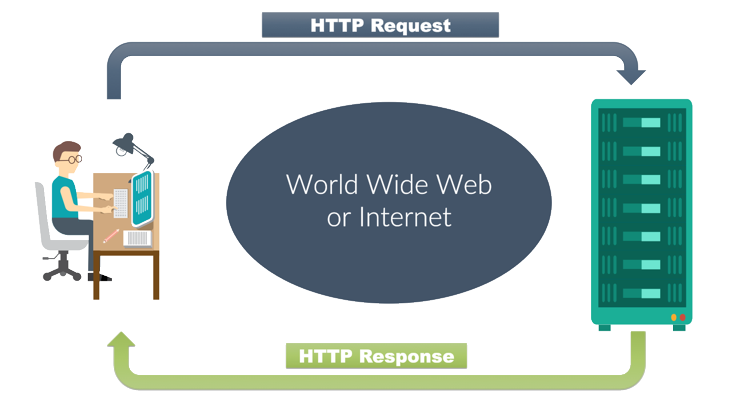
# Post di Matteo Licciardino

## Protocollo HTTP e response code

In questo **post** parlerò:

* Che cos’è un **HTTP**
* Che cos’è un **response code**
* **HTTP** nel sistema crittografico
* I vari di tipi di **response code**
* Cosa indica per ogni **response code**



### Che cos’è un HTTP?



**HTTP** è un protocollo pensato dall’ente **IETF** (**Internet Engineering Task Force**) con una **architettura client-server** questo si dimostrò molto funzionale perché con gli scambi di informazioni sul web tramite i link si stabiliscono i **collegamenti di natura stateless** si intende che le connessioni tra client e server vengono chiuse perché la singola richiesta è stata soddisfatta.

Con ciò si garantisce un **minor impiego di risorse**, perché si ha il **minore carico** e **minor occupazione** sia dal lato client e sia dal lato server, però c’è l’impossibilità di mantenere lo stato dell’utente con lo scambio di una sessione e l’altra con quale per ovviare questo problema si utilizzano per esempio i **cookie**.

La prima versione di questo protocollo è stata fatta nel 1996 con il **HTTP/0.9** dove c’erano delle imperfezioni.

Ci sono versioni migliorate rispetto alla prima versione per esempio c’è **HTTP/1.1** del 1999 e poi nel 2015 c’è la versione **HTTP 2** ancora in uso.

**HTTP** funziona quando noi digitiamo sul browser una pagina di un sito che dà le istruzioni alle due macchine che devono utilizzare lo stesso protocollo di rete, quindi il **browser** manda al server un **messaggio di richiesta** e quest’ultimo risponde con un **messaggio di risposta**.

Il **messaggio di richiesta** dal lato client contiene:

* **request line** che indica le informazioni con l’utilizzo del metodo HTTP,
* **header** che specifica quale URL da raggiungere,
* **riga vuota** con **due caratteri** che si chiamano **carriage return** e **line feed** che viene indicato con (**CRLF**),
* il corpo del messaggio cioè il **body**.

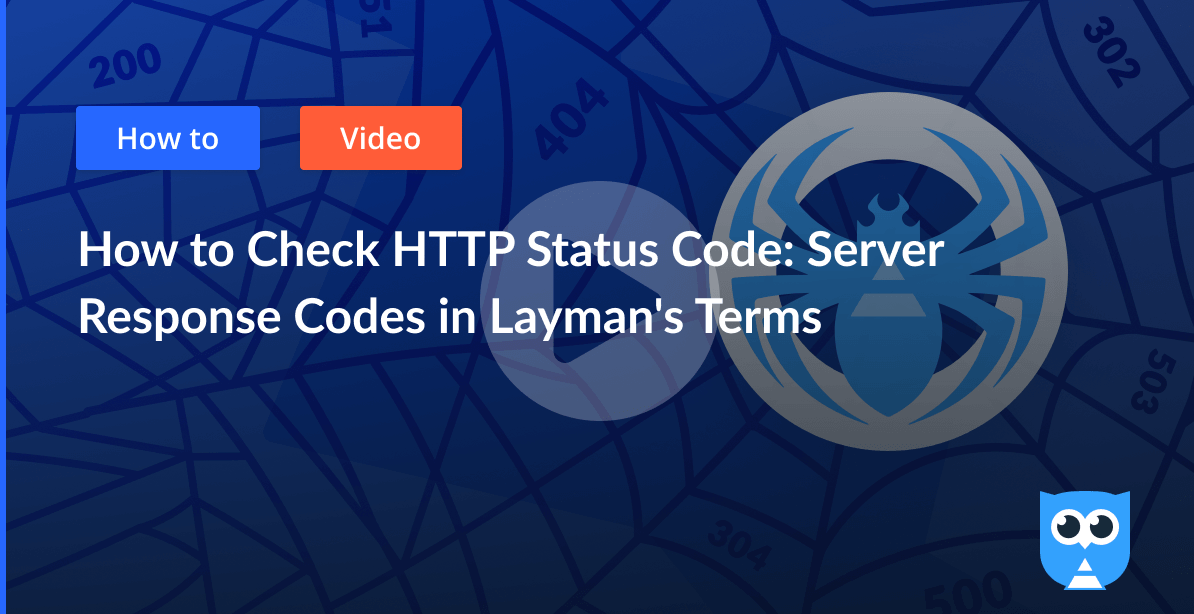
Con l’utilizzo del metodo **HTTP** dove contiene delle istruzioni per stabilire le due risorse in rete e sono:

* **GET** che indica il contenuto della risorsa in questione,
* **HEAD** che legge solo i contenuti nell’header,
* **POST** viene utilizzato per segnalare l’invio delle informazioni che parte dal client e arrivano al server.

Il messaggio di risposta dal lato server sono:

* **riga di stato**,
* **header**,
* **riga vuota** con **CRLF**,
* **body**.

### Che cos’è un response code?



**Response code** è un indicatore che indica se una specifica richiesta da parte del HTTP è stata completata correttamente e quest’ultime sono integrante nel protocollo HTTP, si intende quando un browser fa una richiesta ad una risorsa di un server e lui manda una risposta con uno status code formato da **tre cifre** e queste risposte dalla parte del server sono composte di due parti che sono **l’header** e il **body payload**.

* L’**header** contiene il response code del HTTP e altre informazioni, inoltre non sono mostrati all’utente, ma all’interno ci sono delle istruzionindove il client sa come gestire la risposta,
* il **body payload** contiene la risposta del HTTP e viene usato per dare una risposta grafica usando il termine di rendering e poi mostrate la pagina all’utente.

### HTTP nel sistema crittografico



Nei tempi moderni gli sviluppatori hanno sviluppato delle **versioni più sicure dell’HTTP**:

* La prima versione è stata **SHTTP**, ma ormai non si usa più.

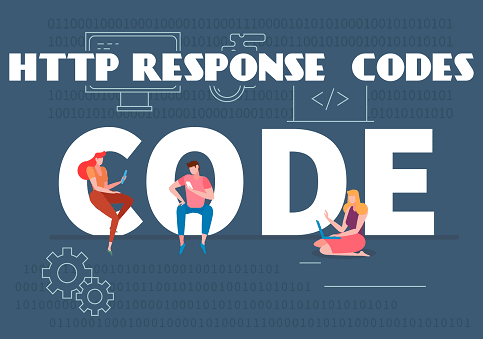
Ma si usa il **protocollo HTTPS** sarebbe l’unione del **protocollo HTTP** con l’aggiunta:

* dell’**algoritmo crittografico (TLS)**,
* del protocollo di crittografia **SSL**.

Questo algoritmo e con questo protocollo sono fondamentali perché offrono diversi vantaggi che sono:

* **autenticazione**: è un processo dove l’utente si identifica per partecipare ad una comunicazione,
* **segretezza**: è un fattore molto importante perché nessuno possa leggere il messaggio, ma tranne il destinatario,
* **integrità**: è la protezione che si usa su un messaggio dove non vengono autorizzate modifiche,
* **non ripudio**: è un meccanismo dove il mittente ha la certezza assoluta che il messaggio è stato inviato al destinatario.

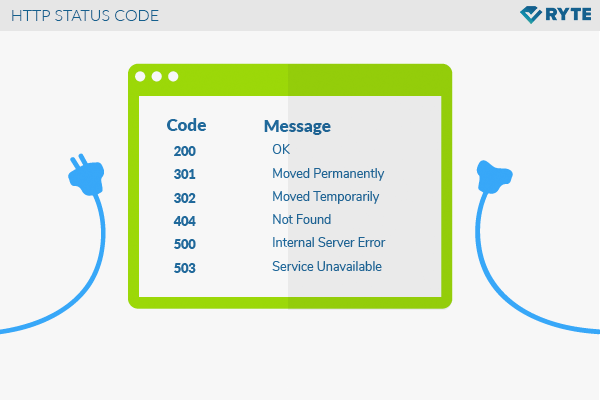
### I vari tipi di response code



I response code oppure detti anche status code ci sono di diversi tipi che però sono suddivisi in cinque categorie che ognuno di quest’ultime trasmettono un messaggio di diverso tipo che sono:

* **1xx** sono delle risposte informative che indica che hanno ricevuto la richiesta, ma stanno elaborando,
* **2xx** sono delle risposte di successo che indica che hanno ricevuto la richiesta con successo ed esce quello che hai richiesto,
* **3xx** sono delle risposte di reindirizzamento, cioè per completare la richiesta viene pretesa un’ulteriore azione per soddisfare,
* **4xx** sono delle risposte con gli errori del client, cioè che la richiesta fatta non può essere soddisfatta,
* **5xx** sono delle risposte con gli errori del server, cioè il server sta avendo dei problemi a completare la tua richiesta, ma allo stesso tempo la tua richiesta è valida.

### Cosa indica per ogni response code



* **1xx** è un tipo di status code dove indica che una richiesta è stata ricevuta e anche compresa, ma non è stata ancora processata, questo codice di stato di questa categoria è abbastanza raro da trovare.

Per esempio:

* il **101** oppure chiamato **101 switching protocols** che significa cambiare i protocolli indica il server che cambia il protocollo e questo viene indicato nell’intestazione richiesta da un client e questo si chiama **upgrade** è un meccanismo che si usa peraggiornare una connessione tra il client e il server che hanno già stabilitoun protocollo diverso.
* **2xx** è un tipo di status code dove indica che le richieste fatte dal client hanno avuto successo.

Per esempio:

* il **200 OK** quando il server manda questo indica che la richiesta del client è ricevuta con successo, inoltre fornisce il contenuto della richiesta fatta.

**ATTENZIONE! —>** se dal client viene richiesto una pagina che non esiste come risposta da parte del server manda la risposta response code 200 anziché il 404 è questo fenomeno si chiama:

* **errore soft 404** è un **URL** che restituisce una pagina che non esiste con un codice di stato 200 oppure delle pagine senza contenuti o vuota e queste vengono generate per vari motivi che sono:

1. mancanza di un file del server,
2. interruzione della connessione al database,
3. non carica nella pagina un file JS.

* **3xx** è un tipo di status code dove indica la richiesta fatta dal client è stata ricevuta dal server, però sul momento non può essere soddisfatta, perché il client deve inviare nuovamente una richiesta al server perché lui ha mandato come risposta con il **redirect**.

Per esempio:

* il **303** chiamato **303 see other** o **redirect 303**, questo codice di stato comunica al server che sta redirezionando l’URL richiesto con un altro URL, quest’ultimo si usa quando gli utenti sbagliano a cliccare due volte il form e nel frattempo cliccano il pulsante per tornare indietro del motore di ricerca.
* **4xx** è un tipo di status code dove indica che c’è stato un errore dalla parte del client, ci sono tre tipi di errori e sono:

1. Il **client** non è stato autorizzato ad accedere una pagina,
2. fatto delle **richieste** di pagine che non esistono,
3. fatto troppe **richieste**.

Per esempio:

* Il **404 not found** è un errore dove indica che la risorsa che è stata richiesta dal client non si trova, a volte possono restituire questo errore anche nelle pagine esistenti perchè:

1. a causa di un **bug**,
2. di problemi di manutenzione del **server**.

* **5xx** è un tipo di status code quando il client fa una richiesta al server, ma il server ha qualche problema a mandare la risposta perché quest’ultimo è troppo occupato oppure è temporaneamente non disponibile per causa della manutenzione.

Per esempio:

* il **504 Gateway Timeout** la definzione data dall’ente **IETF** dice che indica un server che sta svolgendo come funzione di **gateway** o di **proxy** che non ha ricevuto la risposta in tempo da un server **upstream** con il quale doveva completare la richiesta.