

- 初始化数据库

```
// 初始化数据库
const sequelize = require('./util/database');
const Product = require('./models/product');
const User = require('./models/user');
const Cart = require('./models/cart');
const CartItem = require('./models/cart-item');
const Order = require('./models/order');
const OrderItem = require('./models/order-item');
```

```
Product.belongsTo(User, {
  constraints: true,
  onDelete: 'CASCADE'
});
User.hasMany(Product);
User.hasOne(Cart);
Cart.belongsTo(User);
Cart.belongsToMany(Product, {
  through: CartItem
});
Product.belongsToMany(Cart, {
  through: CartItem
});
Order.belongsTo(User);
User.hasMany(Order);
Order.belongsToMany(Product, {
  through: OrderItem
});
Product.belongsToMany(Order, {
  through: OrderItem
});
```

- 同步数据

```
// 同步数据
sequelize.sync().then(
  async result => {
    let user = await User.findByPk(1)
    if (!user) {
      user = await User.create({
        name: 'Sourav',
        email: 'sourav.dey9@gmail.com'
      })
      await user.createCart();
    }
    app.listen(3000, () => console.log("Listening to port 3000"));
  })
```

- 中间件鉴权

```
app.use(async (ctx, next) => {  
  const user = await User.findByPk(1)  
  ctx.user = user;  
  await next();  
});
```

- 功能实现

```
const router = require('koa-router')()  
  
/**  
 * 查询产品  
 */  
router.get('/admin/products', async (ctx, next) => {  
  // const products = await ctx.user.getProducts()  
  const products = await Product.findAll()  
  ctx.body = { prods: products }  
})  
  
/**  
 * 创建产品  
 */  
router.post('/admin/product', async ctx => {  
  const body = ctx.request.body  
  const res = await ctx.user.createProduct(body)  
  ctx.body = { success: true }  
})  
  
/**  
 * 创建产品  
 */  
router.delete('/admin/product/:id', async (ctx, next) => {  
  const id = ctx.params.id  
  const res = await Product.destroy({  
    where: {  
      id  
    }  
  })  
  ctx.body = { success: true }  
})  
  
/**  
 * 查询购物车  
 */  
router.get('/cart', async ctx => {  
  const cart = await ctx.user.getCart()  
  const products = await cart.getProducts()
```

```

    ctx.body = { products }
  })

  /**
   * 添加购物车
   */
  router.post('/cart', async ctx => {
    const body = ctx.request.body
    console.log('ctx.body', ctx.request.body)
    const prodId = body.id;
    let fetchedCart;
    let newQty = 1;

    // 获取购物车
    const cart = await ctx.user.getCart()
    console.log('cart', cart)
    fetchedCart = cart;
    const products = await cart.getProducts({
      where: {
        id: prodId
      }
    });

    let product;
    // 判断购物车数量
    if (products.length > 0) {
      product = products[0];
    }
    if (product) {
      const oldQty = product.cartItem.quantity;
      newQty = oldQty + 1;
      console.log("newQty", newQty);
    } else {
      product = await Product.findByPk(prodId);
    }

    await fetchedCart.addProduct(product, {
      through: {
        quantity: newQty
      }
    });
    ctx.body = { success: true }
  })

  /**
   * 添加订单
   */
  router.post('/orders', async ctx => {
    let fetchedCart;
    const cart = await ctx.user.getCart();
    fetchedCart = cart;
    const products = await cart.getProducts();
    const order = await ctx.user.createOrder();
  })

```

```

    const result = await order.addProducts(
      products.map(p => {
        p.orderItem = {
          quantity: p.cartItem.quantity
        };
        return p;
      })
    );
    await fetchedCart.setProducts(null);
    ctx.body = { success: true }
  })

  /**
   * 删除购物车
   */
  router.delete('/cartItem/:id', async ctx => {
    const id = ctx.params.id
    const cart = await ctx.user.getCart()
    const products = await cart.getProducts({
      where: { id }
    })
    const product = products[0]
    await product.cartItem.destroy()
    ctx.body = { success: true }
  })

  /**
   * 查询订单
   */
  router.get('/orders', async ctx => {
    const orders = await ctx.user.getOrders({ include: ['products'], order: [['id', 'DESC']] })
    ctx.body = { orders }
  })

  app.use(router.routes())

```