



Frameworks WEB
Criando a estrutura de rotas
Método POST

Sistemas de Informação - UNISANTA



Índice

Routes.....	3
Criando a estrutura de rotas para os endpoints.....	3
Criando a rota para o endpoint /api/post.....	4
Testando o endpoint /api/post.....	5
Histórico de revisões.....	6



Routes

Criando a estrutura de rotas para os endpoints

Segue tabela com os endpoints dos serviços a serem disponibilizados por nossa API:

Endpoint	Método HTTP	Objetivo
/api/getAll	GET	Obter todas as tarefas (itens)
/api/getOne/id	GET	Obtém uma tarefa identificada pelo id
/api/post	POST	Grava uma nova tarefa
/api/update/id	PATCH	Atualiza a tarefa identificada pelo id
/api/delete/id	DELETE	Elimina a tarefa identificada pelo id

Para cada endpoint deverá ser criada uma rota para tratar as solicitações realizadas pelo cliente. Para isso, vamos criar um arquivo chamado **"/routes/routes.js"** dentro da pasta raiz do projeto, com o seguinte conteúdo:

```
const express = require('express');  
  
const router = express.Router()  
  
module.exports = router;
```

Agora vamos adicionar as seguintes linhas no arquivo **index.js**, logo após a linha **app.use(express.json());** para direcionar todos os endpoints iniciados por **"/api"** para o arquivo **routes.js**:

```
const routes = require('./routes/routes');  
app.use('/api', routes)
```



Criando a rota para o endpoint /api/post

Antes de criar a rota do primeiro endpoint, vamos definir o modelo dos dados a serem armazenados no MongoDB. Para isso vamos criar um arquivo chamado de **"/models/task.js"** com o seguinte conteúdo:

```
const mongoose = require('mongoose');

const taskSchema = new mongoose.Schema({
  description: {
    required: true,
    type: String
  },
  done: {
    required: true,
    type: Boolean
  },
  {
    versionKey: false
  }
})

module.exports = mongoose.model('Task', taskSchema)
```

O modelo acima define a criação de uma **Collection** chamada "tasks" com os seguintes campos para cada **Document**:

- description: String
- done: Boolean

Agora vamos inserir a rota para o método "POST" ao final do arquivo **"routes.js"**:

```
//Post Method
const taskModel = require('../models/task');
router.post('/post', async (req, res) => {
  const data = new taskModel({
    description: req.body.description,
    done: req.body.done
  })
  try {
    const dataToSave = await data.save();
    res.status(200).json(dataToSave)
  }
  catch (error) {
    res.status(400).json({message: error.message})
  }
})
```



Testando o endpoint `/api/post`

Utilize o programa Postman para testar o endpoint **`/api/post`**.

Utilize o método POST passando um JSON com a descrição e estado (done) da tarefa no corpo (body) da requisição.

Entre em sua conta no **MongoDB Atlas** para constatar que o dado foi corretamente gravado.



Histórico de revisões

Revisão: 00

Data: 21/03/2022

Descrição das alterações:
Documento original