Stripe queries:  
<https://stripe.com/docs/api/plans/update?lang=node>  
1- Create plan  
2-delte plan  
3-update plan  
4-get plan   
5-get all plans

Stripe update customer subscription plan:  
<https://stripe.com/docs/billing/subscriptions/upgrade-downgrade>

First log into strip account.  
Select develops and then API Keys here copy Secrect key and paste into node code

const stripe = require("stripe")("Secret Key");

and use this code into node project. Create token API will create token and this token will be used into Access\_Token API at source

const stripe = require("stripe")("sk\_test\_Z6Enxf5PgmQsWNQpOks1glIs00yXgmityu");

const express = require('express');

const router = express.Router();

//const models = require('../../Modal/Modal');

const Sequelize = require('sequelize');

const Op = Sequelize.Op;

router.get('/Access\_Token', (req, res, next) =>{

  stripe.charges.create(

    {

      amount:1223, //req.body.data.amount\*100,    //100 is equual to 1

      currency:'gbp',

      source: 'tok\_1H1H6VKfmbJj7r1EQjc9P0eJ',//req.body.data.token,  //token that is generated by card

      description: 'my description',//req.body.data.description,

    },

    function (err, charge) {

      // asynchronously called

      if (charge != null) {

        res.status(200)

          .json({

            data: charge,

            Message: 'Successful',

            Successful: true

          })

      } else {

        res.status(404)

          .json({

            data: err,

            Message: 'Failed',

            Successful: false

          })

      }

    }

  );

})

router.get('/Create\_Token', (req, res, next) => {

  stripe.tokens.create(

    {

      card: {

        number: '4242424242424242',//req.body.data.number,

        exp\_month: '12',//req.body.data.exmonth,

        exp\_year: '2020',//req.body.data.exyear,

        cvc: '123'//req.body.data.cvc,

      },

    },

    function (err, token) {

      if (token != null) {

        res.status(200)

          .json({

            Token: token,

            Message: 'Successful',

            Successful: true

          })

      } else {

        res.status(404)

          .json({

            data: err,

            Message: 'Failed',

            Successful: false

          })

      }

    }

  );

})

module.exports = router;

or learn more then click on learn more about api then click on APIs&SDK and then on API Refrence select NODE. And read document.  
if you are working in another language then select this language.  
  
  
  
**Some Code from All In Academy:**

const stripe = require('stripe')(process.env.STRIPE\_SECRET\_KEY);

const { StripeKeys } = require('../utils/constants');

const createPlan = async (data) => {

try {

// amount: 9700,

// currency: 'usd',

// interval: 'month',

// product: 'prod\_HgGm916gGzaBQQ',

data.product = StripeKeys.courses\_prductid;

const plan = await stripe.plans.create(data);

return plan;

} catch (err) {

throw new Error(err.message);

}

};

const getPlan = async (id) => {

try {

//price\_HMu1caXjJHp6yK

const plan = await stripe.plans.retrieve(id);

return plan

} catch (error) {

throw new Error(error.message);

}

};

const chargePayment = async (data) => {

const payment = await stripe.charges.create(data);

return payment;

}

const updatePlan = async (id, amount) => {

try {

//price\_HMu1caXjJHp6yK

const prevplan = await stripe.plans.retrieve(id);

await stripe.plans.del(id);

let data = {

id: prevplan.id,

amount: amount,

currency: prevplan.currency,

interval: prevplan.interval,

product: prevplan.product

}

const plan = await stripe.plans.create(data);

return plan;

} catch (err) {

throw new Error(err.message);

}

};

const deletePlan = async (id) => {

try {

//price\_HMu1caXjJHp6yK

const plan = await stripe.plans.del(id);

return plan

} catch (error) {

throw new Error(err.message);

}

};

const createCustomer = async (data) => {

try {

//email: body.email,

//source: body.source //customer token

const customer = await stripe.customers.create(data);

return customer;

} catch (err) {

throw new Error(err.message);

}

};

const subscription = async (data) => {

try {

return stripe.subscriptions.create({

customer: data.customer,

items: [{ price: data.price }]

});

} catch (err) {

throw new Error(err.message);

}

};

const get\_subscription = async (data) => {

try {

return stripe.subscriptions.retrieve(data.customer);

} catch (err) {

throw new Error(err.message);

}

};

const getPaymentMethods = async (data) => {

try {

const result = await stripe.paymentMethods.list({

customer: data.customerId,

type: 'card'

});

const customer = await stripe.customers.retrieve(data.customerId);

const pmWithDefaultCard = result.data.map((card) => {

if (card.id === customer.default\_source) {

card.isDefault = true;

} else {

card.isDefault = false;

}

return card;

});

return pmWithDefaultCard;

} catch (err) {

throw new Error(err.message);

}

};

module.exports = {

createPlan,

getPlan,

updatePlan,

deletePlan,

createCustomer,

subscription,

getPaymentMethods,

chargePayment,

get\_subscription

};