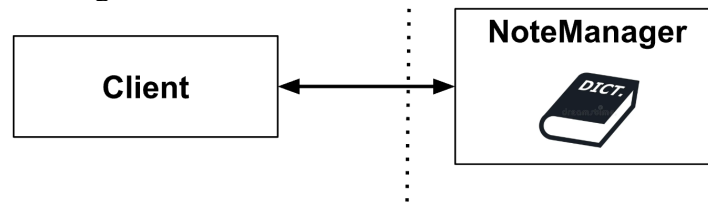


Università degli Studi di Napoli Federico II

Advanced Computer Programming

Il candidato realizzi un'applicazione Python basata su Flask per la gestione di note. Il sistema è composto da 2 entità, come illustrato in figura:



NoteManagement

- POST /note
- GET /note/<id>
- GET /notes
- PUT /note/<id>
- DELETE /note
- DELETE /notes

Una nota è descritta da un **text** (stringa) ed un **tag** (stringa), ed è in formato JSON, e.g., `{"text": "prova", "tag": "work"}`.

1. **NoteManager** è una web application basata su Flask che mette a disposizione una REST API con le seguenti Route HTTP:

- **POST /note:** Permette di aggiungere una nota presso il server. Il Manager riceve la nota (in formato JSON) attraverso il body della richiesta POST e genera un *id univoco* per quella nota. Tale id viene utilizzato per la memorizzazione della nota all'interno di un dizionario. Il dizionario avrà come chiave l'id generato, e come valore la nota ricevuta. Il Manager restituisce una risposta JSON contenente il risultato dell'operazione e l'id generato, e.g., `{"result": "added", "id": id}`.
- **GET /note/<id>:** permette di recuperare una nota dato il suo *id*. Il Manager riceve un *id* attraverso la URL della richiesta GET, che sfrutta per ricercare l'ordine all'interno del dizionario. Se la nota esiste, la risposta conterrà la nota, altrimenti essa conterrà un messaggio (in formato JSON) con l'esito dell'operazione, e.g., `{"result": "Note not found"}`. In quest'ultimo caso è necessario anche includere lo *status code 404* nella risposta.
- **GET /notes:** permette di recuperare tutte le note disponibili presso il Manager. Ricevuta la richiesta, il Manager recupera tutte le note presenti nel dizionario, e le restituisce sottoforma di lista di note, dove per ogni nota deve essere fornito anche il rispettivo id, e.g., `{"id": id, "note": note}`.
- **PUT /note/<id>:** permette l'aggiornamento di una nota a partire dal suo *id*. Il Manager riceve un *id* attraverso la URL della richiesta PUT e la nota da aggiornare nel body della richiesta. Se la nota esiste nel dizionario gestito dal Manager, la nota nel dizionario è sostituita con quella ricevuta nella richiesta. In caso contrario, il Manager crea una nuova nota. Il Manager ritornerà un messaggio con indicazione dell'esito dell'operazione (che può essere *added* o *updated*) e dell'id della nota, e.g., `{"result": msg, "id": id}`.
- **DELETE /note:** permette di eliminare una nota a partire dal suo *id*. Il Manager riceve l'id attraverso il parametro *id* incluso nella query string della richiesta. Se la nota è presente nel dizionario, il Manager rimuove la nota dal dizionario e ritorna un messaggio con l'indicazione dell'esito dell'operazione e con l'id della nota rimossa, e.g., `{"result":`

"deleted", "id": id}. In caso contrario, la risposta conterrà il solo esito dell'operazione, e.g., {"result": "Note not found"}, nonché lo *status code* 404.

- **DELETE /notes:** permette di cancellare tutte le note gestite dal Manager. Ricevuta la richiesta il Manager svuota il dizionario e ritorna l'esito dell'operazione, e.g., {"result": "No more notes"}.
2. **Client:** utilizza il Manager per la gestione delle note, e ne conserva una versione locale in un dizionario. Nel dettaglio, il Client effettua le seguenti operazioni:
- Genera 2 note, e ne richiede l'aggiunta al Manager attraverso la rotta *POST /note*. Le note sono salvate anche nel dizionario locale.
 - Chiede al Manager di fornire una nota partendo dall'*id* della stessa, attraverso la rotta *GET /note/<id>*. La nota ottenuta è utilizzata per aggiornare la versione locale della nota.
 - Chiede al Manager di fornire tutte le note da esso gestite, attraverso la rotta *GET /notes*. Le note ottenute sono utilizzate per aggiornare la versione locale delle note.
 - Genera una versione aggiornata di una nota, e chiede al Manager di aggiornarla, fornendo l'*id* e la nuova versione della nota, ed utilizzando la rotta *PUT /note/<id>*. La nuova versione della nota sarà utilizzata per aggiornare anche la versione locale.
 - Chiede al Manager la cancellazione di una nota partendo dall'*id* della stessa, attraverso la rotta *DELETE /note*. La nota è rimossa anche dal dizionario locale. DA notare che l'*id* deve essere incluso nella query string della richiesta.
 - Chiede al Manager la cancellazione di tutte le note gestite, attraverso la rotta *DELETE /notes*. Le note sono rimosse anche dal dizionario locale.