Node.js-HTTP-клиент

1. **HTTP-клиент:** клиентская часть web-приложения
2. **HTTP-клиент:** простейший клиент

```
var http = require('http');

let  options = {
    host:'localhost',
    path:'/mypath',
    port: 3000,
    method:'GET'
}
const req = http.request(options, (res) => {

    console.log('http.request: method = ', req.method);
    console.log('http.request: response:', res.statusCode);
    console.log('http.request: statusMessage:', res.statusMessage);
    console.log('http.request: socket.remoteAddress:', res.socket.remoteAddress);
    console.log('http.request: res.socket.remotePort:', res.socket.remotePort);
    console.log('http.request: res.headers:', res.headers);

    let data = '';
    res.on('data', (chunk) => {
        console.log('http.request: data: body =', data += chunk.toString('utf8'));
    });
    res.on('end', () => { console.log('http.request:  end: body =', data);  });

});

req.on('error', (e) =>  {  console.log('http.request: error:', e.message);});
req.end();
```

3. **HTTP-клиент:** простейший клиент, GET, параметры

```
let http = require('http');
let query = require('querystring');


let parms =  query.stringify({x:3, y:4, s:'xxx'});
let path  = `/mypath?${parms}`

console.log('parms',parms);
console.log('path', path);

let  options = {
    host:'localhost',
    path: path,
    port: 3000,
    method:'GET'
}
const req = http.request(options, (res) => {

    let data = '';
    res.on('data', (chunk) => {
        console.log('http.request: data: body =', data += chunk.toString('utf8'));
    });
    res.on('end', () => { console.log('http.request:  end: body =', data);  });

});

req.on('error', (e) =>  {  console.log('http.request: error:', e.message);});
req.end();
```

```
D:\PSCA\Lec08>node 08-02
parms x=3&y=4&s=xxx
path /mypath?x=3&y=4&s=xxx
http.request: data: body = <h1>GET-параметры</h1>
http.request: data: body = <h1>GET-параметры</h1>href: /mypath?x=3&y=4&s=xxx<br/>path
= 3<br/>y = 4<br/>s = xxx<br/>
http.request:  end: body = <h1>GET-параметры</h1>href: /mypath?x=3&y=4&s=xxx<br/>path
= 3<br/>y = 4<br/>s = xxx<br/>
```

4. **HTTP-клиент:** простейший клиент, POST, параметры

```
let http = require('http');
let query = require('querystring');


let parms =  query.stringify({x:3, y:4, s:'xxx'});
console.log('parms', parms);

let  options = {
    host:'localhost',
    path: '/mypath',
    port: 3000,
    method:'POST'
}
const req = http.request(options, (res) => {

    let data = '';
    res.on('data', (chunk) => {
        console.log('http.request: data: body =', data += chunk.toString('utf8'));
    });
    res.on('end', () => { console.log('http.request:  end: body =', data);  });

});

req.on('error', (e) =>  { console.log('http.request: error:', e.message);});

req.write(parms);

req.end();
```

```
D:\PSCA\Lec08>node 08-03
parms x=3&y=4&s=xxx
http.request: data: body = <h1>URL-параметры</h1>
http.request: data: body = <h1>URL-параметры</h1>x=3&y=4&s=xxxx = 3<br />y = 4<br />s = xxx<br />
http.request:  end: body = <h1>URL-параметры</h1>x=3&y=4&s=xxxx = 3<br />y = 4<br />s = xxx<br />
```

5. **HTTP-клиент:** простейший клиент, JSON

```
let http = require('http');

let jsonm = JSON.stringify({x:1, y:2, s:'sss'});
let  options = {
    host:'localhost',
    path: '/',
    port: 3000,
    method:'POST',
    headers:{
        'content-type':'application/json', 'accept':'application/json'
    }
}
const req = http.request(options, (res) => {

    let data = '';
    res.on('data', (chunk) => {  data += chunk.toString('utf8'); });
    res.on('end', () => {
        console.log('http.response:  end: body =', data);
        console.log('http.response:  end: parse(boby) =',  JSON.parse(data));
    });

});

req.on('error', (e) =>  { console.log('http.request: error:', e.message);});
req.end(jsonm);
```

6. **HTTP-клиент:** простейший клиент, XML

```
let http        = require('http');
let xmlbuilder  = require('xmlbuilder');
let parseString = require('xml2js').parseString;

let xmldoc  = xmlbuilder.create('students').att('faculty', 'ИТ').att('speciality','ИСиТ');
                    xmldoc.ele('student').att('id', '7000222').att('name','Иванов И.И.').att('bday','2000-12-02')
                     .up().ele('student').att('id', '7000223').att('name','Петров П.П.').att('bday','2000-11-29')
                          .txt('Прошел собеседование в iTechArt')
                     .up().ele('student').att('id', '7000228').att('name','Казан Н.А.') .att('bday','2001-09-11');


let  options = {
    host:'localhost',  path: '/',  port: 3000,  method:'POST',
    headers:{'content-type':'text/xml', 'accept':'text/xml'}
}

const req = http.request(options, (res) => {
    let data = '';
    res.on('data', (chunk) => {  data += chunk; });
    res.on('end', () => {
        console.log('http.response:  end: body =', data);
        parseString(data, (err, str)=> {
            if (err)  console.log('xml parse error');
            else {
                console.log('str =', str);
                console.log('str.result =', str.result);
            }
        })
    });
});

req.on('error', (e) => {console.log('http.request: error:', e.message);});
req.end(xmldoc.toString({pretty:true}));
```

7. **HTTP-клиент:** простейший клиент, upload

```
let http = require('http');

let bound = 'smw60-smw60-smw60';
let body  =  `--${bound}\r\n`;
    body +=  'Content-Disposition:form-data; name="file"; filename="MyFile.txt"\r\n';
    body +=  'Content-Type:text/plain\r\n\r\n';
    body +=  '11111\n22222\n33333\n44444';                    //  данные из файла
    body +=  `\r\n--${bound}--\r\n`

let  options = {
    host:'localhost',
    path: '/mypath',
    port: 3000,
    method:'POST',
    headers:{'content-type':'multipart/form-data; boundary='+bound}
}
const req = http.request(options, (res) => {

    let data = '';
    res.on('data', (chunk) => {
        console.log('http.request: data: body =', data += chunk.toString('utf8'));
    });
    res.on('end', () => { console.log('http.request:  end: body =', data);  });

});

req.on('error', (e) =>  { console.log('http.request: error:', e.message);});
req.end(body);
```

```
let http = require('http');
let fs   = require('fs');

let bound = 'smw60-smw60-smw60';
let body  =  `--${bound}\r\n`;
    body +=  'Content-Disposition:form-data; name="file"; filename="MyFile.txt"\r\n';
    body +=  'Content-Type:text/plain\r\n\r\n';
    body +=  fs.readFileSync('D:\\MyFile.txt');                //  данные из файла
    body +=  `\r\n--${bound}--\r\n`
```

8. **HTTP-клиент:** простейший клиент, upload, отправка частями

```
let http = require('http');
let fs   = require('fs');

let bound = 'smw60-smw60-smw60';
let body  =  `--${bound}\r\n`;
    body +=  'Content-Disposition:form-data; name="file"; filename="MyFile.bmp"\r\n';
    body +=  'Content-Type:application/octet-stream\r\n\r\n';

let  options = {
       host:'localhost',
       path: '/mypath',
       port: 3000,
       method:'POST',
       headers:{'content-type':'multipart/form-data; boundary='+bound}
}
let req = http.request(options, (res) => {

    let data = '';
    res.on('data', (chunk) => {data += chunk;});
    res.on('end', () => { console.log('http.response: end: length body =', Buffer.byteLength(data)); });

});
req.on('error', (e) =>  { console.log('http.request: error:', e.message);});
req.write(body);    // отправляем 1 часть

let stream = new fs.ReadStream('D:\\xxx.bmp');
stream.on('data', (chunk)=>{req.write(chunk); console.log( Buffer.byteLength(chunk))});  // отправляем 2ю часть порциями
stream.on('end',       ()=>{req.end( `\r\n--${bound}--\r\n`);});                         // отправляем 3ю часть
```

9. **HTTP-клиент:** простейший клиент, download file

```
const http = require('http');
const fs = require('fs');

const file = fs.createWriteStream("file.bmp");

let  options = {
    host:'localhost',
    path: '/bmp/MyFile.bmp',
    port: 3000,
    method:'GET'
}

const req = http.request(options, (res) => { res.pipe(file); });
 req.on('error', (e) =>  {  console.log('http.request: error:', e.message);});
 req.end();
```

10. **HTTP-клиент:** https://nodejs.org/api/http.html#http_class_http_clientrequest
11. **HTTP-клиент:** https://www.twilio.com/blog/2017/08/http-requests-in-node-js.html
12. **HTTP-клиент:** http://qaru.site/questions/13161/how-to-make-an-http-post-request-in-nodejs
13. **HTTP-клиент:** https://nodejs.dev/making-http-requests-with-nodejs
14. **HTTP-клиент:** https://attacomsian.com/blog/http-requests-in-nodejs