|  |  |
| --- | --- |
| TOPIC/TITLE  POST a new article | |
| Keywords/Questions | Notes : |
|  |
| We don’t necessarily need to have a front end with HTML to test our API we can use POSTMAN as a client |
|  |
|  |
|  |
| Our goal in Postman is to make a post request to our server |
| First we have to code it in our app.js |
|  |
|  |
| The route has to be the same as the get request to be RESTful |
| First use console log to see if the post request is working |
| app.post("/articles",function(req,res){  console.log(req.body.title);  console.log(req.body.content);  }) |
| Then go to postman and select post instead of get and type your serevr address |
| localhost:3000/articles |
| Then go to the body tab and change the form to x-www-form-urlencoded (this is what bodyparser is designed to handle. |
| We can add in those wariable names that we define in our app.js (title and content) |
| The value is the data we ar going to send over along with our post |
|  |
|  |
| Now we see it can work we can send the data to mongoDB |
|  |
|  |
| app.post("/articles",function(req,res){    const newArticle = new Article ({  title: req.body.title,  content: req.body.content  });  newArticle.save();  }) |
| However we have to add a call back function in our save method  app.post("/articles",function(req,res){  const newArticle = new Article ({  title: req.body.title,  content: req.body.content  });  newArticle.save(function(err){  if(!err){  res.send("Success");  }else{  res.send(err);  };  });  }); |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | |