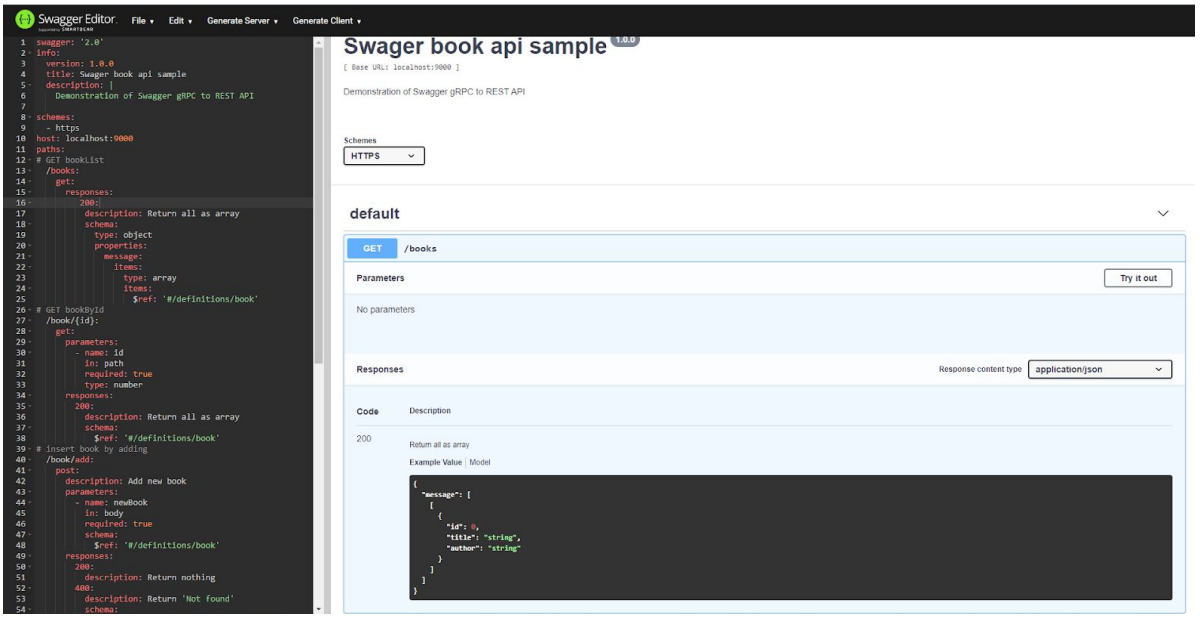
Soft Arch. Assignment ::1 four guys one cup

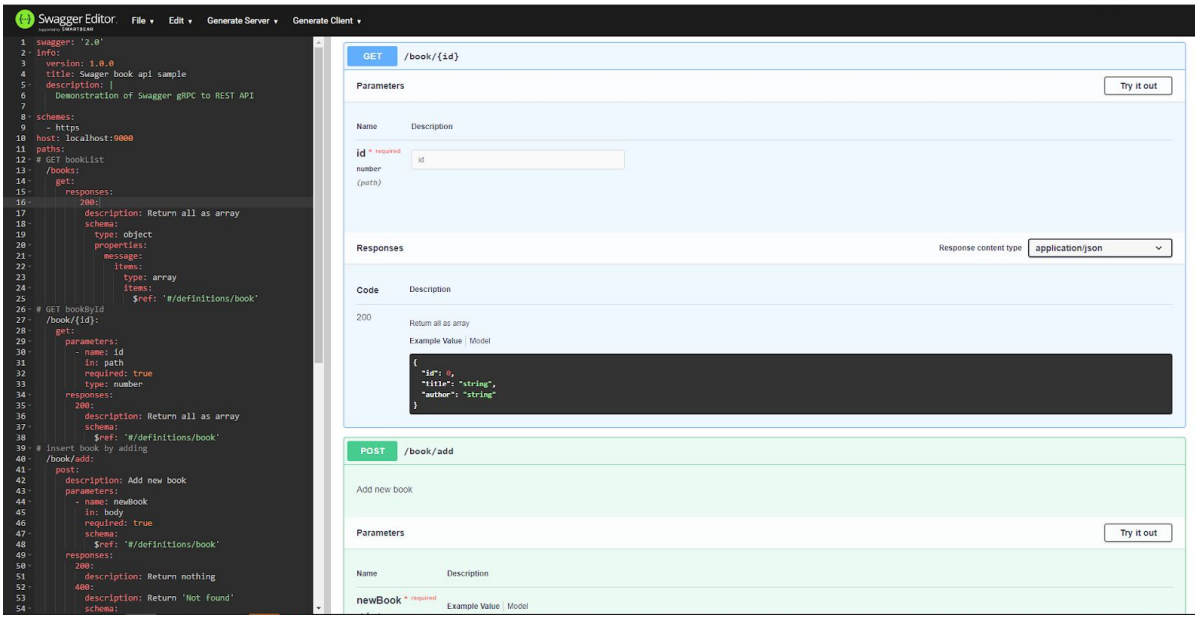
Team :four guys one cup

6031052421 วริทธิ์พล ศรีพรม
6030515521 วรรธนัย ทิพย์สถาพร
6030285121 ธาม ยองใย
6030200821 ณัฐนัย กิจวิวัฒนการ

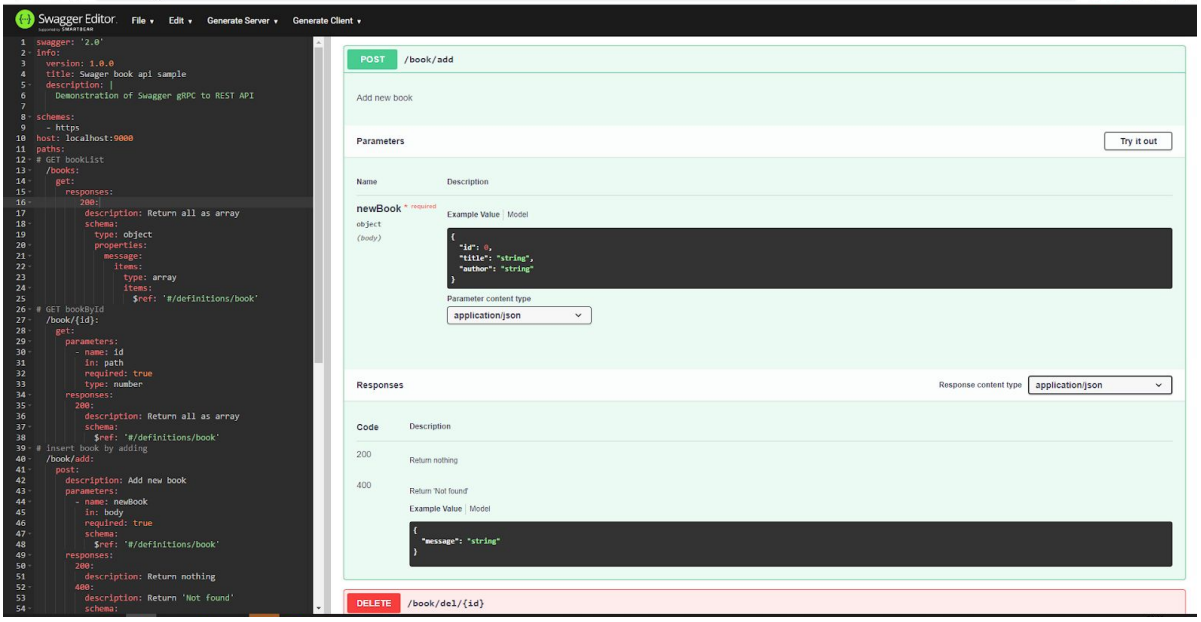
Things to be delivered:

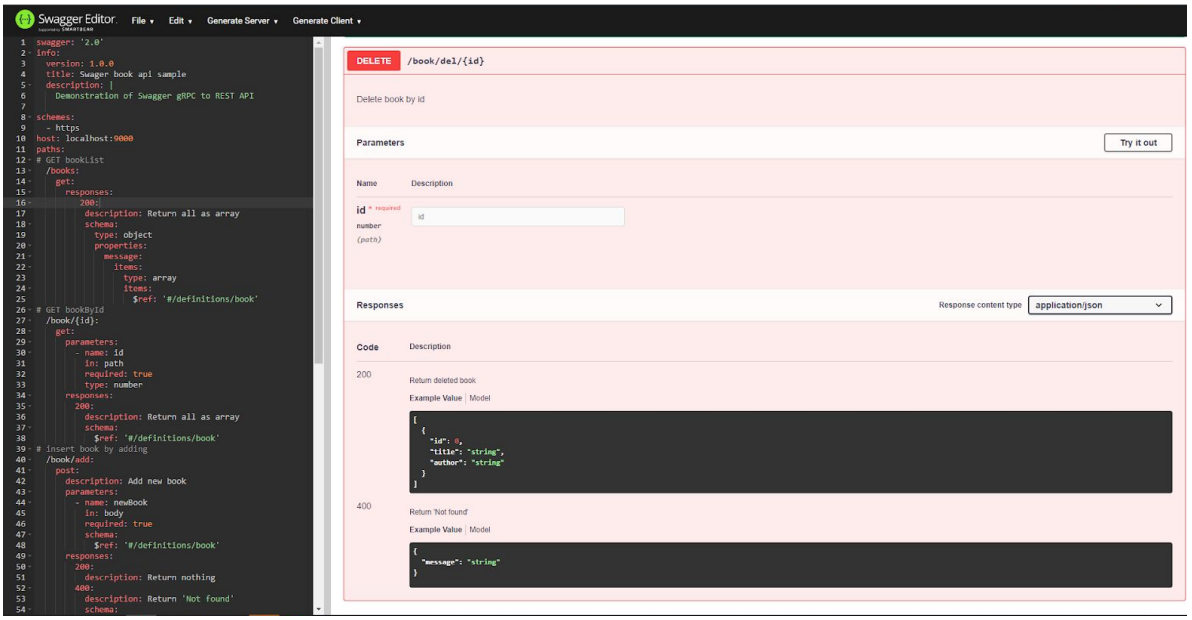
1. Screenshots of Swagger for your APIs in 2.





Soft Arch. Assignment ::1 four guys one cup





2. Source codes of 2 and 3.

```
const express = require('express');

const app = express();

var books = [
    {
        id: 123, title: 'A Tale of Two Cities', author: 'Charles Dickens'
    },
]


app.get('/books', (req, res)=>{
    return res.send(
        {
            "books": books
        }
    )
});

app.get('/book/:id', (req, res)=>{

    for (var i = 0; i < books.length; i++){
        if (books[i].id == req.params.id){
            return res.send(
                books[i]
            );
        }
    }
    return res.status(404).json({
        "message": "Not found"
    })
});

app.post('/book/add', (req, res)=>{

    let {_id, _title, _author} = req.body

    if (_id && _title && _author){
        books.push({
            id: _id,
            title: _title,
            author: _author
        })
        return res.status(200).json({})
    }
    else return res.sendStatus(404)
});

app.get('/book/del/:id', (req, res)=>{

    if (req.params.id){
        for (var i = 0; i < books.length; i++)
            if (books[i].id == req.params.id){
                books.splice(i, 1);
                return res.status(200).json({})
            }
        return res.status(404).json({
            "message": "Not found"
        })

    }
});


app.listen(50050, () =>
  console.log(`App listening on port ${50050}!`),
);
```

3. Compare how to call the methods based on gRPC and REST API side-by-side, e.g. in a Table format as shown below.

Function	gRPC	REST API
List books	execute client.go list command in command line to call service list in books.proto	GET http:domain/books
insert book	execute client.go insert with id in command line to call service Insert in books.proto	POST http:domain/book/add
Get book by id	execute client.go get with id in command line to call service Get in books.proto	GET http:domain/book/{id}
Delete book by id	execute client.go delete with id in command line to call service Delete in books.proto	GET http:domain/book/del/{id}

4. What are the main differences between REST API and gRPC?

gRPC is strongly typed messages. so, REST API is more flexible to send requests with JSON format.

5. What are the benefits of introducing interfaces in front of the gRPC and REST API of the book services.

Easy to migrate or change the backend functions while not interfering with the frontend calling to service

6. Based on the introduced interface, compare how to call the methods based on gRPC and REST API side-by-side, e.g. in a

Function	gRPC	REST API
List books	call from interface client.js to access function and message in books.proto	GET http:domain/books
insert book	call from interface client.js to access function and message in books.proto	POST http:domain/book/add
Get book by id	call from interface client.js to access function and message in books.proto	GET http:domain/book/{id}
Delete book by id	call from interface client.js to access function and message in books.proto	GET http:domain/book/del/{id}

7. Draw a component diagram representing the book services with and without interfaces.