



ROMA, 3/4 MARZO 2017
DRUPALDAY

<http://www.drupalday.it>
#drupalday #ddroma17

.wellnet[®]
web-being


SPARKFABRIK

b DIGITAL FACTORY
meme

ROBO

MODERN TASK RUNNER FOR PHP

LUCA CRACCO

Freelance & Drupal Developer

@luca_cracco

github.com/lucacracco



CHE COS'È ROBO

- Robo è un **task runner**
- E' scritto in PHP con una programmazione orientata agli oggetti
- Ha a disposizione moltissimi task/comandi per lo sviluppo, test, distribuzione, ecc..

TASK RUNNERS

Sono tools che permettono di automatizzare le operazioni più ricorrenti; sia semplici operazioni di minificazione del css sia compilazione e pubblicazione di codice

Tra i più conosciuti troviamo **Buildbot**, **Jenkins**, .. ma anche **Grunt**, **Gulp**, ...

TASK RUNNER PHP

- Task PHP
- BLDR
- Rocketter

USARE ROBO PER

- Eseguire comandi
- Lanciare test
- Automatizzare le operazioni più comuni
- Eseguire attività parallele
- Percepire cambiamenti sul filesystem
- Inizializzare ambienti e progetti

..PERCHÈ ROBO?

- E' scritto in PHP ad oggetti
- Supporto per i comandi: robo list
- Comandi con convalida, colori ANSI, ecc..
- Si può installare con Composer per il progetto
- Vi sono un sacco di task già pronti

INSTALLAZIONE

COMPOSER

Aggiungendolo al *composer.json* del progetto

```
require-dev: {  
    "consolidation/robo": "^1.0.0"  
}
```

o installandolo globalmente

```
composer global require consolidation/robo
```

INSTALLAZIONE

PHAR

```
wget http://robo.li/robo.phar
```

o scaricandolo via HTTPS da [GitHub releases](#)

Oppure installandolo in /usr/bin

```
sudo chmod +x robo.phar && mv robo.phar /usr/bin/robo
```

PER INIZIARE...

Creiamo il nostro **Robofile.php** con

```
cd mio_progetto  
robo init
```

Ed ecco il file:

```
/**  
 * This is project's console commands  
 * configuration for Robo task runner.  
 *  
 * @see http://robo.li/  
 */  
class RoboFile extends \Robo\Tasks  
{  
    // define public methods as commands  
}
```

COMANDI

```
class RoboFile extends \Robo\Tasks {  
  
    function hello($world) {  
        $this->say("Hello, $world");  
    }  
}
```

Eseguendo:

```
robo hello Paolo  
→ Hello, Paolo
```

ARGOMENTI

```
class RoboFile extends \Robo\Tasks {  
  
    function helloOne($world = "Francesco") {  
        $this->say("Hello, $world");  
    }  
  
    function helloMulti(array $world) {  
        $this->say("Hello, " . implode(', ', $world));  
    }  
}
```

Eseguendo:

```
robo hello:one  
→ Hello, Francesco
```

```
robo hello:multi Paolo Francesco  
→ Hello, Paolo Francesco
```

OPZIONI

```
function hello($opts = ['silent' => false]) {  
    if (!$opts['silent']) $this->say("Hello, world");  
}
```

Eseguendo:

```
robo hello  
→ Hello, world
```

```
robo hello --silent  
→
```

ROBO TASKS

Base Tasks, FileSystem Tasks, File Tasks, Composer Tasks, VCS Tasks, Development Tasks, Assets Tasks, Archive Tasks, SSH, Rsync Tasks, Testing Tasks, NPM Tasks, Bower Tasks, Gulp Tasks, Docker Tasks, ApiGen Tasks, e molti altri in [Packagist.org](https://packagist.org)

INCLUDERE UN TASK

Aggiungerlo al progetto

```
composer require boedah/robo-drush
```

Richiamarlo nel RoboFile

```
class RoboFile extends \Robo\Tasks {  
  
    use \Boedah\Robo\Task\Drush\loadTasks;  
  
    public function test() {  
        // ...  
    }  
}
```


CREARE UN TASK

TRAIT LOADTASKS.PHP

```
namespace MyAssetTasks;

trait loadTasks {

    /**
     * Example task to compile assets
     *
     * @param string $pathToCompileAssets
     * @return \MyAssetTasks\CompileAssets
     */
    protected function taskCompileAssets($path = null) {
        // Always construct your tasks with the `task()` task builder.
        return $this->task(CompileAssets::class, $path);
    }
}
```

CREARE UN TASK

TASK IMPLEMENTATION

```
class CompileAssets implements \Robo\Contract\TaskInterface {

    // configuration params
    protected $path;
    protected $to;
    function __construct($path) {
        $this->path = $path;
    }

    function to($filename){
        $this->to = $filename;
        // must return $this
        return $this;
    }

    // must implement Run
```

IO: ROBO\OUTPUT

Stampare messaggi

```
$this->say("Hello");
```

Oppure interagire:

```
$name = $this->ask("What is your name?");
```

TASK VS STACK

Le classi Stack contengono un insieme di task da eseguire uno dopo l'altro. Inoltre può essere configurato per bloccarsi al primo errore che incontra senza proseguire con gli altri task.

```
$this->stopOnFail(true);
```

STACK VS COLLECTION

Robo offre delle *collection* di task che permettono il rilevamento degli errori e il recupero/rollback. Se uno dei task della *collection* va in errore, l'operazione viene interrotta; attività di rollback può essere quindi eseguita per ripristinare il sistema allo stato originale.

ROBOFILES MULTIPLI

E' possibile fornire il maggior numero di classi di comandi (Robofile) direttamente al costruttore **RoboRunner**

```
$commandClasses = [  
    \MyProject\Commands\BuildCommands::class,  
    \MyProject\Commands\DeployCommands::class  
];  
$statusCode = \Robo\Robo::run(  
    $_SERVER['argv'],  
    $commandClasses,  
    'MyAppName',  
    '0.0.0-alpha0'  
);  
exit($statusCode);
```

DRUPAL8 & COMPOSER

- Usare Composer con Drupal
- Composer template per progetti Drupal

ROBO & DRUPAL

Integrazione Robo con Drush:

[boedah/robo-drush](https://github.com/boedah/robo-drush)

```
$this->taskDrushStack()  
  ->drupalRootDirectory('/var/www/html/some-site')  
  ->uri('sub.example.com')  
  ->maintenanceOn()  
  ->updateDb()  
  ->revertAllFeatures()  
  ->maintenanceOff()  
  ->run();
```


ROBO DRUPAL8 BUILDER

- Robo
- Drush
- Tasks Robo/Drush
- Yaml Files → Configurations

Work in progress..

github.com/lucacracco/robo-drupal8

YAML CONFIGURATIONS

```
# Enviroment
environment: local

# Base Path
base_path: /var/www/html

# Path to drush
drush_path: ./vendor/bin/drush

# Domani used
domain: 'http://test.dd'

# Database Configuration
database:
  url: 'mysql://drupal:drupal@mysql_test:3306/drupal'
  prefix: ''
```

TASKS CUSTOM

```
public function deploy($opts = self::OPTS) {  
  
    $this->say("Deploy");  
  
    // Init parameters.  
    $this->init($opts['environment'], $opts['site']);  
  
    // Execute functions.  
    $this->backupDatabase();  
    $this->importConfig();  
    $this->rebuildCache();  
    $this->entityUpdates();  
    $this->coreCron();  
}
```

ROBO COMMANDS

```
→ vendor/bin/robo list  
Robo version 1.0.5
```

Available commands:

deploy	Deploy.
help	Displays help for a command
list	Lists commands
build	
build:conf	Build a site from configuration files.
build:from-database	Build an existing site by importing the data.
build:new	Build a site from scratch.
configuration	
configuration:export	Export configuration after clear cache.
configuration:import	Import configuration.
features	
features:import	Import features

docker exec -i -t -u www-data example_web_1 bash



File Modifica Visualizza Cerca Terminale Aiuto

www-data@d6a6d89ae6a6:~/html\$

DRUPALDAY

ROMA 2017

ORGANIZZATO DA

.wellnet[®]
web-being


SPARKFABRIK

DIGITAL FACTORY
bmeme



3/4 MARZO, SAPIENZA UNIVERSITÀ DI ROMA

DRUPALDAY

con il Patrocinio e la collaborazione di



SAPIENZA
UNIVERSITÀ DI ROMA

Il DRUPALDAY è stato possibile grazie a

Gold Sponsor



Silver Sponsor



Media Sponsor

