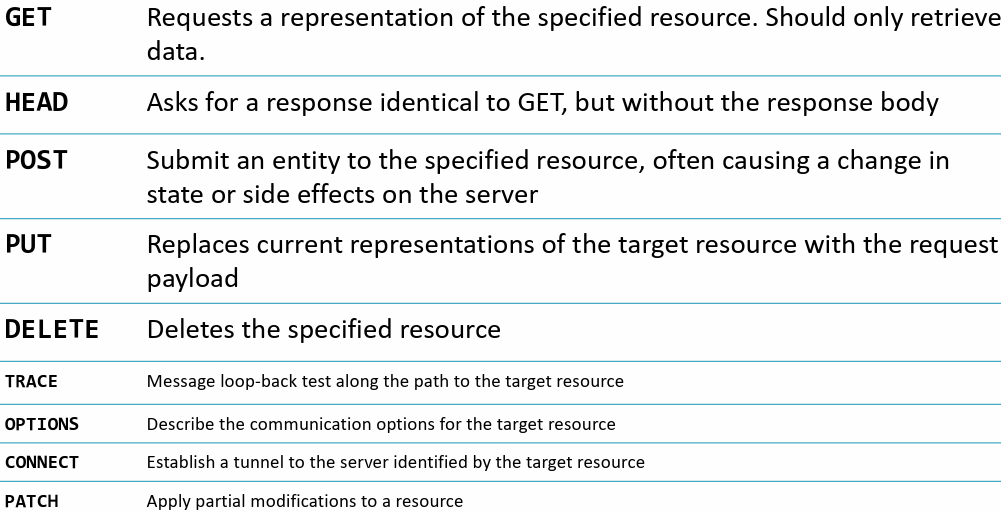
# EXPRESS

## http

**DOMANDA**

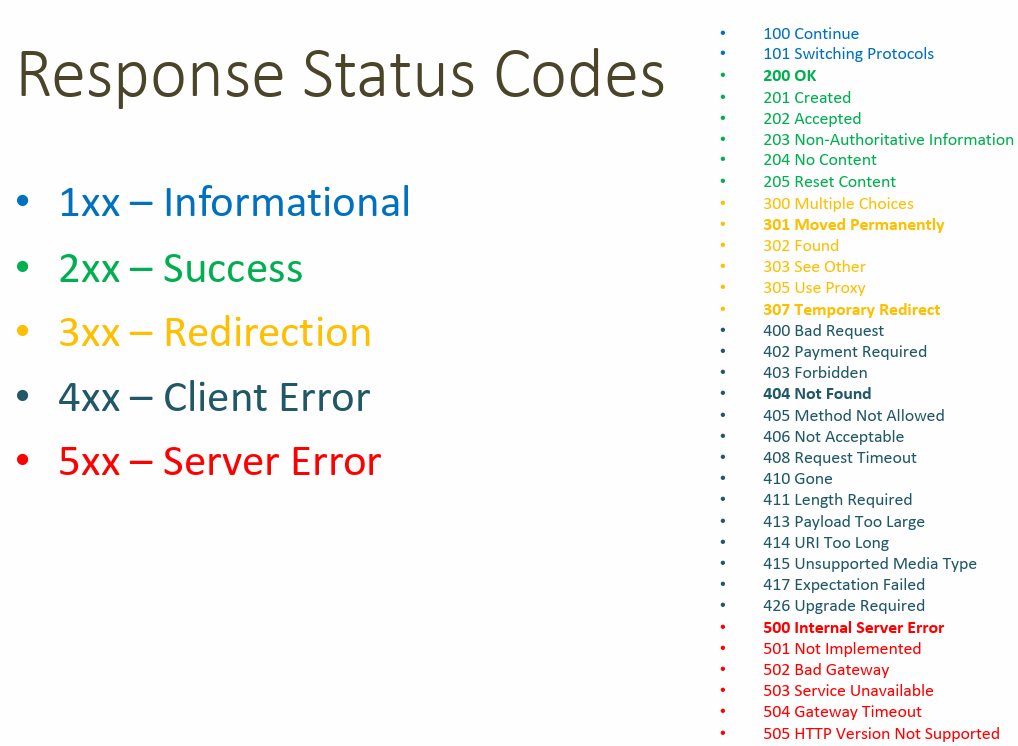
GET → manda richiesta per prendere dei dati

POST → form (es: login)

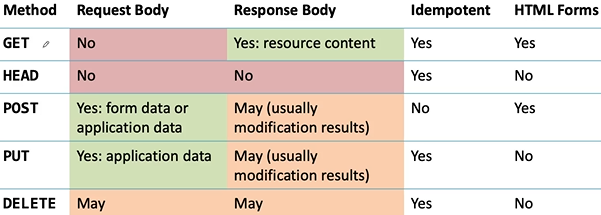
DELETE → manda richiesta per cancellare qualcosa

PUT → aggiorna dati già esistenti

PATCH → aggiornamento parziale di dati già esistenti

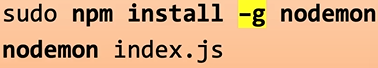


**RISPOSTA**

* Corpo
* Metodi
* Codice di risposta

## EXPRESS

Framework web in node.

* Una volta lanciato, Il server resta attivo tranne se applicazione crasha o utente fa ctrl C.
* Nel caso in cui si effettua una modifica, bisogna stoppare il server e poi restartarlo. (node ha un modulo che si accorge se un file è stato modificato e nel caso restarta il server)

Applicazione express formata da **3 aree**:

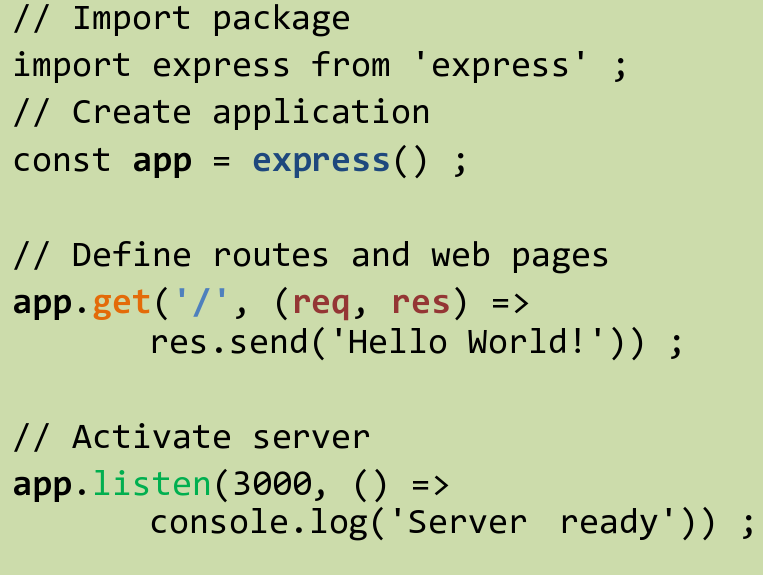
* Importa modulo e crea applicazione
  + Import express from ‘express’;
  + Const app =express();
* Configurazione di route, percorsi a cui il server deve fornire una risposta
  + App.get(‘url’, (req, res) => corpo callback)
  + App.method\_name(path, handler);
    - Metodi: get, post, put, delete, all, etc
      * all → prova a gestire qualsiasi tipo di richiesta
    - path → path a cui deve rispondere il server
* Attivazione server
  + App.listen(nr\_porta, callback)

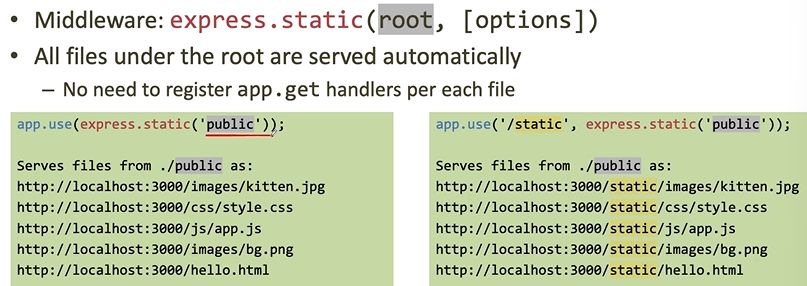
Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

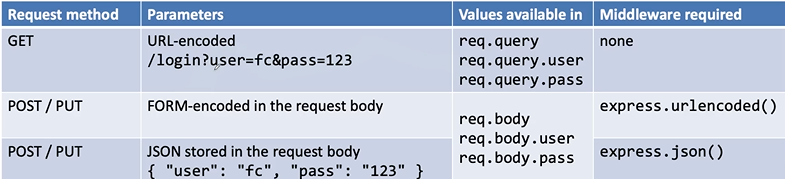


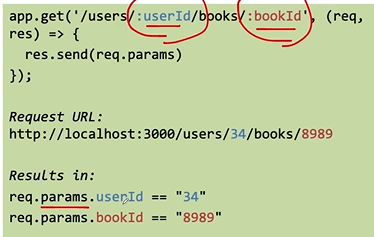
**Immagine che contiene testo, Carattere, bianco

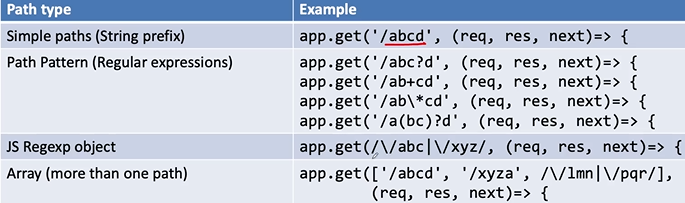
Descrizione generata automaticamenteMIDDLEWER**: funzioni che vengono chiamate ad ogni/certe richiesta che un server riceve.

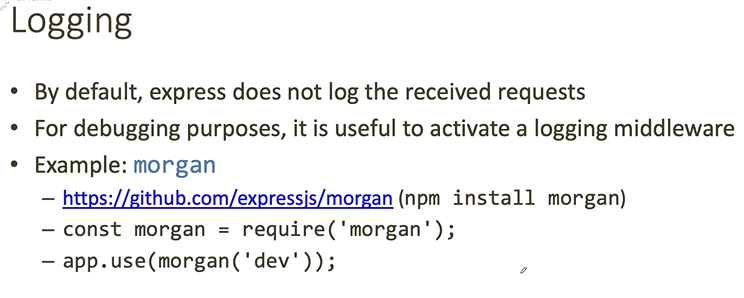
* app.use(MiddlwareCallback) → attivato per tutte le route
* app.use(path, MiddCallback) → attivato per path specific
* app.method(path, MiddCallback, (req,res) =>{}) → attivato per una specifica route su quel metodo

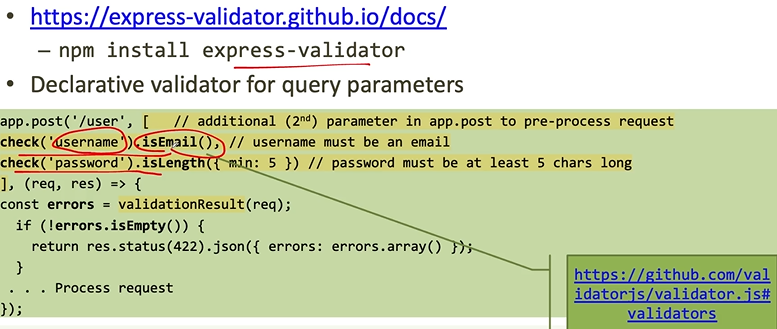
→ per oggetti statici ad esempio

**Richieste**: usando il middleware json, la stringa viene convertita in una stringa json con le relative proprietà.

**PATH**:

****



**VALIDATORE DI BODY:**