**mongoose.connect("mongodb://localhost:27017/wikiDB", { useNewUrlParser: true, useUnifiedTopology: true });**

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

**title: String,**

**content: String**

**})**

**const Article = mongoose.model("Article", articleSchema);**

**//ПРИНЦИПЫ RESTfull приложения:**

**app.route("/articles") //инициализация цепочки роутинга для адреса /articles**

**.get(function (req, res) {//GET запрос на URL первого уровня (**[**http://site.com*/articles***](http://site.com/articles)**) должен выводить все статьи**

**Article.find(function (err, founddata) {**

**if (!err) {**

**res.send(founddata);**

**}**

**else {**

**console.log(err)**

**}**

**})**

**})**

**.post(function (req, res) {/POST запрос на URL первого уровня (**[**http://site.com*/articles***](http://site.com/articles)**) должен записывать новую статью в БД**

**const newArt = new Article({**

**title: req.body.title,**

**content: req.body.content**

**});**

**newArt.save(function (err) {**

**if (!err) {**

**console.log("Successfully added!")**

**}**

**else {**

**console.log(err)**

**}**

**})**

**})**

**.delete(function (req, res) {/DELETE запрос на URL первого уровня**

**(**[**http://site.com*/articles***](http://site.com/articles)**) должен удалить все статьи из БД**

**Article.deleteMany(function (err) {**

**if (!err) {**

**console.log("Deleted succesfully!")**

**}**

**else {**

**console.log(err)**

**}**

**})**

**});**

**app.route("/articles/:articleTitle") //инициализация цепочки роутинга для адреса /articles/article-name**

**.get(function (req, res) {//GET запрос на URL второго уровня**

**(**[**http://site.com*/articles/article-name***](http://site.com/articles/article-name)**) должен выводить указанную в адресе статью**

**Article.findOne({ title: req.params.articleTitle }, function (err, foundData) {**

**if (!err) {**

**res.send(foundData)**

**}**

**else {**

**console.log(err)**

**}**

**})**

**})**

**.put(function (req, res) { //PUT запрос на URL второго уровня**

**(**[**http://site.com*/articles/article-name***](http://site.com/articles/article-name)**) должен перезаписывать указанную в адресе статью**

**Article.updateOne(**

**{ title: req.params.articleTitle },**

**{ title: req.body.title, content: req.body.content },**

**{ overwrite: true },**

**function (err) {**

**if (!err) {**

**console.log("Good!")**

**}**

**else {**

**console.log(err)**

**}**

**})**

**})**

**.patch(function (req, res) {//PATCH запрос на URL второго уровня**

**(**[**http://site.com*/articles/article-name***](http://site.com/articles/article-name)**) должен редактировать указанную в адресе статью и перезаписывать только указанные в $set поля**

**Article.updateOne(**

**{ title: req.params.articleTitle },**

**{ $set: req.body },**

**function (err) {**

**if (!err) {**

**res.send("Good!")**

**}**

**else {**

**res.send(err)**

**}**

**})**

**});**