage: *fab -R dev server.usersudo:user,sudogroup*. Adds deploying user to the given sudo group.

fabric_colors.distro.server.**visudo**
Usage: *fab -R all server.visudo:sudogroup*. Enable the given sudo group in /etc/sudoers programmatically.

This is the same as typing *visudo* and manually commenting out a commented line for the group so it has ALL=(ALL) ALL access rights, i.e. sudo rights.

Subpackages

arch Package

arch Package

postgis Module

postgresql Module

mac Package

postgis Module

postgresql Module

8.5.4 environment Package

environment Package

fabric_colors.environment.**set_target_env**(*f*)
decorator function that dynamically sets the current host's env variables using _env_set(target)

Usage on a fabric function:

```
@set_target_env
def host_type():
    run('uname -s')
```

8.5.5 monitor Package

newrelic Module

fabric_colors.monitor.newrelic.**newrelic**
Check if newrelic is sending data to rpm.newrelic.com's data collector

fabric_colors.monitor.newrelic.**record_deploy**
values is a dictionary containing key and value. Equivalent of this example (please use the correct API key of course) curl -H "x-api-key:f55da53a178959f9130381396b2172ac5e26cfa2383503c" -d "deployment[application_id]=2142298" -d "deployment[host]=localhost" -d "deployment[description]=This deployment was sent using curl" -d "deployment[revision]=1242" -d "deployment[changelog]=many hands make light work" -d "deployment[user]=Calvin Cheng" https://rpm.newrelic.com/deployments.xml

8.5.6 tests Package

tests Package

class fabric_colors.tests.**testFabricColors**(*methodName='runTest'*)
Bases: unittest.case.TestCase

dummyTest()

setUp()
Set up data used in the tests. setUp is called before each test function execution.

8.5.7 utilities Package

utilities Package

fabric_colors.utilities.**chk_req**()
Usage *fab chk_req*. Check if the current requirements.txt file matches what is in user's virtualenv. Returns True or False.

fabric_colors.utilities.**info**
Usage: *fab -R dev info*. Show env details of target host "dev".

backups Module

fabric_colors.utilities.backups.**media_backup**(*target='localhost'*, *local_path=None*)
Usage: *fab media_backup:dev fab media_backup:dev,/path/to/backup/folder/*. Backups the target's static files to local. Backup path defaults to [cwd]/backups/static/[target]/.

fabric_colors.utilities.backups.**mysql_db**
usage: *fab -r <server_name> backups.mysql_db /path/to/backup/folder/* backups the target's database to local destination. backup path defaults to /backups/db/[target]/. 1. creates the target folder if doesnt exists 2. by default keeps 30 days of backups

fabric_colors.utilities.backups.**postgres_backup**(*target*, *local_path=None*)
Usage: *fab postgres_backup:dev fab postgres_backup:dev,/path/to/backup/folder/*. Backups the target's database to local. Backup path defaults to [cwd]/backups/db/[target]/.

django Module

fabric_colors.utilities.django.**collectstatic**
Usage: *fab -R dev django.collectstatic*. Run *python manage.py collectstatic* on specified target.

fabric_colors.utilities.django.**compilemessages**
Usage: *fab -R dev django.compilemessages*. Run *python manage.py compilemessages* on specified target.

fabric_colors.utilities.django.**create_public**
Usage: *fab -R dev django.create_public*. Create public directory for django media and static files in our localhost.

fabric_colors.utilities.django.**makemessages**
Usage: *fab -R dev django.makemessages:de*. Run *python manage.py makemessages* on specified target and language

emails Module

fabric_colors.utilities.emails.**email_on_success**(**args*, ***kwargs*)

8.5.8 virtualenv Package

virtualenv Package

fabric_colors.virtualenv.**chkvirtualenv**
Usage: *fab -R dev chkvirtualenv*. Check that we have the virtualenv for our project on the target machine.

fabric_colors.virtualenv.**lsvirtualenv**
Usage: *fab -R dev lsvirtualenv*. List all virtualenvs.

fabric_colors.virtualenv.**mkvirtualenv**
Usage: *fab -R all mkvirtualenv*. Create the virtualenv for our project on the target machine.

fabric_colors.virtualenv.**rmvirtualenv**
Usage: *fab -R dev rmvirtualenv:name_of_virtualenv*.

8.5.9 webserver Package

webserver Package

fabric_colors.webserver.**initializer**
Wrapper initializer

fabric_colors.webserver.**webserver**
command=start/stop/restart/restart_violent; defaults to graceful 'restart' newrelic(optional)=True/False. Wrapper function that restarts the webserver that is current in-use. Defaults to uwsgi.

nginx Module

fabric_colors.webserver.nginx.**initializer**
Enable initializer for nginx

fabric_colors.webserver.nginx.**install**
Install nginx and start it if it isn't running.

fabric_colors.webserver.nginx.**installed**
Check if nginx is installed.

fabric_colors.webserver.nginx.**restart**
Restart nginx

fabric_colors.webserver.nginx.**start**
Start nginx

fabric_colors.webserver.nginx.**status**
Is nginx already running?

fabric_colors.webserver.nginx.**stop**
Stop nginx

uwsgi Module

fabric_colors.webserver.uwsgi.**initializer**
Check if host has the current project's uwsgi initializer. If not, create it.

fabric_colors.webserver.uwsgi.**replace_all**(*text*, *dic*)

fabric_colors.webserver.uwsgi.**uwsgi**
Usage: *fab -R all uwsgi:command*. command=start/stop/restart/restart_violent/status, newrelic(optional)=True/False.

CHAPTER

NINE

TESTS PACKAGE

9.1 tests Package

class fabric_colors.tests.**testFabricColors**(*methodName='runTest'*)
Bases: unittest.case.TestCase

dummyTest()

setUp()
Set up data used in the tests. setUp is called before each test function execution.

CHAPTER

TEN

UTILITIES PACKAGE

10.1 utilities Package

fabric_colors.utilities.**chk_req**()
Usage *fab chk_req*. Check if the current requirements.txt file matches what is in user's virtualenv. Returns True or False.

fabric_colors.utilities.**info**
Usage: *fab -R dev info*. Show env details of target host "dev".

10.2 backups Module

fabric_colors.utilities.backups.**media_backup**(*target='localhost'*, *local_path=None*)
Usage: *fab media_backup:dev fab media_backup:dev,/path/to/backup/folder/*. Backups the target's static files to local. Backup path defaults to [cwd]/backups/static/[target]/.

fabric_colors.utilities.backups.**mysql_db**
usage: *fab -r <server_name> backups.mysql_db /path/to/backup/folder/* backups the target's database to local destination. backup path defaults to /backups/db/[target]/. 1. creates the target folder if doesnt exists 2. by default keeps 30 days of backups

fabric_colors.utilities.backups.**postgres_backup**(*target*, *local_path=None*)
Usage: *fab postgres_backup:dev fab postgres_backup:dev,/path/to/backup/folder/*. Backups the target's database to local. Backup path defaults to [cwd]/backups/db/[target]/.

10.3 django Module

fabric_colors.utilities.django.**collectstatic**
Usage: *fab -R dev django.collectstatic*. Run *python manage.py collectstatic* on specified target.

fabric_colors.utilities.django.**compilemessages**
Usage: *fab -R dev django.compilemessages*. Run *python manage.py compilemessages* on specified target.

fabric_colors.utilities.django.**create_public**
Usage: *fab -R dev django.create_public*. Create public directory for django media and static files in our localhost.

fabric_colors.utilities.django.**makemessages**
Usage: *fab -R dev django.makemessages:de*. Run *python manage.py makemessages* on specified target and language

10.4 emails Module

fabric_colors.utilities.emails.**email_on_success**(**args*, ***kwargs*)

CHAPTER

ELEVEN

VIRTUALENV PACKAGE

11.1 `virtualenv` Package

fabric_colors.virtualenv.**chkvirtualenv**
Usage: *fab -R dev chkvirtualenv*. Check that we have the virtualenv for our project on the target machine.

fabric_colors.virtualenv.**lsvirtualenv**
Usage: *fab -R dev lsvirtualenv*. List all virtualenvs.

fabric_colors.virtualenv.**mkvirtualenv**
Usage: *fab -R all mkvirtualenv*. Create the virtualenv for our project on the target machine.

fabric_colors.virtualenv.**rmvirtualenv**
Usage: *fab -R dev rmvirtualenv:name_of_virtualenv*.

CHAPTER

TWELVE

WEBSERVER PACKAGE

12.1 webserver Package

fabric_colors.webserver.**initializer**
Wrapper initializer

fabric_colors.webserver.**webserver**
command=start/stop/restart/restart_violent; defaults to graceful 'restart' newrelic(optional)=True/False. Wrapper function that restarts the webserver that is current in-use. Defaults to uwsgi.

12.2 nginx Module

fabric_colors.webserver.nginx.**initializer**
Enable initializer for nginx

fabric_colors.webserver.nginx.**install**
Install nginx and start it if it isn't running.

fabric_colors.webserver.nginx.**installed**
Check if nginx is installed.

fabric_colors.webserver.nginx.**restart**
Restart nginx

fabric_colors.webserver.nginx.**start**
Start nginx

fabric_colors.webserver.nginx.**status**
Is nginx already running?

fabric_colors.webserver.nginx.**stop**
Stop nginx

12.3 uwsgi Module

fabric_colors.webserver.uwsgi.**initializer**
Check if host has the current project's uwsgi initializer. If not, create it.

fabric_colors.webserver.uwsgi.**replace_all**(*text*, *dic*)

fabric_colors.webserver.uwsgi.**uwsgi**
Usage: *fab -R all uwsgi:command*. command=start/stop/restart/restart_violent/status, newrelic(optional)=True/False.

CHAPTER

THIRTEEN

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

f