Notes

- Request: data send from client to server and text in blue color in request JSON are required fields.
- Response: data send from server to client.
- There is no server side validation. So, kindly add all the validation on the front-end.

Author API

API URL	172.27.12.104:3000/author/new
REST Method	POST
Description	This service will add new author.

Request	{
	"**empid**": Number,
	"**name**": String,
	"**email**": String,
	"website": String,
	"department": String,
	"**skills**": Array
	}
	
	website is “null” by default, department is “Engineering” by default.

Response	{
	"message": "Author is saved."
	}

API URL	172.27.12.104:3000/author/update
REST Method	PUT
Description	This service will update existing author information.

Request	{
	"**empid**": Number,
	"name": String,
	"email": String,
	"website": String
	"department": String
	"skills": Array
	}
	
	User will not able to update empid, rest of the fields can be updated.

Response	{
	"message": "Author is updated."
	}

API URL	172.27.12.104:3000/author/remove
REST Method	DELETE
Description	This service will remove existing author information.

Request	{
	"**empid**": Number
	}

Response	{
	"message": "Author is deleted."
	}

API URL	172.27.12.104:3000/author/list
REST Method	GET
Description	This service will send the list of authors information
Request	No request body is required, just hit the services and you will get the data.

Response	[{
	"empid": 1,
	"name": "Ashwin Hegde",
	"email": "ashwinh@gmail.com",
	"website": "https://en.wikipedia.org/wiki/User:Hegdeashwin",
	"department": "Engineering",
	"skills": ["Node.js", "JavaScript", "Go", "Python", "PHP", "Backbone", "React"]
	}, {
	"empid": 5,
	"name": "Ajay Sajwan",
	"email": "ajays@gmail.com",
	"department": "Management",
	"skills": ["Team Management", "Communication"]
	}]

API URL	172.27.12.104:3000/author/byid
REST Method	POST
Description	This service will send the author information as per the empid matched

Request	{
	"**empid**": Number
	}

Response	{
	"empid": 1,
	"name": "Ashwin Hegde",
	"email": "ashwinh@gmail.com",
	"website": "https://en.wikipedia.org/wiki/User:Hegdeashwin",
	"department": "Engineering",
	"skills": ["Node.js", "JavaScript", "Go", "Python", "PHP", "Backbone", "React"]
	}

API URL	172.27.12.104:3000/author/byname
REST Method	POST
Description	This service will send the author information as per the name matched

Request	{
	"**name**": String
	}
	
	This is direct string match; wild card characters are not supported.

Response	{
	"empid": 1,
	"name": "Ashwin Hegde",
	"email": "ashwinh@gmail.com",
	"website": "https://en.wikipedia.org/wiki/User:Hegdeashwin",
	"department": "Engineering",
	"skills": ["Node.js", "JavaScript", "Go", "Python", "PHP", "Backbone", "React"]
	}

Book API

API URL	172.27.12.104:3000/book/new
REST Method	POST
Description	This service will add new book.

Request	{
	"**isbn**": String,
	"**title**": String,
	"**author**": String,
	"**price**": Number,
	"**availableOn**": Array
	}

Response	{
	"message": "Book is saved."
	}

API URL	172.27.12.104:3000/book/update
REST Method	PUT
Description	This service will update existing book information.

Request	{
	"**isbn**": String,
	"title": String,
	"author": String,
	"price": Number,
	"availableOn": Array
	}
	
	User will not able to update ISBN, rest of the fields can be updated.

Response	{
	"message": "Book is updated."
	}

API URL	172.27.12.104:3000/book/remove
REST Method	DELETE
Description	This service will remove existing book information.

Request	{
	"isbn": String
	}

Response	{
	"message": "Book is deleted."
	}

API URL	172.27.12.104:3000/book/list
REST Method	GET
Description	This service will send the list of book information
Request	No request body is required, just hit the services and you will get the data.

Response	[{
	"isbn": "ISBN-123-3456-789",
	"title": "Learning Node.js",
	"author": "Ashwin Hegde",
	"price": 350,
	"availableOn": ["Amazon", "eBay", "Flipkart"]
	}, {
	"isbn": "ISBN-987-8765-876",
	"title": "Learning Knockout",
	"author": "Jerin John",
	"price": 250,
	"availableOn": ["Amazon", "Flipkart"]
	}, {
	"isbn": "ISBN-356-8935-548",
	"title": "Learning Angular",
	"author": "Saju Sasidharan",
	"price": 650,
	"availableOn": ["Amazon", "eBay", "Flipkart"]
	}]

API URL	172.27.12.104:3000/book/byisbn
REST Method	POST
Description	This service will send the book information as per the ISBN matched
Request	{ "isbn": Number

	}

Response	{
	  "isbn": "ISBN-123-3456-789",
	  "title": "Learning Node.js",
	  "author": "Ashwin Hegde",
	  "price": 350,
	  "availableOn": ["Amazon", "eBay", "Flipkart"]
	}