

# PHP (9)



Marina Ribaudó, [marina.ribaudó@unige.it](mailto:marina.ribaudó@unige.it)

# cURL

2

- <https://curl.se/>
- *“curl is used in command lines or scripts to **transfer data**. It is also used in cars, television sets, routers, printers, audio equipment, mobile phones, tablets, settop boxes, media players and is the internet transfer backbone for thousands of software applications affecting billions of humans daily.”*

# cURL

3

- Contrazione di “Client for URLs”, cURL è un software open-source ampiamente utilizzato per l'automazione di richieste e risposte sulla rete Internet
- Permette di **inviare e ricevere dati** da e verso server remoti
- Si possono **personalizzare le richieste HTTP**, specificando il metodo, gli header, il body
- Si possono **gestire le risposte HTTP**, specificando se l'output deve essere restituito a schermo oppure su file, o ancora se deve essere passato ad un altro processo

# PHP cURL

4

- cURL is a PHP extension, that allows us to receive and send information. By doing so, cURL makes it easy to communicate between different websites and domains
- cURL non sempre è installato di default, va verificato

# Scaricare una pagina web

5

```
<?php
    $handle = curl_init();
    $url = "...url qui...";
    // Set the url
    curl_setopt($handle, CURLOPT_URL, $url);
    // Set the result output to be a string
    curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
    $output = curl_exec($handle);
    curl_close($handle);
    echo $output;
?>
```

# Scaricare una pagina web

6

```
<?php
```

```
$handle = curl_init();
```

```
$url = "...url"
```

```
// Set the url
```

```
curl_setopt($
```

```
// Set the res
```

```
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
```

```
$output = curl_exec($handle);
```

```
curl_close($handle);
```

```
echo $output;
```

```
?>
```

Inizializza una **nuova sessione curl** e restituisce un handle che viene usato con le funzioni `curl_setopt()`, `curl_exec()`, and `curl_close()`

# Scaricare una pagina web

7

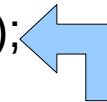
```
<?php
```

```
$handle = curl_init();
```

```
$url = "...url qui...";
```

```
// Set the url
```

```
curl_setopt($handle, CURLOPT_URL, $url);
```



```
// Set the result buffer
```

```
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
```

```
$output = curl_exec($handle);
```

```
curl_close($handle);
```

```
echo $output;
```

```
?>
```

Specifica le **opzioni della sessione curl**

CURLOPT\_URL dice che questa sessione è relativa a una connessione verso un dominio web, specificato nel terzo parametro

In alternativa si può passare il dominio web come parametro della funzione `curl_init($url)`

# Scaricare una pagina web

8

```
<?php
```

```
$handle = curl_init();
```

```
$url = "...url qui...";
```

```
// Set the url
```

```
curl_setopt($handle, CURLOPT_URL, $url);
```

```
// Set the result output to be a string
```

```
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
```

```
$output = curl_exec($handle);
```

```
curl_close($handle);
```

```
echo $output;
```

```
?>
```



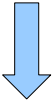
Se il terzo parametro è TRUE (1) il risultato della chiamata a curl\_exec() viene trasferito come stringa



# Scaricare una pagina web

9

```
<?php
    $handle = curl_init();
    $url = "...url qui...";
    // Set the url
    curl_setopt($handle, CURLOPT_URL, $url);
    // Set the result output to be a string
    curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
    $output = curl_exec($handle);
    curl_close($handle);
    echo $output;
?>
```



Le opzioni possono essere scritte in forma compatta usando un array

# Scaricare una pagina web

10

```
<?php
```

```
$handle = curl_init();
```

```
$url = "...url qui...";
```

```
// Set the url
```

```
curl_setopt($handle, CURLOPT_URL, $url);
```

```
// Set the result output to be a string
```

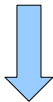
```
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
```

```
$output = curl_exec($handle);
```

```
curl_close($handle);
```

```
echo $output;
```

```
?>
```



**Esegue la sessione curl** con tutti i parametri che sono stati specificati

# Scaricare un file remoto

11

```
<?php
```

```
// Remote file
```

```
$url = "https://saw21.dibris.unige.it/images/webdev.jpg";
```

```
// Local file
```

```
$file = "img.jpg";
```

```
$handle = curl_init();
```

```
// Open the local file for writing
```

```
$fp = fopen($file, "w");
```

```
curl_setopt_array($handle, array(CURLOPT_URL=> $url,CURLOPT_FILE => $fp));
```

```
curl_exec($handle);
```

```
curl_close($handle);
```

```
fclose($fp);
```

```
?>
```

# Scaricare un file remoto

12

```
<?php

// Remote file
$url = "https://saw21.dibris.unige.it/images/webdev.jpg";

// Local file
$file = "img.jpg";

$handle = curl_init();

// Open the local file for writing
$fp = fopen($file, "w");

curl_setopt_array($handle, array(CURLOPT_URL=> $url,CURLOPT_FILE => $fp));

curl_exec($handle);
curl_close($handle);

fclose($fp);

?>
```



File nel quale scrivere quanto ritornato  
dalla chiamata

# Scaricare un file remoto

13

```
<?php
```

```
// Remote file
```

```
$url = "https://www.rfc-archive.org/getrfc?rfc=1149";
```

```
// Local file
```

```
$file = "rfc1149.html";
```

```
$handle = curl_init();
```

```
// Open the local file for writing
```

```
$fp = fopen($file, "w");
```

```
curl_setopt_array($handle, array(CURLOPT_URL=> $url,CURLOPT_FILE => $fp,
```

```
CURLOPT_HEADER => true));
```

```
curl_exec($handle);
```

```
curl_close($handle);
```

```
fclose($fp);
```

```
?>
```



Permette di gestire gli **header della risposta** per debugging

# Per gestire gli errori

14

```
$responseAll = curl_getinfo($handle);
```

```
$responseCode  = curl_getinfo($handle, CURLINFO_HTTP_CODE);  
$responseType  = curl_getinfo($handle, CURLINFO_CONTENT_TYPE);
```

```
if (curl_errno($handle)) {  
    echo("Error: " . curl_error($handle));  
    oppure  
    error_log("Error: " . curl_error($handle));  
}
```

# Inviare dati in POST via cURL

15

```
<?php
```

```
$url = "https://localhost/saw/registration.php";  
$handle = curl_init($url);
```

```
// Array with form fields names and values
```

```
$postData = array( 'firstname' => 'Marina',  
                  'lastname'  => 'Ribaud',  
                  'email'     => 'marina.ribaud@unige.it',  
                  'pass'      => 'secret123',  
                  'confirm'   => 'secret123',  
                  'submit'    => 'submit');
```

```
curl_setopt_array($handle, array( CURLOPT_POST => true,  
                                  CURLOPT_POSTFIELDS => $postData,  
                                  CURLOPT_RETURNTRANSFER => true ));
```

```
$data = curl_exec($handle);  
curl_close($handle);  
echo $data;  
?>
```

# Inviare dati in POST via cURL

16

```
<?php
```

```
$url = "https://localhost/saw/registration.php";  
$handle = curl_init($url);
```

```
// Array with form fields names and values
```

```
$postData = array( 'firstname' => 'Marina',  
                  'lastname' => 'Ribaud',  
                  'email'   => 'marina.ribaud@unige.it',  
                  'pass'=>'secret123',  
                  'confirm'=>'secret123',  
                  'submit'=>'submit');
```



Valori da inviare

```
curl_setopt_array($handle, array(  CURLOPT_POST => true,  
                                   CURLOPT_POSTFIELDS => $postData,  
                                   CURLOPT_RETURNTRANSFER => true ));
```

```
$data = curl_exec($handle);  
curl_close($handle);  
echo $data;  
?>
```



# Inviare dati in POST via cURL

17


```
<?php
```

```
$url = "https://localhost/saw/registration.php";  
$handle = curl_init($url);
```

```
// Array with form fields names and values  
$postData = array( 'firstname' => 'Marina',  
                  'lastname' => 'Ribaud',  
                  'email' => 'marina.ribaud@unige.it',  
                  'pass'=>'secret123',  
                  'confirm'=>'secret123',  
                  'submit'=>'submit');
```

```
curl_setopt_array($handle, array( CURLOPT_POST => true,  
                                  CURLOPT_POSTFIELDS => $postData,  
                                  CURLOPT_RETURNTRANSFER => true ));
```

```
$data = curl_exec($handle);  
curl_close($handle);  
echo $data;  
?>
```



Permette di inviare i dati in POST allo script  
specificato in \$url

# Gestire i cookie con cURL

18

```
<?php
$handle = curl_init();

$url = "....";
$file = "cookie.txt";
curl_setopt_array($handle,
    array(CURLOPT_URL => $url,
          CURLOPT_COOKIEFILE => $file,
          CURLOPT_COOKIEJAR => $file,
          CURLOPT_RETURNTRANSFER => true,));

$data = curl_exec($handle);

curl_close($handle);

?>
```

# Gestire i cookie con cURL

19

```
<?php
```

```
$handle = curl_init();
```

```
$url = "....";
```

```
$file = "cookie.txt";
```

```
curl_setopt_array($handle,
```

```
array(CURLOPT_URL => $url,
```

```
    CURLOPT_COOKIEFILE => $file,
```

```
    CURLOPT_COOKIEJAR => $file,
```

```
    CURLOPT_RETURNTRANSFER =>
```

Il file dal quale il cookie viene letto  
Il file nel quale il cookie viene scritto

```
$data = curl_exec($handle);
```

```
curl_close($handle);
```

```
?>
```

# Testing del progetto di SAW

20

A hand-drawn diagram illustrating a sign-up form and a test assertion. The form is enclosed in a green rectangular border. Inside the form, there are two input fields: the first is labeled 'Username:' and contains the text 'sammy123'; the second is labeled 'Password:' and contains eight asterisks '\*\*\*\*\*'. Below the input fields is a green button labeled 'Sign up >'. To the right of the form, there is a purple bracket that spans the height of the form. To the left of the bottom of this bracket, there is a green checkmark followed by the text 'User sammy123 exists in the database (Action: account creation)'.

Username:  
sammy123

Password:  
\*\*\*\*\*

Sign up >

✓ User sammy123 exists  
in the database  
(Action: account creation)

End to End  
Test:

Can a user  
accomplish an  
action?