DAY34 FormData、Token

2023年6月2日 18:43

FormData构造函数

- **FormData对象**
- 可以将form表单元素的name与value进行组合，实现表单数据的序列化，从而简化表单元素的拼接
- formData仅支持post请求 无需设置请求头

1.表单必须添加name属性

2.利用formdata将表单的数据自动拼接成ajax发送的格式(name=value&name=value)

前端

```
<body>
  <form id="form1">
    用户名：<input type="text" name="username">
    密码：<input type="password" name="password">
    <input type="button" value="提交" class="btn">
  </form>
</body>
```

```
const btn = document.querySelector('.btn');
const form1 = document.querySelector('#form1');
btn.onclick = function () {
  let formdata = new FormData(form1); //创建formdata实例，将表单当作参数，自动拼接name以及value
```

介绍formdata下面的核心方法

append方法：给表单对象追加值

```
// formdata.append('tel', '13312341234');
```

set方法：给表单对象设置值(区别：set如果属性存在，覆盖属性，append不会覆盖，都会传输)

```
// formdata.set('username', '尼古拉斯')
```

get方法：获取表单对象设置的值

```
// console.log(formdata.get('username'));
```

delete方法：删除表单对象的值。

```
    // formdata.delete('username');
    // formdata.delete('password');
    const xhr = new XMLHttpRequest();
    xhr.open('POST', 'http://localhost:5000/formdata');
    xhr.onload = function () {
      console.log(xhr.responseText);
    }
    xhr.send(formdata);
  };
```

文件上传-前端

```
    <link href="https://cdn.bootcdn.net/ajax/libs/twitter-
  bootstrap/3.4.1/css/bootstrap.min.css" rel="stylesheet">

  <body>
    <div class="container">
      <div class="uplaod">
        <input type="file" id="file">
      </div>
      <br><br>
      <div class="progress">
        <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-
  valuemin="0" aria-valuemax="100"
          style="width: 0%;">
          0%
        </div>
      </div>
      <br><br>
      <div id="pic">
      </div>
    </div>
  </body>
```

服务端

```
  const express = require("express");
  const formidable = require("formidable");
  const path = require("path");
  const app = express();
  // 加载静态资源(前端页面)
  app.use(express.static(path.join(__dirname, "public")));
```

1.formdata的接口，获取前端传入的值

使用body-parser第三方模块无法获取formdata传入的值

使用另外的第三方的模块formidable进行获取

```
  app.post("/formdata", (req, res, next) => {
    let fileData = new formidable.IncomingForm();
    fileData.parse(req, (err, fields, files) => {
```

```
    console.log(fields); //获取前端的数据{ username: 'zhangsan', password:
'123456' }
    res.send(fields);
  });
});


// 2.获取前端传入的文件
app.post("/upload", (req, res, next) => {
  let fileData = new formidable.IncomingForm();
  // 设置接收文件的目录
  fileData.uploadDir = path.join(__dirname, "public", "upload");
  // 保持文件的扩展名
  fileData.options.keepExtensions = true;
  fileData.parse(req, (err, fields, files) => {
    console.log("/upload/" + files.attrName.newFilename);
    res.send({
      //将前端上传的图片继续给前端，让前端渲染到页面
      url: "/upload/" + files.attrName.newFilename,
    });
  });
});
app.listen(5000);
console.log("服务器启动成功,监听的端口是5000");
```

mongodb概述

数据库：仓储数据的仓库。

一.列举常见的关系和非关系型数据库

1.关系型：Oracle、DB2、Microsoft SQL Server、MySQL等

2.非关系型：Nosql mongodb

二.NoSQL，指的是非关系型的数据库,是对不同于传统的关系型数据库的数据库管理系统的统称。

三.MongoDB简介

MongoDB 是非关系型数据库当中功能最丰富，最像关系型数据库的。

BSON是一种类似json的二进制形式的存储格式，简称Binary JSON，它和JSON一样，支持内嵌的文档对象和数组对象，但是BSON有JSON没有的一些数据类型，如Date和BinData类型。

四.MongoDB下载与安装

- MongoDB下载地址：https://www.mongodb.com/try/download/community
- 根据你的系统下载 32 位或 64 位的.msi 文件，下载后双击该文件，按操作提示安装即可。
- mac下mongodb安装和配置

安装流程

http://c.biancheng.net/mongodb2/install-on-windows.html window
https://www.runoob.com/mongodb/mongodb-osx-install.html mac安装流程

五.mongodb的应用

1.终端命令行操作 - 了解使用

关系型数据库：数据库->表(table)->数据(表格行列组成，行：字段，列：记录)

非关系型数据库：数据库->集合(collections)->文档(document)

六.可视化工具操作数据库

1.Robo 3T是mongodb数据库的可视化工具

2.直接从软件资料复制软件，无需安装，解压后就可以使用。

1.点击Download Robo 3T - https://robomongo.org/

2.选择操作系统下载对应版本并安装

3.点击：File / Connect调出MonogoDB Connections窗口

4.点击Create 设置 连接MongoDB服务器，输入mongoDB的地址和端口号

mongodb和可视化操作工具Robo 3T已连接成功，cmd命令行或者Robo 3T里面操作，数据均可同步

5.Mac下Robo 3T(MongoDB可视化工具)安装和使用

token

一.token介绍

- 作为计算机术语时，是“令牌”的意思。
- Token是服务端生成的一串字符串，以作客户端进行请求的一个令牌
- 当第一次登录后，服务器生成一个Token便将此Token返回给客户端，
- 以后客户端只需带上(请求头带上)这个Token前来请求数据即可，无需再次带上用户名和密码。

二.token是用来干嘛的

- 使用token机制的身份验证方法，在服务器端不需要存储用户的登录记录。

三.token的大概流程

1.客户端使用用户名和密码请求登录，服务端收到请求，验证用户名和密码。

2.验证成功后，服务端会生成一个token，然后把这个token发送给客户端。

3.客户端收到token后把它存储起来，可以放在cookie或者Local Storage（本地存储）里。

4.客户端每次向服务端发送请求的时候都需要带上服务端发给的token。

5.服务端收到请求，然后去验证客户端请求里面的token，如果验证成功，就向客户端返回请求的数据。

四.了解后端token的生成

```
// 使用第三方工具:jsonwebtoken
const jwt = require("jsonwebtoken");
// 生成token函数
// token的组成
// jwt主要由三个部分组成：头部(HEADER)，载荷(PAYLOAD)，签证(SIGNATURE)。
function createToken(uid) {
  const token = jwt.sign(
    {
      //用户id
      uid,
    },
    "尼古拉斯", //秘密
    {
      expiresIn: 86400, //过期时间
    }
  );
  return token;
}
let token = createToken("1"); //生成token，参数一般都是用户的id
console.log(token);
// 校验函数
function checkToken(token) {
  try {
    return jwt.verify(token, "尼古拉斯");
  } catch (error) {
    console.log(error);
  }
}
console.log(checkToken(token + "123")); //返回对象视为匹配成功，否则
console.log(1);
```