**//jshint esversion:6**

**const express = require("express"); //подключение express**

**const bodyParser = require("body-parser"); //подключение body-parser**

**const ejs = require("ejs"); //подключение ejs**

**const \_ = require("lodash"); //подключение lodash**

**const mongoose = require("mongoose"); //подключение mongoose**

**const app = express(); //иницализация app express**

**app.set('view engine', 'ejs'); // настройка движка отображения  ejs**

**app.use(bodyParser.urlencoded({ extended: true })); // настройка bodyParser**

**app.use(express.static("public")); // подключение папки public для хранения кастомных стилей, скриптов и т.д.**

**const homeStartingContent = "Lacus vel ..."; // переменные**

**const aboutContent = "Hac habitasse ...";**

**mongoose.connect("mongodb://localhost:27017/ljBlogDB", { useNewUrlParser: true, useUnifiedTopology: true }); // подключение (создание) к БД ljBlogDB на локальном сервере**

**const homePostsSchema = new mongoose.Schema({**

**title: String,**

**content: String**

**}) // настройка схемы коллекции**

**const HomePost = mongoose.model("HomePost", homePostsSchema); // инициализация схемы коллекции HomePost**

**app.get("/", function (req, res) {**

**HomePost.find(function (err, foundData) {**

**if (!err) {**

**res.render("home", {**

**startingContent: homeStartingContent,**

**posts: foundData**

**}); // отображение контента**

**}**

**}) // поиск контента в коллекции   HomePost**

**}); // ответ сервера на get запрос клиента-браузера по корневому адресу**

**app.get("/about", function (req, res) {**

**res.render("about", { aboutContent: aboutContent }); // отображение с передачей объекта данных для обработки EJS**

**}); // ответ сервера на get запрос клиента-браузера по адресу /about**

**app.get("/compose", function (req, res) {**

**res.render("compose"); // отображение**

**}); // ответ сервера на get запрос клиента-браузера по адресу /compose**

**app.post("/compose", function (req, res) {**

**const newPost = new HomePost({**

**title: req.body.postTitle,**

**content: req.body.postBody**

**}) // подготовка данных на основе схемы коллекции для записи в БД**

**newPost.save(); // запись данных в БД**

**res.redirect("/"); // переадресация в корень**

**});**

**app.get("/posts/:postId", function (req, res) {**

**const postId = req.params.postId; // назначение переменной части адресной строки**

**HomePost.findById(postId, function (err, postData) {**

**if (!err) {**

**res.render("post", {**

**title: postData.title,**

**content: postData.content**

**}); // отображение с передачей объекта данных для обработки EJS**

**}**

**else{**

**res.render('error')**

**} // отображение ошибки, если нет страницы с этим адресом в БД**

**})**

**});**

**app.listen(3000, function () {**

**console.log("Server started on port 3000");**

**}); // прослушивание локального порта — вывод сообщения при готовности сервера**