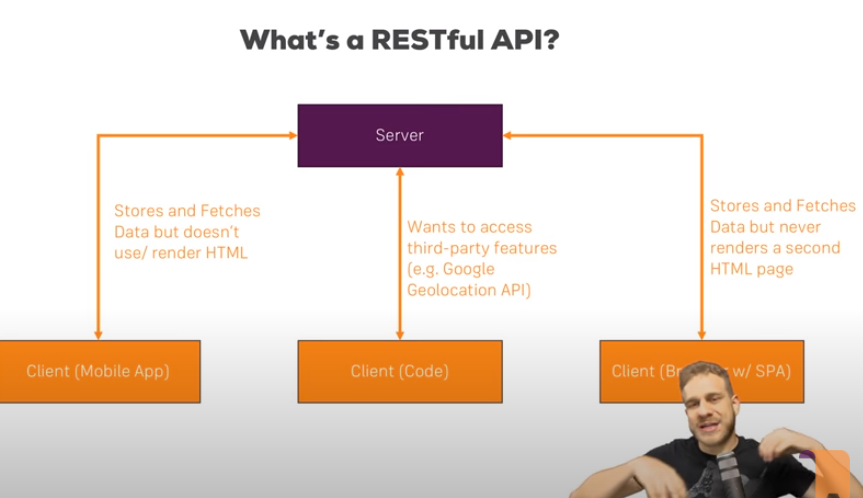
REST API

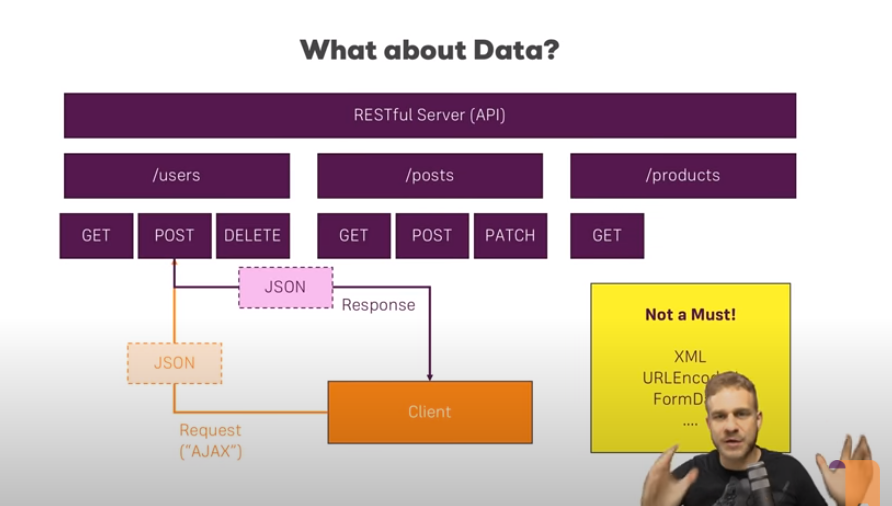
It stands for representational state transfer

Used to exchange data between client and server, like sending and fetching data to and from server. Data mostly exchange in JSON format, but not mandatory

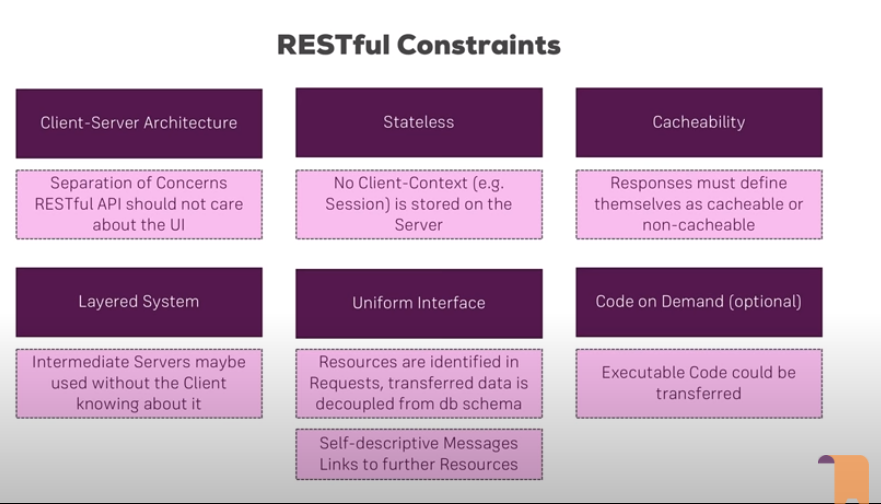
Does not render html pages

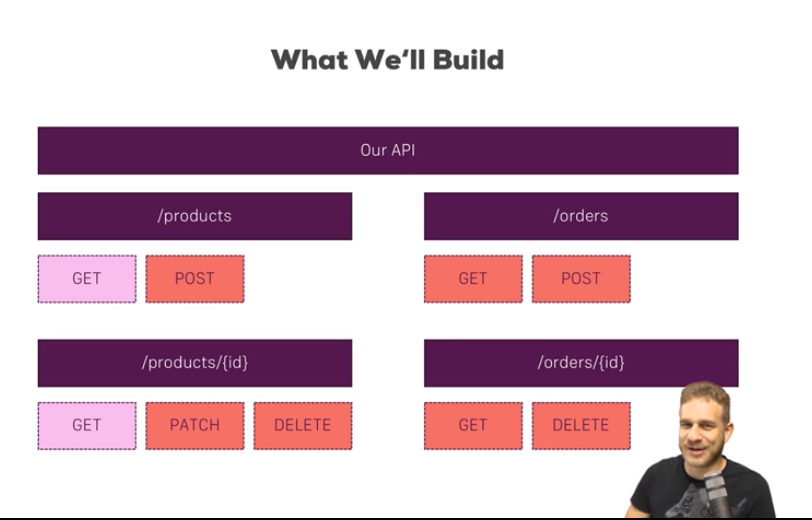


It is a stateless backend, does not care of individual client connected to it.



RESTful Constraints:





Dependencies to install:

Npm init

Npm I express

SIMPLE GET AND POST METHOD:

Server.js

const http = require('http');

const app = require('./app')

const port = process.env.PORT || 3000

const server = http.createServer(app);

server.listen(port);

app.js

const express = require('express')

const app = express();

const productRoutes = require('./api/routes/products')

//middleware

app.use('/products', productRoutes);

module.exports = app;

products.js

const express = require('express')

const router = express.Router(); //to create routes

router.get('/', (req, res, next)=>{

    res.status(200).json({

        message: "Handling GET requests to /products"

    })

})

router.post('/', (req, res, next)=>{

    res.status(201).json({

        message: "Handling POST requests to /products"

    })

})

//for particular product in products

router.get('/:productId', (req, res, next)=>{

    const id = req.params.productId;

    if(id === 'special'){

        res.status(200).json({

            message: 'You discover the special ID',

            id: id

        })

    }

    else {

        res.status(200).json({

            message: 'You passed an ID'

        })

    }

})

router.patch('/:productId', (req, res, next)=>{

    res.status(200).json({

        message: 'Updated product!'

    })

})

router.delete('/:productId', (req, res, next)=>{

    res.status(200).json({

        message: 'Deleted product'

    })

})

module.exports = router;

Till here, we have covered get, post, delete and update route(patch). This is how it works

Now adding another parent route which is for the orders, creating file orders.js which contains all the routes and which is called in app.js in app.use

Order.js

const express = require('express')

const router = express.Router(); //to create routes

router.get('/', (req, res, next)=>{

    res.status(200).json({

        message: "Orders were fetched"

    })

})

router.post('/', (req, res, next)=>{

    res.status(201).json({

        message: "Order was created"

    })

})

//for particular order in orders

router.get('/:orderId', (req, res, next)=>{

    res.status(200).json({

            message: 'Order details',

            orderId: req.params.orderId

        })

})

router.delete('/:orderId', (req, res, next)=>{

    res.status(200).json({

        message: 'Order deleted',

        orderId: req.params.orderIds

    })

})

module.exports = router;

calling in app.js

const express = require('express')

const app = express();

const productRoutes = require('./api/routes/products')

const orderRoutes = require('./api/routes/orders')

//middleware

app.use('/products', productRoutes);

app.use('/orders', orderRoutes);

module.exports = app;