

PROTO FILE

SERVICES

service SRV

```
{  
    PRINTING BOOK  
    rpc get_book(limit)returns(dispbook){}  
  
    INSERTING BOOK  
    rpc ins_book(book)returns(ack){}  
  
    INSERTING REVIEW  
    rpc ins_review(review)returns(ack){}  
  
    PRINTING REVIEW  
    rpc get_review(review)returns(dispreview){}  
}
```

MESSAGE TYPES

inserting book in couchbase

```
message book{  
    int64 book_id=1;  
    string name=1;  
    repeated string author=2;  
    string short_desc=3;  
}
```

getting the limit

```
message limit{  
    int64 l=1;  
}
```

getting the books

```
message dispbook{  
    repeated book book_list=1;  
    ack ak= ;  
}
```

inserting the review

```
message review{  
    int64 book_id=1;  
    string name=2;  
    int64 score=3;  
    string text=4;  
}
```

getting the reviews

```
message dispreview{
    repeated review review_list=1;
    ack ak=2 ;
}
```

pass/fail

```
message ack{
    int64 status=1;
    string msg=2;
}
```

MICROSERVICES

MICROSERVICE 1

Mux handler which handles HTTP equests
Connects to grpc Server
forwards the http requests
receives data from grpc server

MICROSERVICE 2

Accepts grpc client
receives the http requests
forwards the requests to DB
receives data from DB
forwards the data to grpc client

CLIENT

Starts an HTTP server with a given address and GorillaMux handler

Creates a grpc channel to communicate with the server
Creates A Struct for parsing from json

```
type book struct{
    Name string`json:"name"`
    Author []string`json:"author"`
    Shortdesc string`json:"shortdesc"`
}
type review struct{
    Name string`json:"name"`
    Score int64`json:"score"`
    Text string`json:"text"`
}
```

```
    Id int64`json:"id"`  
}
```

Creates mux HandleFunc to respond to the HTTP request

insbook **POST** Sending data to server for book insertion
getbook **GET** Receiving data from server for getting books
insreview **POST** Sending data to server for review insertion
getreview **GET** Receiving data from server for getting review(s)

eg:

**For inserting a book used a function defined in handlefunc
sends the data by using POST method
sent it to server
waits for the acknowledgement i.e pass/fail**

SERVER

**Creates a grpc server and waits for connection
Defines the services which will communicate with couchbase sdk
for R/W operations and returns to the client**

Insbook **Send data to DB sdk for book insertion**
Insreview **Send data to DB sdk for review insertion**
Getbook **Receives data from DB sdk for getting books**
Getreview **Receives data fom DB sdk for getting reviews**

eg:

**For inserting a book
Define the service rpc insbk(addbook)returns(ack){}
Contains the data to be inserted in db
returns the acknowledgment integer to the client i.e pass/fail**

COUCBASE sdk

**Connects to DB
Creates Bucket
Creates Scope
Creates Collection
Creates primary key**

Creates A Struct for DB

```
type book struct{
    Name string`json:"name"`
    Author []string`json:"author"`
    Shortdesc string`json:"shortdesc"`
}
type review struct{
    Name string`json:"name"`
    Score int64`json:"score"`
    Text string`json:"text"`
    Id int64`json:"id"`
}
```

defining functions for communication b/w server & CB

ConnectDB	Connects to couchbase
Initializer	Creates Bucket,Scope,Collection
Addbook	Inserts book
Addreview	Inserts Review
Retbook	Creates primary key>Returns books
Retreview	Returns review(s)