I 10 strumenti essenziali se lavori con Angular e il front-end

Parte 1: la guida definitiva a postman

Programmare è un processo creativo, ma spesso dobbiamo eseguire dei task noiosi e ripetitivi. Con questi 10 strumenti cercheremo di migliorare e velocizzare l'esperienza di sviluppo. Il primo strumento sarà Postman e vedremo tutte le funzioni utili per sfruttarlo nello sviluppo front-end.

Autore

Ciao, sono Salvatore e trovi tutto su di me sul sito https://linkedin.com/in/salvatoreromeo. Ho scritto questo mini libro principalmente perché mi piace scrivere e condividere esperienza.

Nella vita mi occupo di sviluppo software fin da quando avevo 12 anni. Oggi è il mio lavoro principale. Insegno Ingegneria del Software all'Università degli Studi di Perugia e un corso di 3 giorni su Angular per le aziende insieme al Codemotion. Il corso è da poco stato trasformato in un libro che puoi trovare su Amazon.

Se vuoi maggiori informazioni su di me contattami pure tramite linkedin o tramite il sito https://devexp.io. Lì potrai anche trovare articoli sul front-end, altri libri del gruppo devexp.io e i codici sorgenti pubblicati su Github.

Sommario

Introduzione

Oggi programmare il front-end è divertente e veloce. Ma se potessimo migliorare l'esperienza di sviluppo ed essere ancora più efficienti? Vedremo di seguito i migliori strumenti che utilizzo nello sviluppo di tutti i giorni e che vi assicuro possono arrivare a velocizzare lo sviluppo di un sistema in maniera importante.
Alcuni strumenti saranno forse noti, ma spesso non se ne conoscono tutte le potenzialità. Inoltre scommetto che altri strumenti non erano affatto noti. E sono sicuro di vincere la scommessa, perché sono stati sviluppati proprio da me :-)

Esigenza

Programmare è un processo creativo, ma spesso dobbiamo eseguire dei task noiosi e ripetitivi. E' nell'indole dello sviluppatore cercare di automatizzare più possibile questi task e fortunatamente la community in alcuni casi lo ha già fatto per noi. Se un task automatizzabile ci mette 10 secondi e lo eseguiamo 100 volte al giorno, si tratta di diversi minuti bruciati inutilmente. E' necessario automatizzarlo, o con uno script fatto da noi, oppure usando i tool sviluppati da altri, che spesso sono qualitativamente eccezionali.
Per i tool che vedremo di seguito ho selezionato i migliori nello sviluppo front-end, perché li uso più volte durante tutto il giorno. Inoltre io sono amante dei software portable, cioè quei software che non si installano e non vanno ad appesantire il sistema. I tool di seguito sono tutti portable.

I tool che andremo a conoscere

I tool che andremo a vedere in ogni sezione sono:

- **Postman** per lavorare con le richieste al server
- **AutoHotkey** per automatizzare qualsiasi task ripetitivo
- **Cmder** per ottimizzare l'uso della linea di comando
- **Http-server** per creare un server velocemente nella cartella corrente
- **Surge** per pubblicare la cartella corrente online con un comando
- **Ngrok** per rendere disponibile un server locale all'esterno
- **ColorPicker** per catturare i colori
- **Chrome** con i suoi strumenti base e avanzati
- **JSON Viewer** per visualizzare oggetti JSON

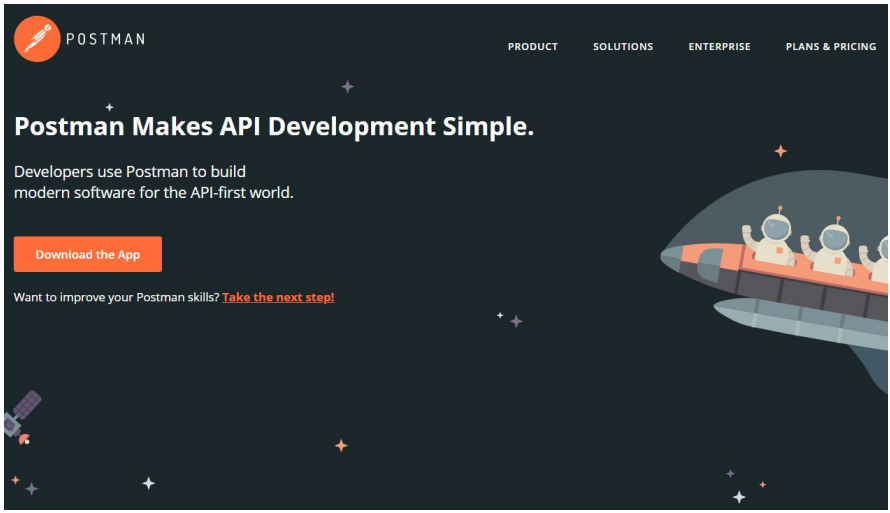
Vedremo un tool nel dettaglio in ogni "puntata". Cominciamo con Postman.

Postman

Il primo tool che uso praticamente ogni giorno è Postman.
Si tratta di uno strumento che permette di eseguire richieste HTTP ad un server di backend. Quando lavoriamo con un altro sviluppatore backend ci permette di condividere le API, ma la sua vera forza è quella di farci sapere tutto di una richiesta HTTP.
Possiamo creare la richiesta specificando tutti i parametri possibili, e conoscere ogni informazione della *request* e della *response*: *headers*, *body*, *risponse* ecc. Alcune di queste funzioni si possono trovare anche in Chrome, ma vedremo presto che Postman è molto più potente.

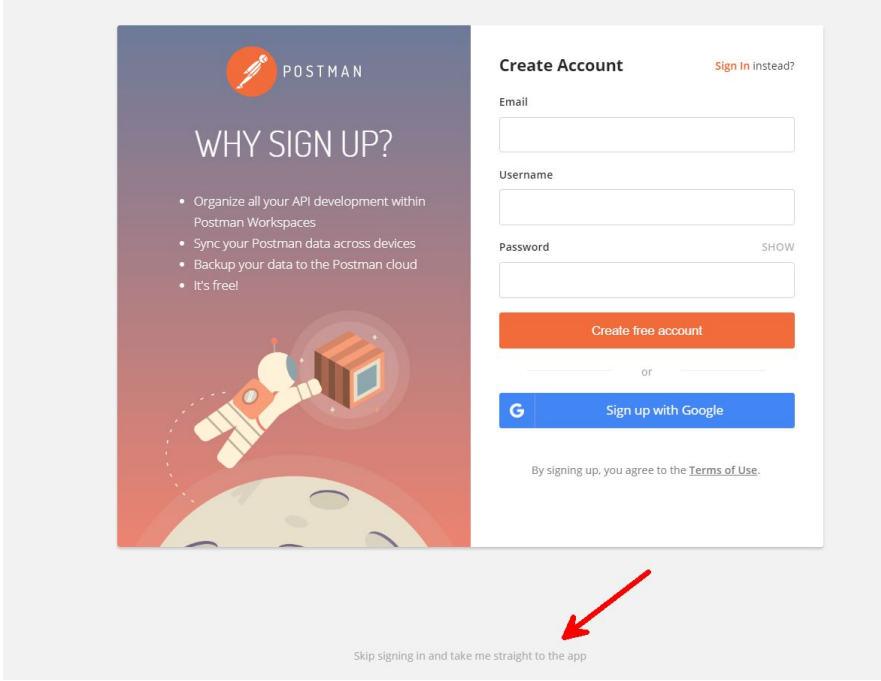
Installazione e primo avvio

Postman si può installare sia come estensione di Chrome, che come app standalone. Quest'ultima versione ha molte più funzionalità dell'estensione Chrome, quindi consiglio vivamente di usare questa.
Il sito per installarlo è https://www.getpostman.com/



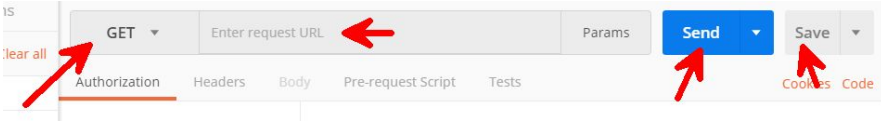
Ma esiste anche una versione portable a questo indirizzo: https://portapps.github.io/app/postman-portable/

Al primo avvio vedremo la schermata mostrata nella figura di seguito. Sembrerebbe che dobbiamo necessariamente registrarci, ma in basso, in grigio, è possibile saltare la fase di registrazione :-)



Postman: Funzioni principali

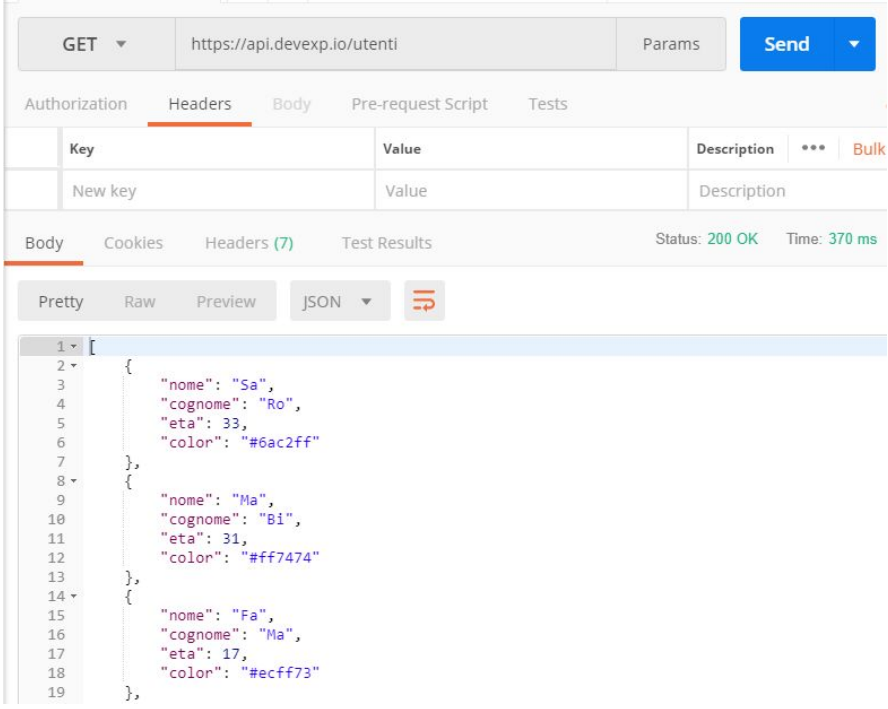
Una volta installato il sistema mostra questa schermata:



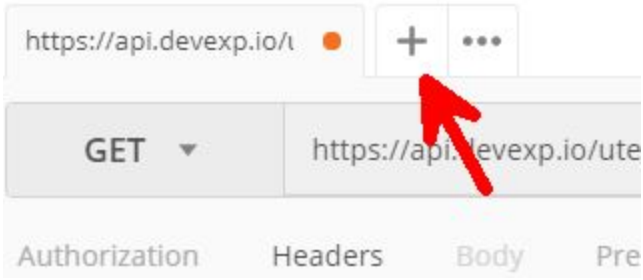
La barra al centro è la più importante. Possiamo:

1. specificare che tipo di richiesta effettuare (GET, POST, ...)
2. a quale indirizzo
3. inviare la richiesta con il pulsante SEND
4. Salvare la richiesta in una *collection*, una sorta di cartella in cui organizzare le nostre richieste, poi visibile nel menu a sinistra, come vedremo più avanti.

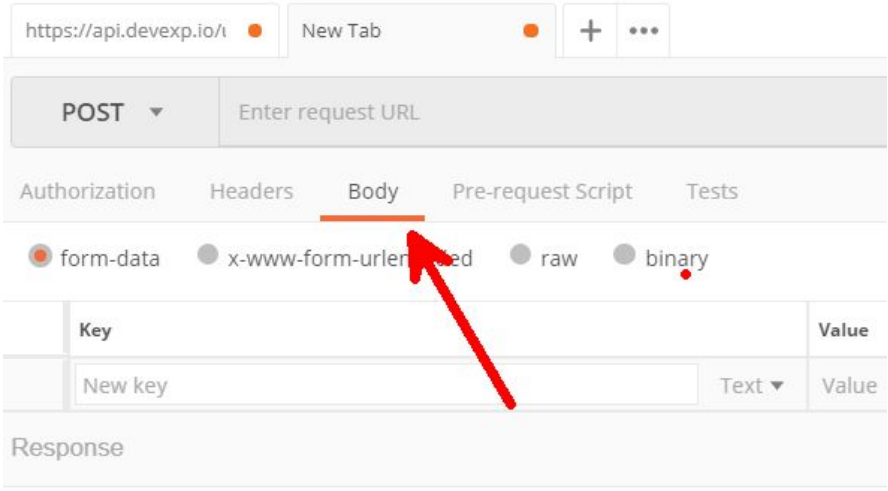
Questa è la richiesta al sito https://api.devexp.io/utenti che simula un'API per avere una lista di utenti. Il risultato è un array di oggetti JSON, come possiamo vedere in figura:



Per fare una nuova richiesta creiamo un nuovo tab col pulsante +

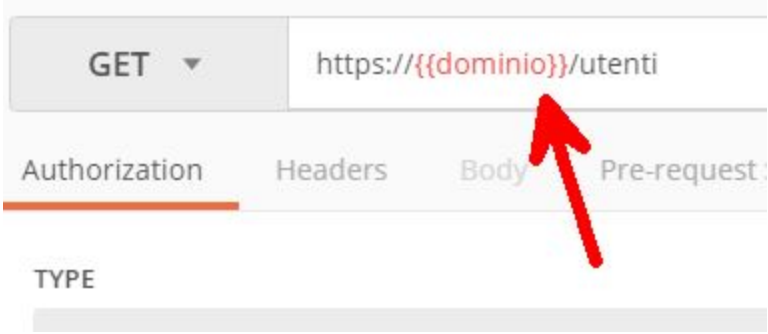


Proviamo a fare una richiesta POST. Nel caso di una POST possiamo specificare i parametri nella sezione sotto la barra:

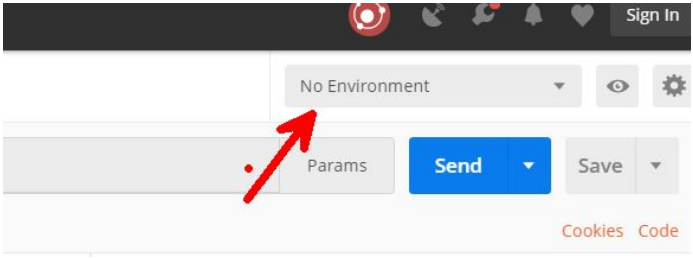


URL dinamici con gli environments

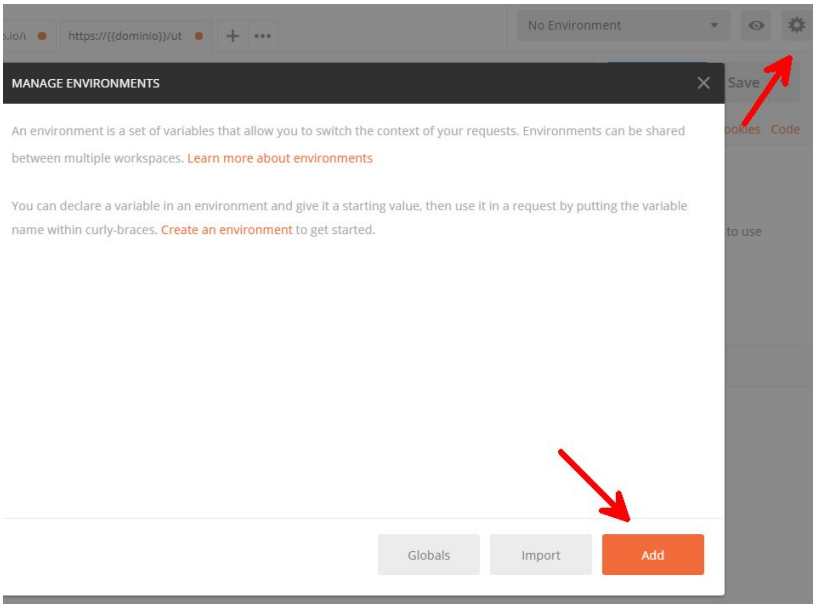
Come detto, Postman è uno strumento per analizzare e simulare le richieste ad un server. L'esigenza che avremo spesso durante lo sviluppo di un sistema è che a volte dobbiamo effettuare chiamate ad un server locale, ad esempio *localhost*, mentre altre volte la stessa chiamata va fatta ad un server esterno: si tratta delle stesse identiche chiamate, che differiscono soltanto per il dominio.
Possiamo rendere parte dell'URL dinamico con dei token usando gli *environment.*
Supponiamo quindi di voler rendere dinamico il dominio del nostro URL:



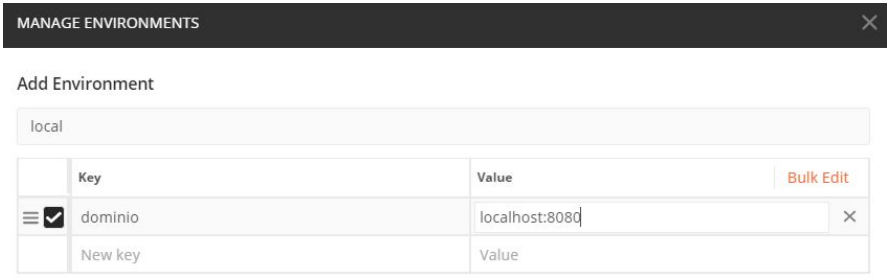
Ho qui sostituito il dominio *api.devexp.io* con il token *dominio*. Questo token è come una sorta di variabile. Ma dove definiamo il valore di questa variabile? Nella scheda environments:



Per creare un nuovo environment premiamo sulla rotella a destra

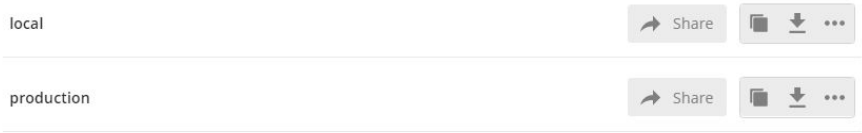


E poi creiamo il primo environment *local*, nel quale specifichiamo il valore della variabile *dominio*:

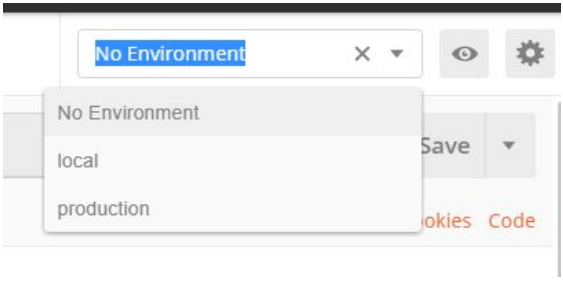


Creiamo quindi un secondo environment production nel quale *dominio* vale *api.devexp.io*:

Avremo quindi questi due *environment*:



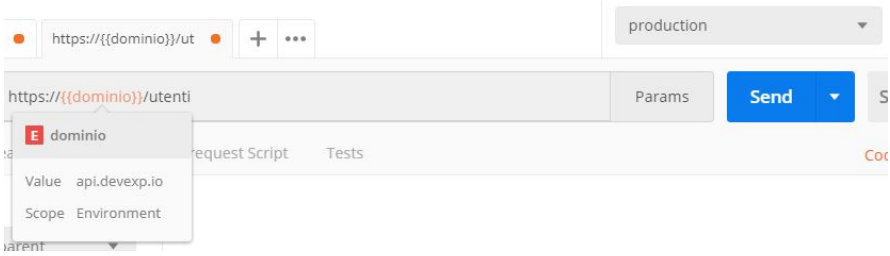
che sono ora disponibili nel menu dropdown in alto a destra:



Quando scegliamo *local*, *dominio* vale *localhost:8080*, mentre quando scegliamo *production*, *dominio* vale *api.devexp.io*.

Notate che se non scegliamo nessun environment, il token dominio nella richiesta è rosso, ad indicare che la variabile non ha nessun valore

mentre quando scegliamo ad esempio production il token *dominio* diventa arancione, ad indicare che la variabile ha un valore, *api.devexp.io* iun questo caso:



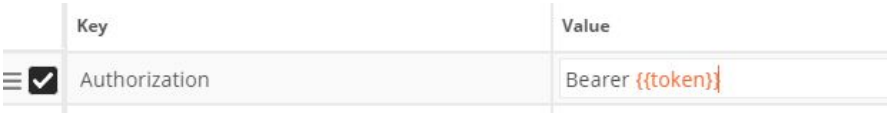
Possiamo specificare dei token in qualsiasi campo di configurazione della richiesta:

- URL (come abbiamo appena visto)
- headers
- body

Le API di postman

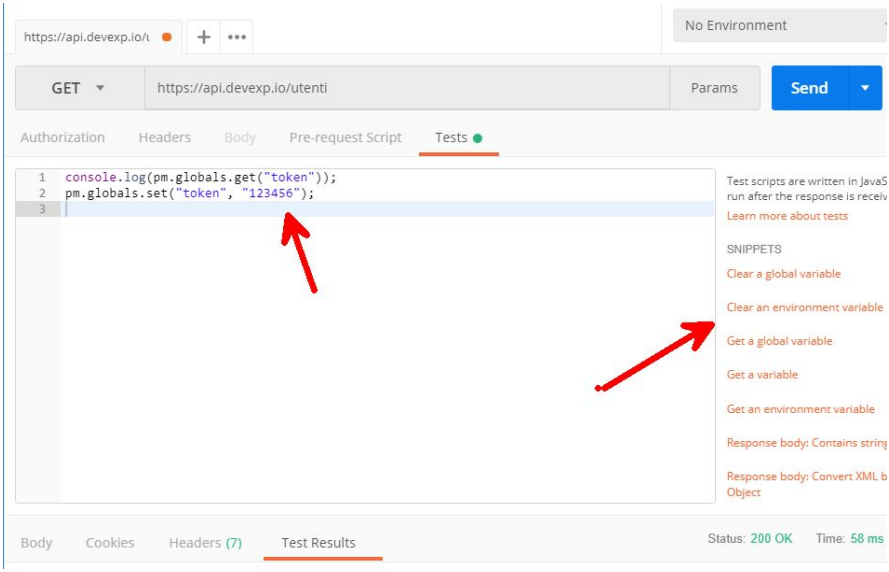
La cosa interessante è che postman dispone anche di API di scripting e possiamo ad esempio salvare un valore di una variabile usando le API. Con le API è possibile fare molte cose, tra cui i test di richieste al server.
Personalmente il caso d'uso più interessante è il seguente: supponiamo di avere un server le cui chiamate sono protette da autorizzazione con un token JWT che inviamo tramite l'header *Authorization*.

Un token JWT è una stringa, quindi dovremmo ogni volta cambiare la stringa con il valore del token JWT aggiornato. Quello che invece possiamo fare, è salvare tale stringa in una variabile come fatto per il dominio:

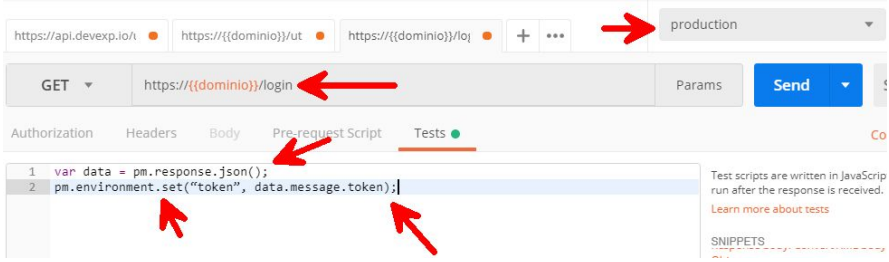


Tuttavia ora useremo le API per aggiornare il valore della variabile automaticamente quando facciamo la richiesta di login.

Lo faremo con uno script. Lo script va configurato nella sezione *Test* della richiesta:



A destra della sezione Test sono suggerite diverse snippet per casi d'uso molto frequenti. Combinando due di questi casi d'uso possiamo salvare il valore restituito dalla richiesta di login nel token del nostro *environment* attuale:

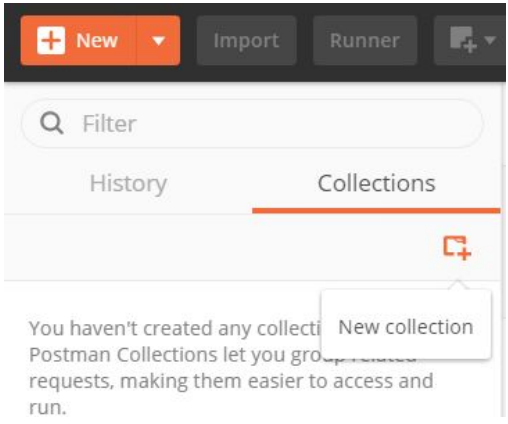


Se vogliamo debuggare lo script, possiamo vedere i risultati di espressioni stampate con il comando console.log(data) nella console.

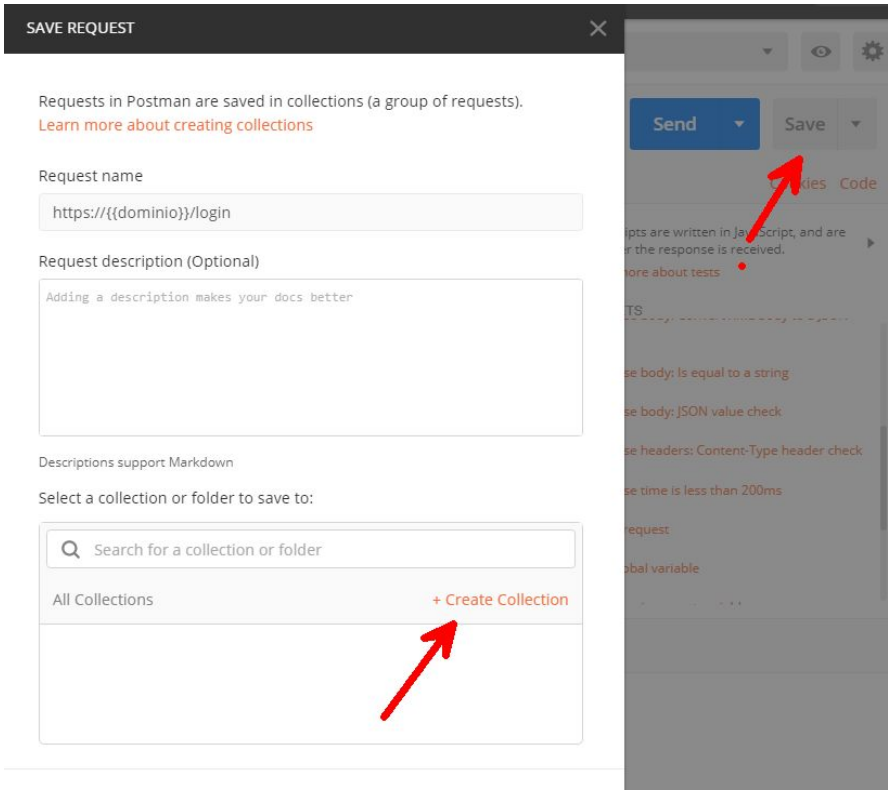
La console si avvia con l'hotkey CTRL-ALT-C:

Organizzare le richieste in collection

E' evidente che quando avremo tante richieste aperte cominceranno ad esserci troppi tab disordinati. Fortunatamente Postman mette a disposizione il concetto di *collection*. Una collection è una sorta di cartella dove possiamo salvare tutte le nostre richieste appartenenti ad uno stesso gruppo.
Per creare una *collection* possiamo usare il tab *Collections* e il pulsante subito sotto:

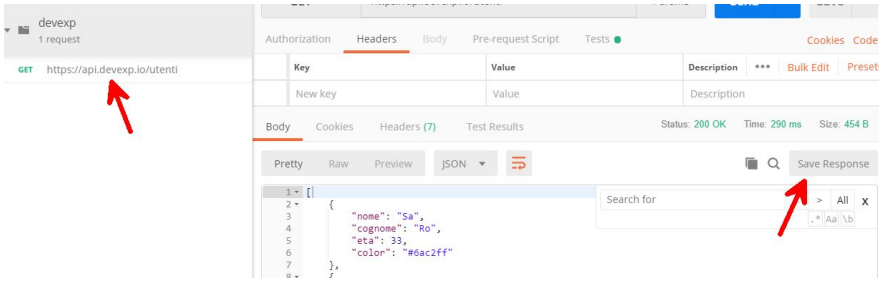


O in alternativa possiamo creare una nuova *collection* direttamente quando salviamo una richiesta:



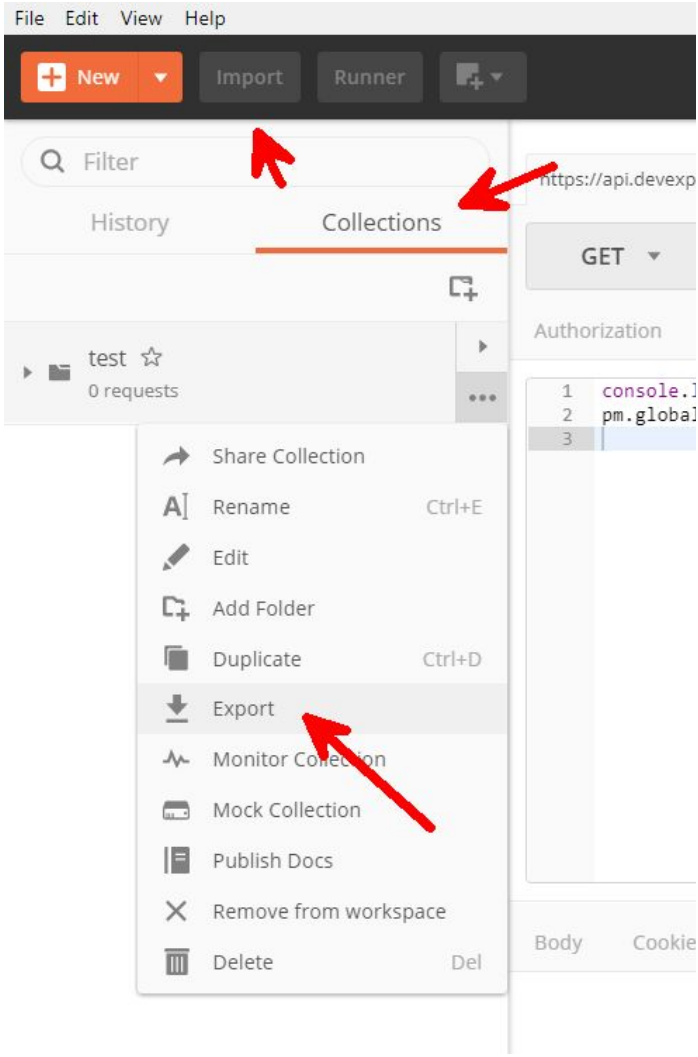
Sempre quando salviamo la richiesta è possibile naturalmente scegliere una *collection* già esistente dove salvarla.

Ma non è finita qui. Vi piacerebbe anche salvare le risposte del server di una particolare richiesta? Ebben nel momento in cui salviamo una richiesta in una collection, possiamo anche salvare tutte le risposte ricevute come riferimento per il futuro.
Basta usare il pulsante **Save Response** a destra subito sopra la risposta:


devexp
1 request
GET
https://api.devexp.io/utenti
Authorization
Headers
Body
Pre-request Script
Tests
Cookies
Code
Key
Value
Description
Bulk Edit
Presets
New key
Value
Description
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 290 ms
Size: 454 B
Pretty
Raw
Preview
JSON
Save Response
Search for
All
[
{
"nome": "Sa",
"cognome": "Ro",
"eta": 33,
"color": "#6ac2ff"
},

Condivisione delle collection

Le collection possono essere anche condivise con i nostri colleghi. Possiamo esportare e importare collection di richieste:



Il file ottenuto dopo l'export è un file json che possiamo condividere e che può essere importato tramite il pulsante

Import. Per completezza, è bene sapere che anche gli *environment* possono essere esportati e importati.

Conclusioni

Postman è un tool molto potente e con molte altre possibilità che scoprirai usandolo.
Nelle prossime puntate vedremo un altro tool potentissimo per automatizzare gran parte delle operazioni che ripetiamo più volte durante il processo di sviluppo: **AutoHotkey**.