SESSIONI E AUTENTICAZIONI

I cookies e le sessioni servono a colmare il vuoto della mancanza di stato di http.

Per ovviare a questo problema si usano le sessioni, dati interattivi salvati sul browser, per esempio gli faccio fare un login per la prima volta per stabilire la sessione, una volta fatto il login vengono salvati i dati della sessione e all’utente viene restituito un cookie per permettere il login.

SESSION ID

Serve per identificare un utente per gestirlo, ogni request http ha il proprio session id univoco, l’informazione del session id è salvata nel cliente ed è mandata da esso verso ogni richiesta per il server

COOKIE

È un piccolo file salvato all’interno del browser, è gestito autonomamente, non salvare informazioni sensibili.

Un cookie è dato da parametri come name, value, secure (si usa solo per siti https) e la data di scadenza.

AUTENTICAZIONE: ti identifico e verifico che tu sia che dici di essere

AUTORIZZAZIONE: il range di cose che puoi fare dopo esserti autenticato

È difficile come meccanismo sia da gestire efficientemente che da creare

Se è tutto giusto crea il session id e da allora usalo, d’ora in poi usa il session id

AUTENTICAZIONE CON FLASK

Va scelta una secret key che viene usata come seed per mandare l’encript di tutti i dati mandati da lì in poi

Si usa @login\_required per andare sul route solo se l’utente è identificato

Si usa @logout per permettere di fare il log out

Da current\_user posso capire che utente è correntemente loggato, va passata come parametro jinja