

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Факультет Информатики   
Кафедра Программных систем

**ОТЧЕТ**  
  
по лабораторному практикуму по дисциплине

«Компьютерные сети»

Лабораторная работа №3

Вариант № 1

Студент А. Д. Аганин

Руководитель Д. С. Пигусов

Самара 2022

Содержание

[Содержание 2](#_Toc97459822)

[Теория 3](#_Toc97459823)

[1 Задание 3](#_Toc97459824)

[2 Результат работы программы 3](#_Toc97459827)

[Список использованных источников 4](#_Toc97459828)

[Приложение А. Исходный код программы. 5](#_Toc97459829)

Теория

1. Задание

Разработать клиентское приложение, позволяющее пользователю получать следующую информацию о FTP-сервере:

- суммарный размер файлов, размещенных на сервере;

- суммарный размер размещенных на сервере файлов, сгруппированных по их типам (тип файлов определяется по его расширению);

- структуру каталогов FTP-сервера.

Адрес FTP-сервера, имя пользователя и пароль вводятся пользователем.

1. Результат работы программы

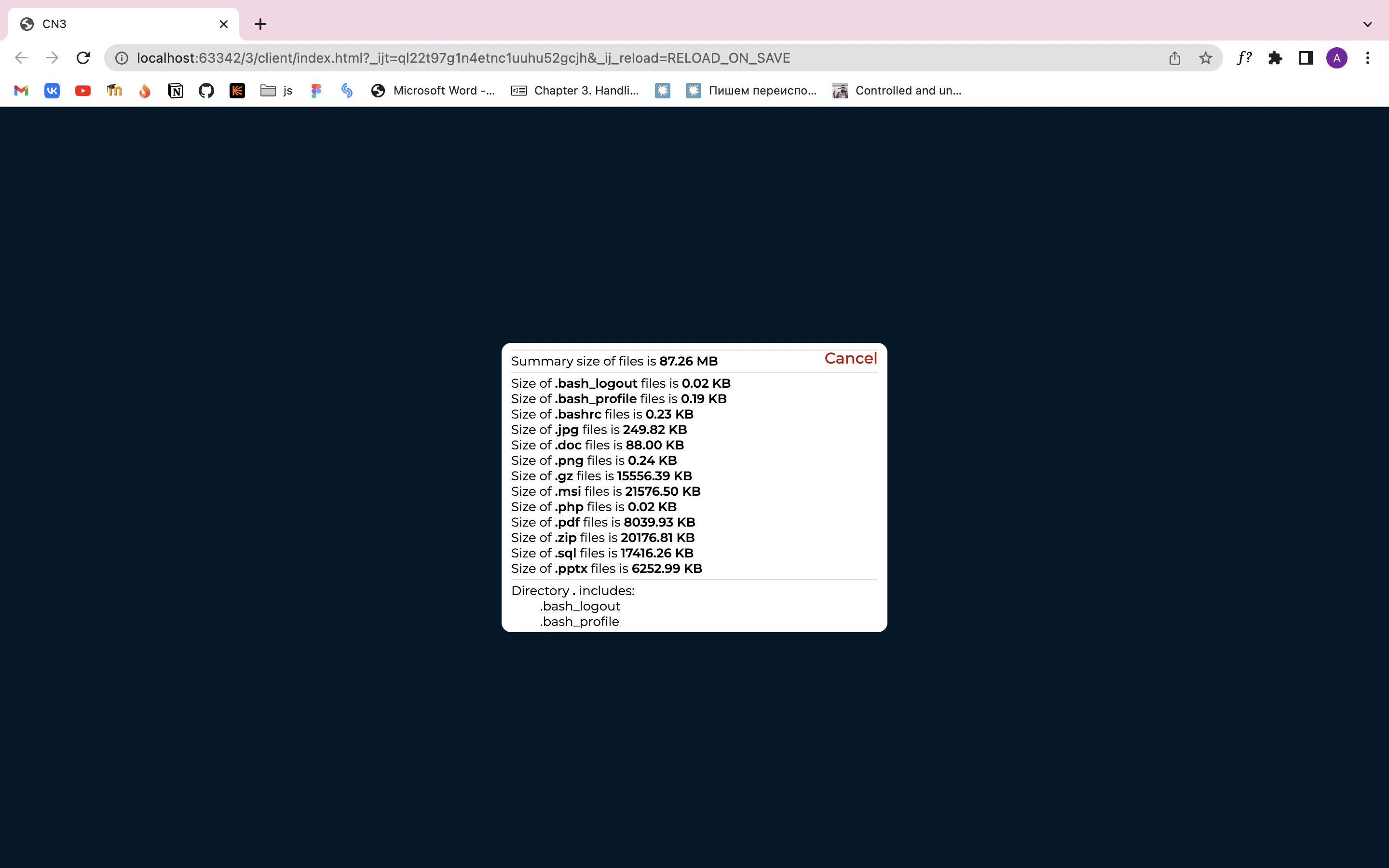


Рисунок 1 – пример работы приложения

Список использованных источников

1. File Transfer Protocol – Wikipedia [Электронный ресурс]. – Дата обращения 01.04.2022;
2. Еленев Д.В. Работа с сервисами сети интернет. - Самара, 2010.

Приложение А. Исходный код программы.

Файл client.js

**const *xhr*** = **new *XMLHttpRequest***()  
**const *button*** = ***document***.querySelector(**'.button-login'**)  
**const *inputHost*** = ***document***.querySelector(**'.input\_host'**)  
**const *inputLogin*** = ***document***.querySelector(**'.input\_login'**)  
**const *inputPassword*** = ***document***.querySelector(**'.input\_password'**)  
**const *buttonCancel*** = ***document***.querySelector(**'.popup\_button\_close'**)  
  
***button***.addEventListener(**'click'**, () => {  
 ***xhr***.open(**'POST'**, **'http://localhost:3000/api/ftp'**)  
  
 ***xhr***.**onload** = () => {  
 **if**(!***JSON***.parse(***xhr***.**response**).**message**) {  
 **const** list = ***JSON***.parse(***xhr***.**response**).**files  
 *console***.log(list)  
 **let** summarySize = 0,  
 typeSizeList = [],  
 pathsList = [{**dirName**: **'.'**, **elements**: []}]  
  
 **for** (**let** i = 0; i < list.**length**; i++) {  
 **if** (list[i].**type** != **'2'**) {  
 summarySize = *calcSummarySizeOfFiles*(summarySize, list[i].**size**)  
 *calcSizeOfEachType*(typeSizeList, list[i])  
 *buildCatalog*(pathsList, list[i])  
  
 } **else** {  
 pathsList.push({**dirName**: list[i].**name**, **elements**: []})  
 }  
 }  
  
 *printSummarySizeOfFiles*(summarySize, ***document***.querySelector(**'.info\_container\_size'**))  
 *printSizeOfEachType*(typeSizeList, ***document***.querySelector(**'.info\_container\_typesize'**))  
 *printCatalog*(pathsList, ***document***.querySelector(**'.info\_container\_catalog'**))  
 ***console***.log(summarySize)  
  
 ***document***.querySelector(**'.container'**).**style**.**visibility** = **'hidden'  
 *document***.querySelector(**'.popup\_background'**).**style**.**visibility** = **'visible'** } **else**{  
 ***console***.log(***JSON***.parse(***xhr***.**response**).**message**)  
 }  
 }  
 ***xhr***.setRequestHeader(**'Content-Type'**, **'application/json'**)  
  
 ***xhr***.send(***JSON***.stringify({  
 **"host"**: ***inputHost***.**value** || **"91.222.128.11"**,  
 **"user"**: ***inputLogin***.**value** || **"testftp\_guest"**,  
 **"password"**: ***inputPassword***.**value** || **"12345"** }  
 ))  
})  
  
***buttonCancel***.addEventListener(**'click'**, () => {  
 ***document***.querySelector(**'.container'**).**style**.**visibility** = **'visible'  
 *document***.querySelector(**'.popup\_background'**).**style**.**visibility** = **'hidden'**})  
  
*//main calc functions***function** *calcSummarySizeOfFiles*(summarySize, listSize) {  
 summarySize += listSize  
 **return** summarySize  
}  
  
**function** *calcSizeOfEachType*(typeSizeList, list) {  
 **let** type = list.**name**.substr(list.**name**.lastIndexOf(**'.'**), list.**name**.**length**)  
 **if** (!typeSizeList.some(elem => {  
 **return** elem.**name** === type  
 })) {  
 typeSizeList.push({**name**: type, **size**: list.**size**})  
 } **else** {  
 (typeSizeList.some(elem => {  
 **if** (elem.**name** === type) elem.**size** += list.**size** }  
 ))  
 }  
}  
  
**function** *buildCatalog*(pathsList, list) {  
 **for** (**let** j = 0; j < list.**parents**.**length** - 1; j++) {  
 pathsList.find(elem => elem.**dirName** == list.**parents**[j]).**elements**.push(list.**name**)  
 }  
}  
  
*//print functions***function** *printSummarySizeOfFiles*(size, div) {  
 div.**innerHTML**=**''  
 const** span = ***document***.createElement(**'span'**)  
 span.**innerHTML** = **`Summary size of files is <b>**${(size / 1024 / 1024).toFixed(2)} **MB</b>`** div.append(span)  
}  
  
**function** *printSizeOfEachType*(list,div){  
 div.**innerHTML**=**''  
 for**(**let** i=0; i<list.**length**;i++) {  
 **const** span = ***document***.createElement(**'span'**)  
 span.**innerHTML** = **`Size of <b>**${list[i].**name**}**</b> files is <b>**${(list[i].**size**/1024).toFixed(2)} **KB</b>`** div.append(span)  
 }  
}  
  
**function** *printCatalog*(list,div){  
 div.**innerHTML**=**''  
 for**(**let** i=0; i<list.**length**;i++) {  
 **const** dirName = ***document***.createElement(**'span'**)  
 **if**(list[i].**elements**.**length**==0) dirName.**innerHTML** = **`Directory <b>**${list[i].**dirName**}**</b> is empty<br>`  
 else** dirName.**innerHTML** = **`Directory <b>**${list[i].**dirName**}**</b> includes: <br>`** div.append(dirName)  
 **for**(**let** j=0; j<list[i].**elements**.**length**;j++) {  
 **const** file = ***document***.createElement(**'span'**)  
 file.**className**=**'file'** file.**innerHTML** += **`**${list[i].**elements**[j]} **<br>`** div.append(file)  
 }  
  
 }  
  
}

Файл index.html

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>CN3</**title**>  
 <**link rel="stylesheet" href="styles.css"**>  
 <**link rel="preconnect" href="https://fonts.googleapis.com"**>  
 <**link rel="preconnect" href="https://fonts.gstatic.com" crossorigin**>  
 <**link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap" rel="stylesheet"**>  
</**head**>  
<**body**>  
<**div class="wrapper"**>  
 <**div class="container"**>  
 <**h1 class="title"**>Research of FTP server</**h1**>  
 <**div class="input\_container"**>  
 <**input type="text" class="input input\_host" placeholder="Host"**>  
 <**input type="text" class="input input\_login" placeholder="Login"**>  
 <**input type="text" class="input input\_password" placeholder="Password"**>  
 </**div**>  
 <**button class="button-login"**>Login</**button**>  
 </**div**>  
</**div**>  
  
<**div class="popup\_background"**>  
 <**div class="popup\_container"**>  
 <**span class="popup\_button\_close"**>Cancel</**span**>  
 <**div class="popup\_container\_content"**>  
 <**div class="info\_container info\_container\_size"**></**div**>  
 <**div class="info\_container info\_container\_typesize"**></**div**>  
 <**div class="info\_container info\_container\_catalog"**></**div**>  
 </**div**>  
  
 </**div**>  
</**div**>  
  
</**body**>  
  
<**script src="client.js"**></**script**>  
</**html**>

Файл styles.css

**body**{  
 **background-color**: **#011E33**;  
 **color**: **white**;  
  
}  
**\***{  
 **font-family**: **'Montserrat'**,**sans-serif**;  
 **margin**: 0;  
}  
  
.**wrapper**{  
 **position**: **absolute**;  
 **top**: 0;  
 **width**: 100%;  
 **height**: 100%;  
 **display**: **flex**;  
 **align-items**: **center**;  
 **justify-content**: **center**;  
}  
  
.**container**,.**input\_container**{  
 **display**: **flex**;  
 **flex-direction**: **column**;  
 **align-items**: **center**;  
}  
  
.**input**{  
 **color**: **white**;  
 **background-color**: **transparent**;  
 **margin**:0;  
 **padding**: 4**px**;  
 **width**: 377**px**;  
 **height**: 27**px**;  
 **border**: **none**;  
 **border-bottom**: 1**px white solid**;  
}  
.**input**::**-webkit-inner-spin-button** {  
 **-webkit-appearance**: **none**;  
 **margin**: 0;  
}  
.**input**::**placeholder** {  
 **color**: **white**;  
 **font-weight**: 400;  
 **font-size**: 15**px**;  
 **line-height**: 18**px**;  
}  
.**input**:**focus**{  
 **outline**: **none**;  
 **color**:**white**;  
}  
  
.**title**{  
 **font-weight**: 600;  
 **font-size**: 36**px**;  
 **line-height**: 44**px**;  
  
 **color**: **#FFFFFF**;  
}  
.**button-login**{  
 **width**: 377**px**;  
 **height**: 27**px**;  
 **margin-top**: 21**px**;  
 **cursor**: **pointer**;  
  
 **font-weight**: 600;  
 **color**: **white**;  
  
 **background**: **#478FFB**;  
 **border-radius**: 10**px**;  
 **border**: **none**;  
}  
  
.**popup\_background**{  
 **visibility**: **hidden**;  
 **transition**: **opacity** 1**s ease-out**;  
 **position**: **absolute**;  
 **top**: 0;  
 **background-color**: **rgba**(0,0,0,.2);  
 **width**: 100%;  
 **height**: 100%;  
 **display**: **flex**;  
 **align-items**: **center**;  
 **justify-content**: **center**;  
}  
  
.**popup\_container**{  
 **position**: **relative**;  
 **border-radius**: 10**px**;  
 **background-color**: **white**;  
 **height**: 300**px**;  
 **width**: 400**px**;  
}  
.**popup\_container\_content**{  
 **overflow**: **auto**;  
 **height**: 100%;  
}  
  
.**popup\_button\_close**{  
 **color**: **#BC0000**;  
 **font-weight**: 500;  
 **position**: **absolute**;  
 **right**: 0;  
 **margin**: 6**px** 10**px** 0 0;  
 **cursor**: **pointer**;  
}  
  
.**info\_container\_size**{  
 **border-top**: 1**px solid #DEDEDE**;  
 **margin-top**: 27**px**;  
}  
  
.**info\_container**{  
 **color**: **black**;  
 **font-size**:13**px**;  
 **display**: **flex**;  
 **flex-direction**: **column**;  
 **padding**: 3**px** 0 ;  
 **overflow**: **auto**;  
 **margin-left**: 10**px**;  
 **width**: 380**px**;  
}  
  
.**file**{  
 **margin-left**: 30**px**;  
}  
  
.**info\_container**:**not**(:**last-child**){  
 **border-bottom**: 1**px solid #DEDEDE**;  
}

Файл [ftp.routes.js](ftp://ftp.routes.js)

**const *Router*** = require(**'express'**)  
**const** ftp = require(**'basic-ftp'**)  
**let** files = []  
  
**const *router*** = **new *Router***()  
  
***router***.**post**(**'/ftp'**, **async** (req, res) => {  
 files=[]  
 **await** ftpConnect(req, res)  
 **return** res.**json**({files})  
})  
  
**async function** getList(client, str = **'./'**) {  
 **const** list = **await** client.**list**(str)  
 **for** (**let** i = 0; i < list.**length**; i++) {  
 list[i].**parents** = str.split(**'/'**)  
 files.push(list[i])  
  
 **if** (list[i].**type** == **'2'**) {  
 ***console***.log(**`Changed to** ${list[i].**name**}**`**)  
 **await** getList(client, str + list[i].**name** + **'/'**)  
 }  
 }  
}  
  
**async function** ftpConnect(req, res) {  
 **const** client = **new** ftp.Client()  
 **const** {host, user, password} = req.**body  
 *console***.log(host, user, password)  
 client.**ftp**.**verbose** = **true  
 try** {  
 **await** client.access({  
 **host**: host,  
 **user**: user,  
 **password**: password,  
 **secure**: **false** })  
 **await** getList(client)  
 } **catch** (err) {  
 ***console***.log(err)  
 **return** res.json({**message**: err})  
 }  
 client.close()  
}  
  
module.**exports** = ***router***

Файл app.js

**const *express*** = require(**'express'**)  
**const *authRouter*** = require(**'./routes/ftp.routes.js'**)  
  
**const** app = ***express***()  
**const** PORT = 3000  
  
app.**use**(require(**'cors'**)())  
app.**use**(***express***.***json***())  
app.**use**(**"/api/"**, ***authRouter***)  
  
  
app.listen(PORT, (error) => {  
 error ? ***console***.log(error) : ***console***.log(**`Server is listening on** ${PORT}**...`**)  
})