

# Attività Svolta: Implementazione di un Servizio REST con Jersey

In questa attività abbiamo implementato un servizio RESTful utilizzando la libreria Jersey.

Abbiamo imparato come configurare un progetto Maven per creare e consumare servizi REST, focalizzandoci sui seguenti passaggi:

1. **Creazione del progetto**: Il primo passo è stato la creazione di un progetto Maven che include le dipendenze necessarie per utilizzare Jersey, una libreria Java per lo sviluppo di applicazioni RESTful. Le dipendenze fondamentali sono:
  - `jersey-server` per creare il server REST
  - `jersey-client` per consumare i servizi REST
  - `jersey-media-json-jackson` per la serializzazione/deserializzazione in formato JSON.
2. **Implementazione del servizio REST**: Abbiamo creato una classe `BookService` annotata con `@Path` che espone un endpoint REST. Gli endpoint REST sono definiti utilizzando le annotazioni `@GET`, `@POST`, `@PUT`, e `@DELETE` di Jersey.
3. **Configurazione del server**: Utilizzando il plugin `jetty-maven-plugin`, abbiamo configurato il nostro progetto Maven per eseguire un server Jetty che ospita il nostro servizio REST.
4. **Consumo del servizio REST**: Successivamente, abbiamo creato un client Java che interagisce con il servizio REST appena creato. Il client utilizza le classi di Jersey per inviare richieste HTTP e ricevere risposte dal server.

In particolare, abbiamo implementato i metodi ``GET`` e ``POST`` per ottenere e aggiungere libri alla nostra libreria. L'esempio di client utilizza il framework Jersey per gestire la serializzazione JSON in modo trasparente.

5. **\*\*Test del client\*\***: Abbiamo infine configurato JUnit per testare il client e il server. Il metodo ``mvn test`` consente di verificare che il client effettui correttamente le richieste al server e che la risposta sia conforme alle aspettative.
6. **\*\*Avvio e testing\*\***: Durante l'esecuzione del progetto, abbiamo avviato il server con il comando ``mvn jetty:run`` e testato il comportamento del client tramite il comando ``mvn test``.

#### Conclusioni:

L'attività ci ha permesso di comprendere come configurare e utilizzare Jersey per creare e consumare servizi REST in modo semplice ed efficace. Inoltre, abbiamo imparato come eseguire test su client e server, assicurandoci che l'integrazione funzioni correttamente in un ambiente di sviluppo Java.