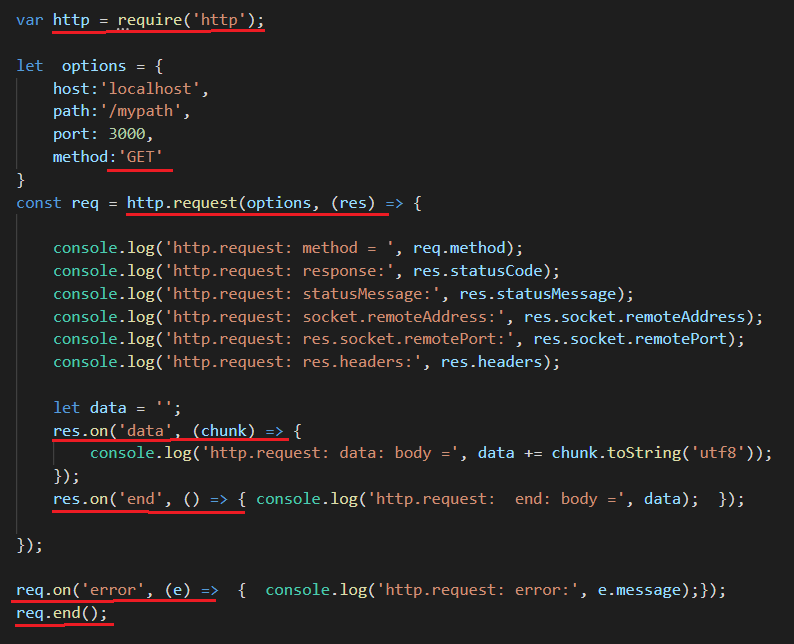
Лекция 07

ПСКП

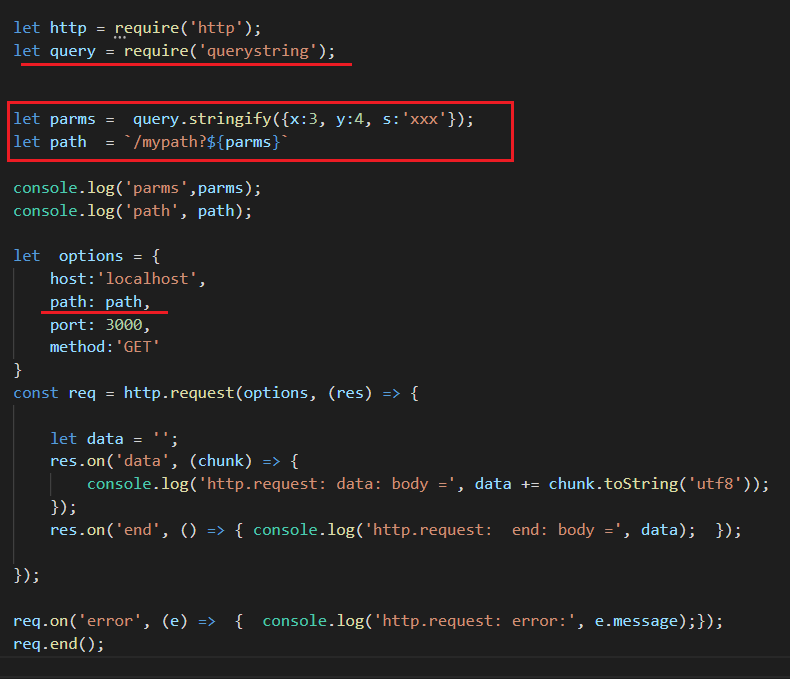
ПОИТ-3

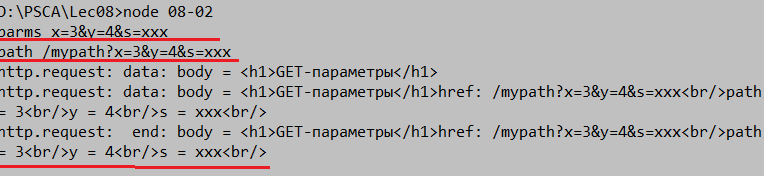
**Node.js-HTTP-клиент**

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

****

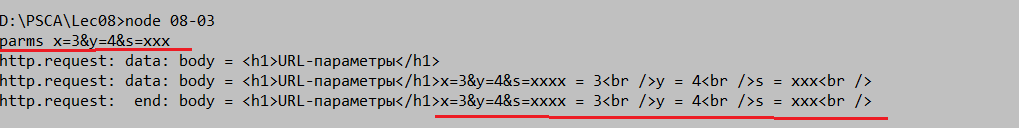
1. **HTTP-клиент:** простейший клиент, GET, параметры

****

****

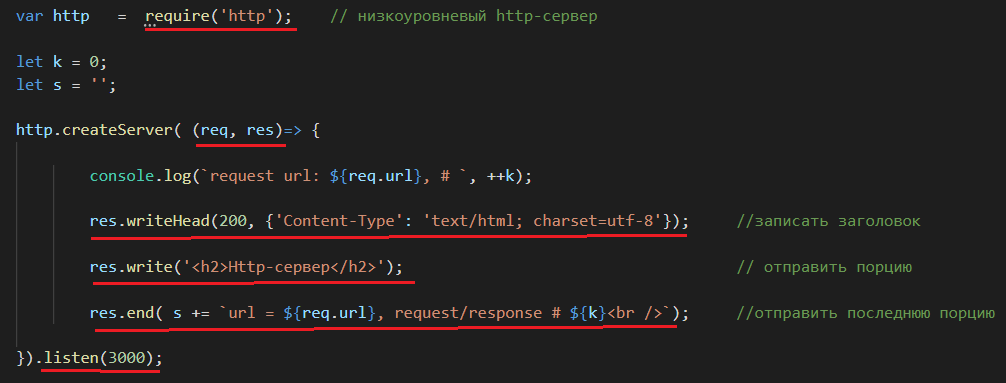
1. **HTTP-клиент:** простейший клиент, POST, параметры

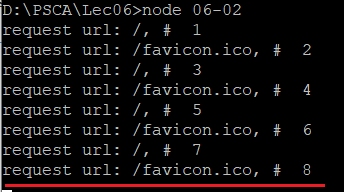
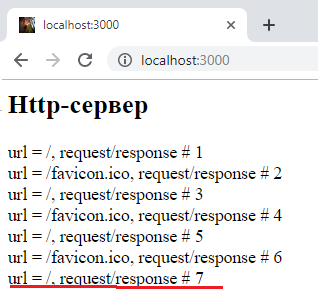
****

****

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

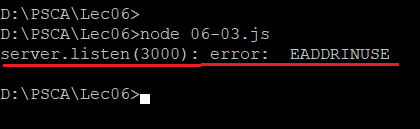
1. **HTTP-клиент:** <https://nodejs.org/api/http.html#http_class_http_clientrequest>
2. **HTTP-клиент:** <https://www.twilio.com/blog/2017/08/http-requests-in-node-js.html>
3. **HTTP-клиент:** <http://qaru.site/questions/13161/how-to-make-an-http-post-request-in-nodejs>
4. **HTTP-клиент:** <https://nodejs.dev/making-http-requests-with-nodejs>
5. **HTTP-клиент:** <https://attacomsian.com/blog/http-requests-in-nodejs>
6. **HTTP-клиент:**
7. **HTTP-клиент:**
8. **HTTP-клиент:**
9. **HTTP-клиент:**

****

****

1. **HTTP-сервер:** простейший сервер

****

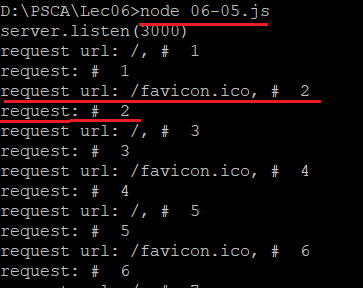
****

1. **HTTP-сервер:** простейший сервер

****

1. **HTTP-сервер:** простейший сервер

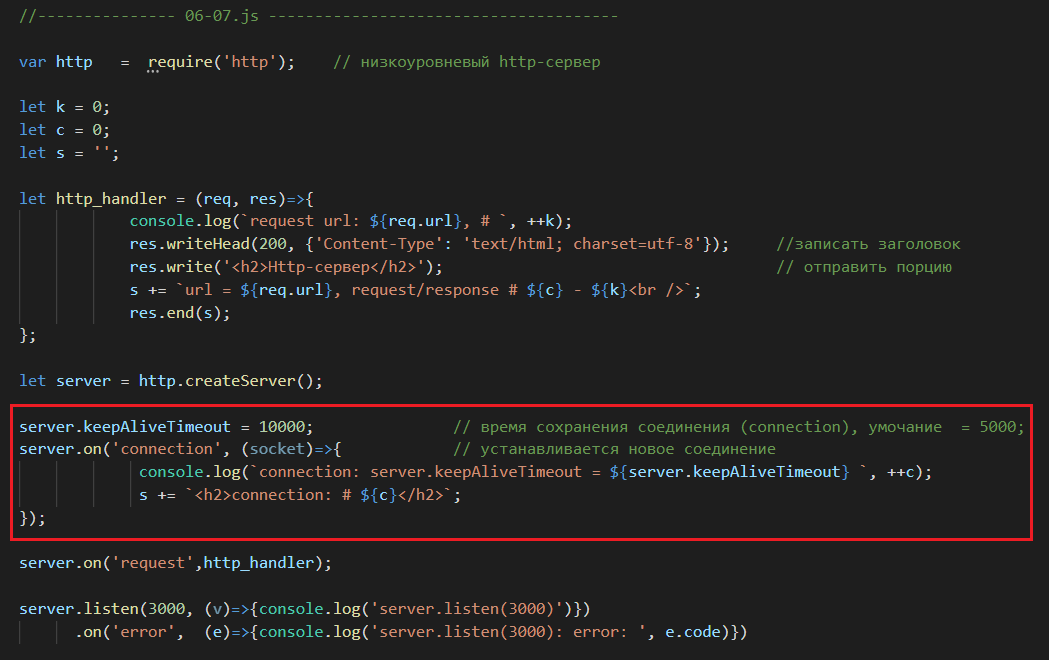
****

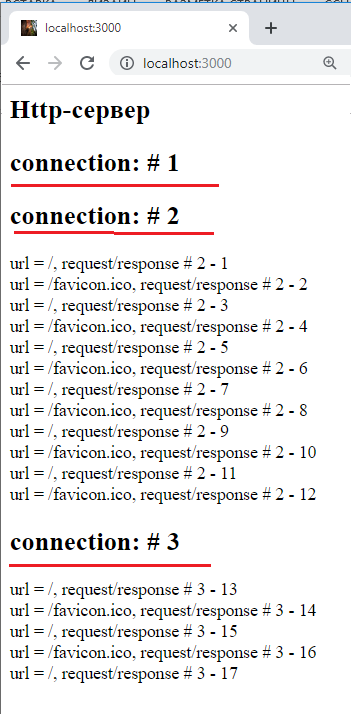
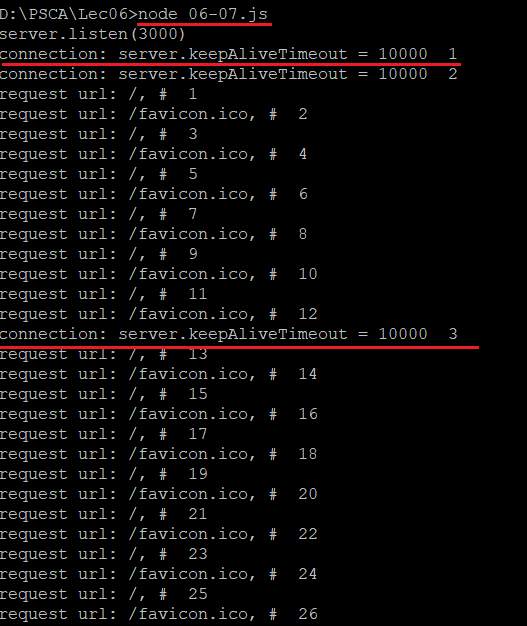
****

1. **HTTP-сервер:** простейший сервер

****

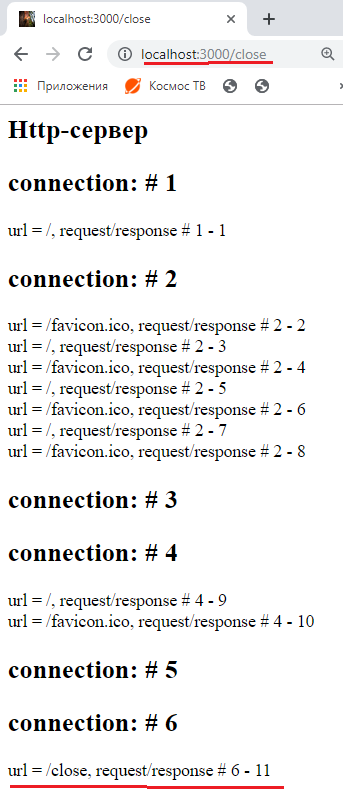
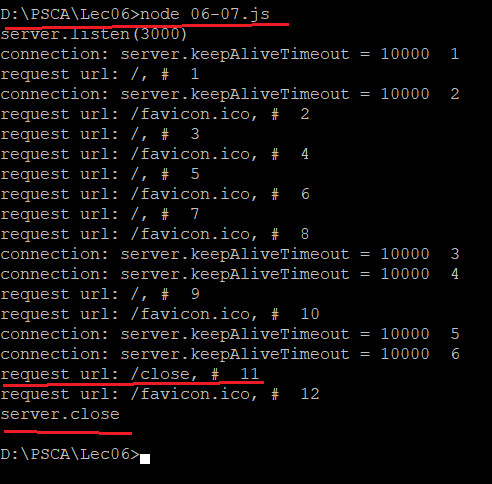
1. **HTTP-сервер:** простейший сервер **server.on(‘conection’)**

****

****

1. **HTTP-сервер:** простейший сервер **server.close()**

****

** **

1. **HTTP-сервер:** простейший сервер **server.on(‘close’)**

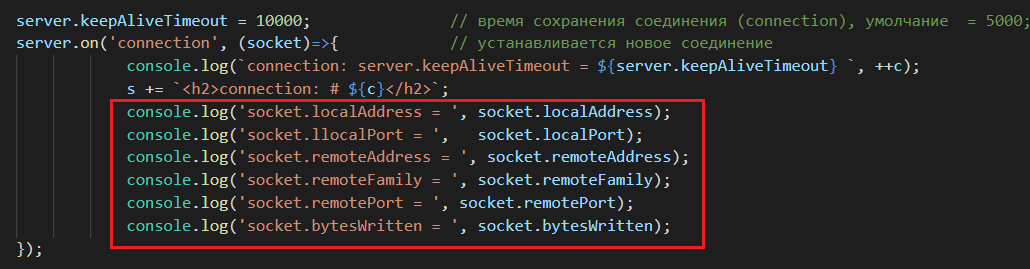
****

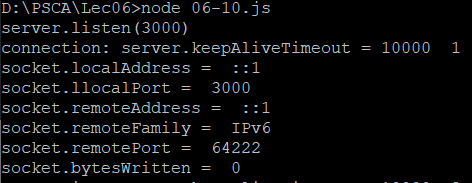
****

1. **HTTP-сервер:** простейший сервер **server.on(‘timeout’)**

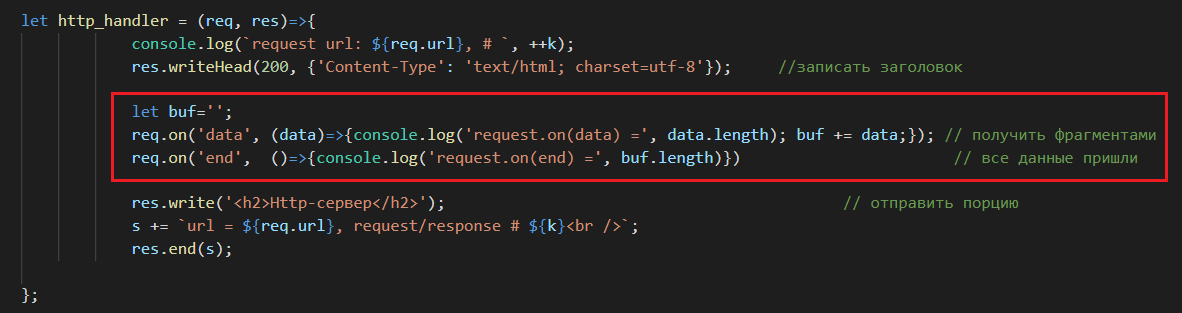
****

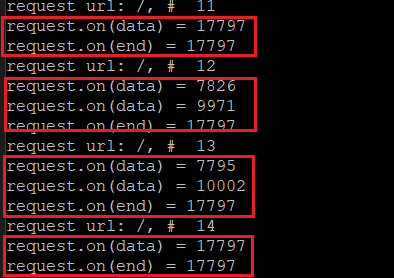
1. **HTTP-сервер:** простейший сервер, **socket properties**

****

****

1. **HTTP-сервер:** простейший сервер **request.on**

****

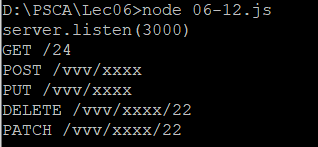
****

1. **HTTP-сервер:** простейший сервер, типичный цикл работы

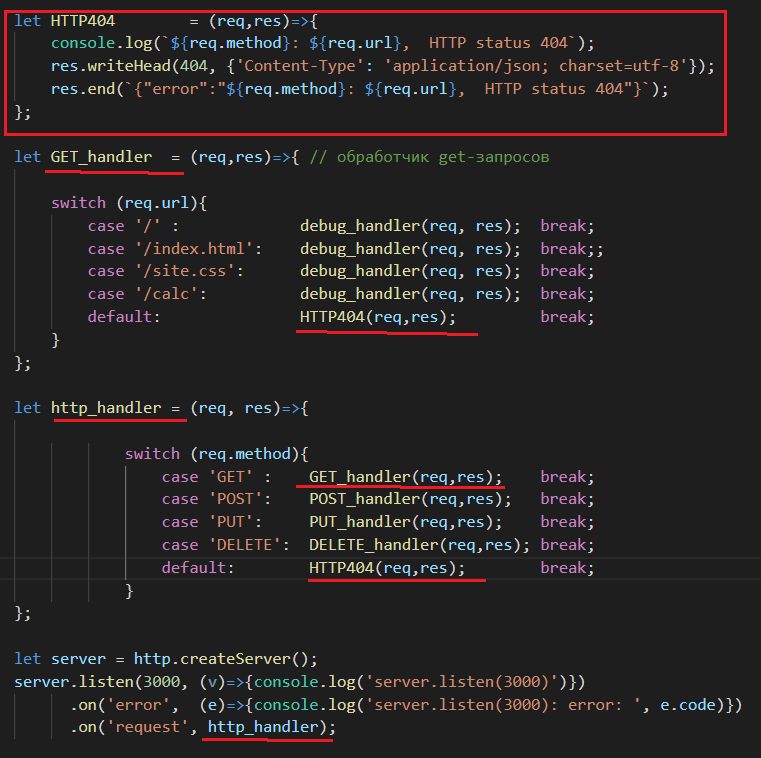


1. **HTTP-сервер:** простейший сервер, **method**

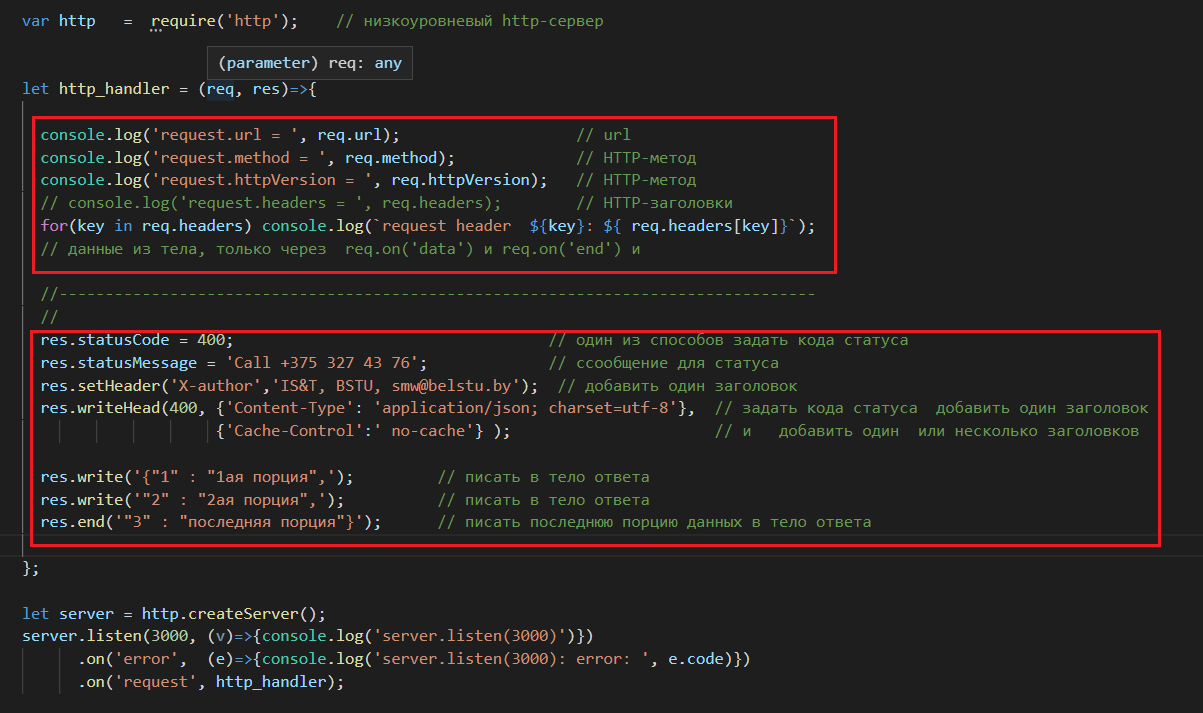
****

****

1. **HTTP-сервер:** простейший сервер, url, 404

****

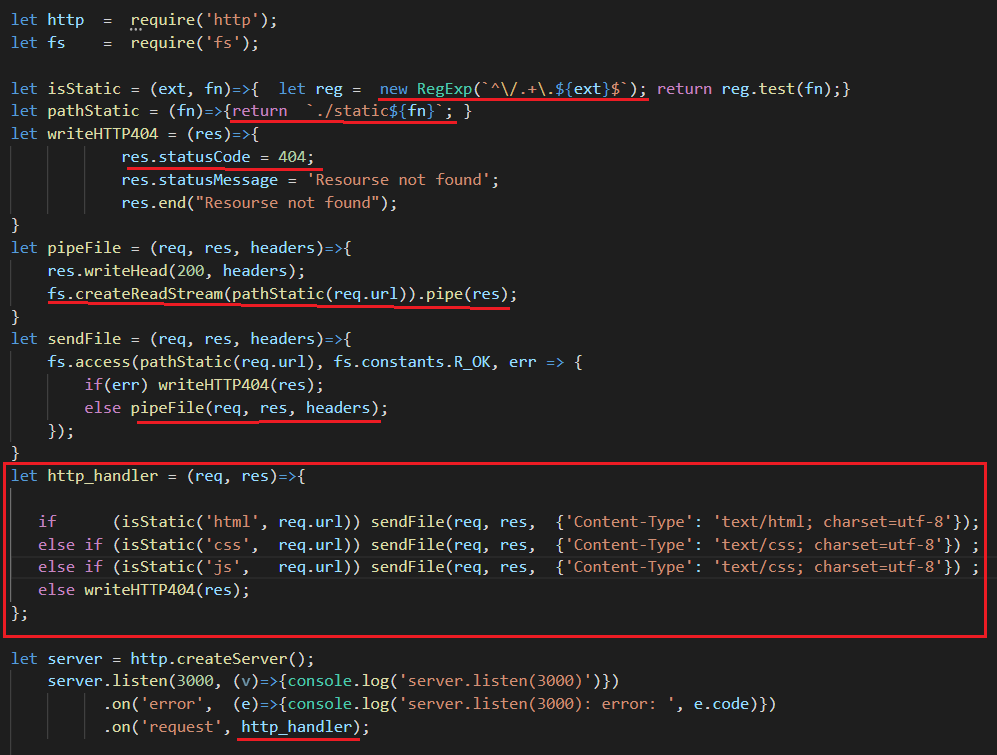
1. **HTTP-сервер:** простейший сервер, request, response

****

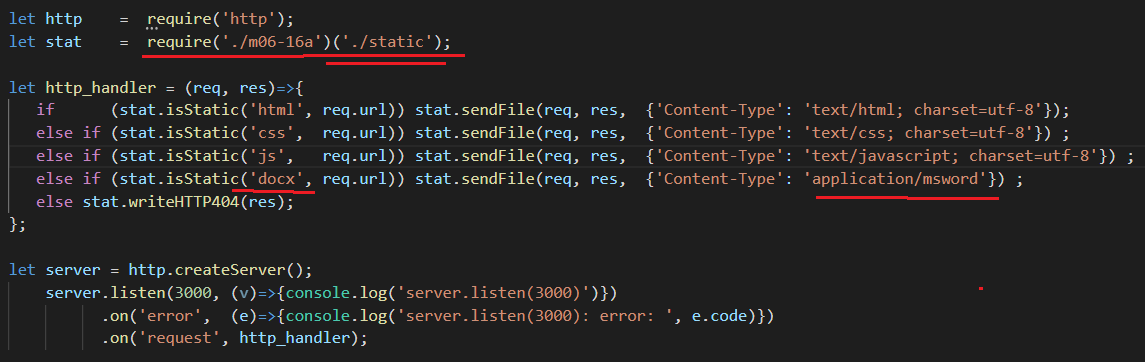
1. **HTTP-сервер:** простейший сервер, статические ресурсы

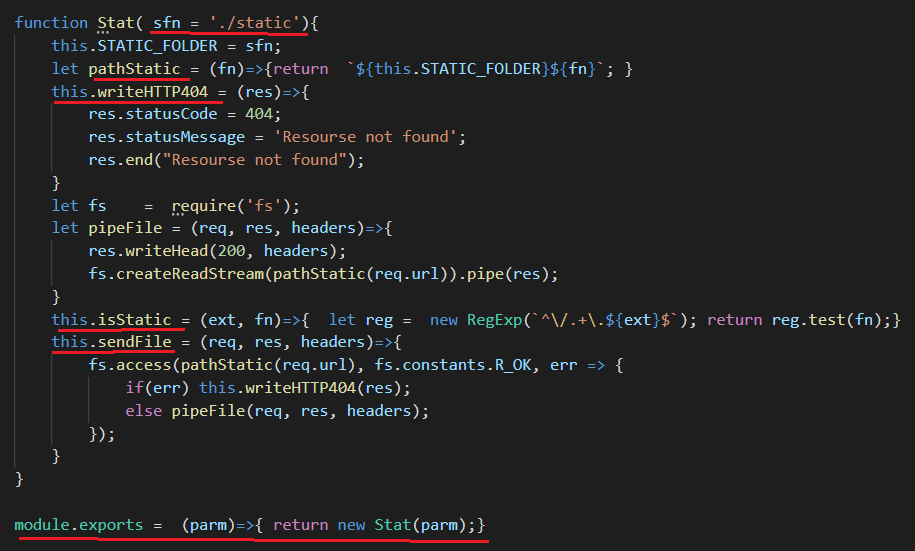


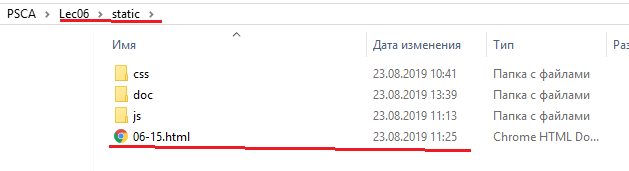
1. **HTTP-сервер:** простейший сервер, статические ресурсы

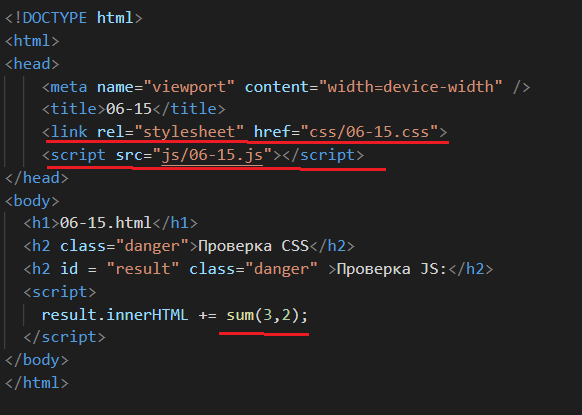
****

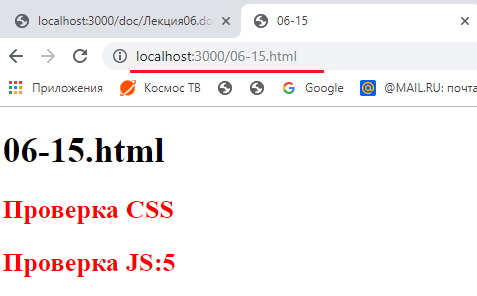
1. **HTTP-сервер:** простейший сервер, статические ресурсы, параметризируемый модуль.

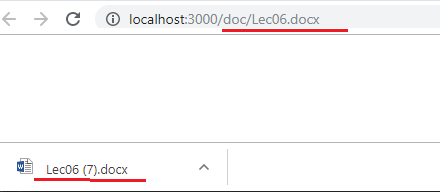
****

****

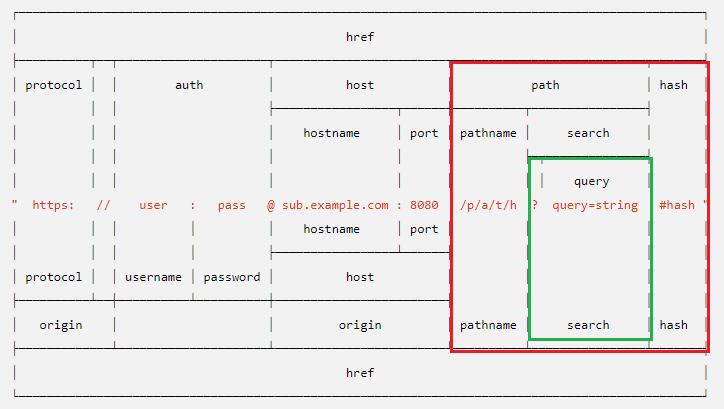
****

****

****

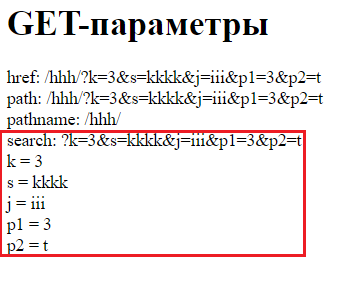
****

1. **HTTP-сервер:** простейший сервер, обработка параметров GET-запроса.

****

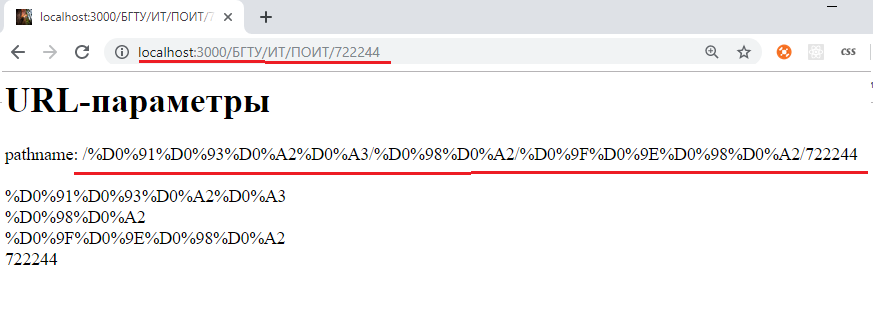
1. **HTTP-сервер:** простейший сервер, query-параметры,GET-запросы.

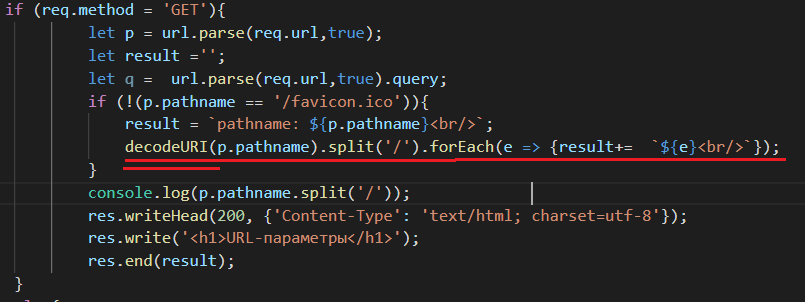
****

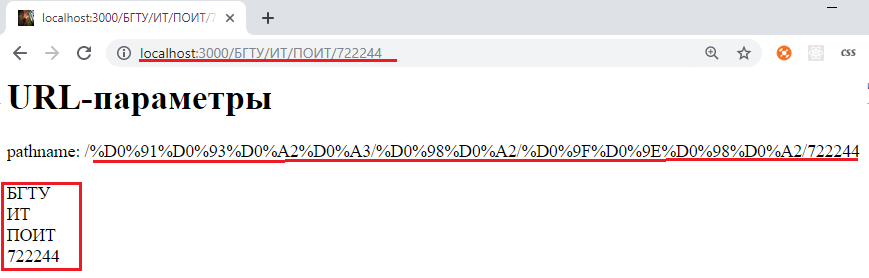
****

1. **HTTP-сервер:** простейший сервер, url-параметры,GET-запросы.









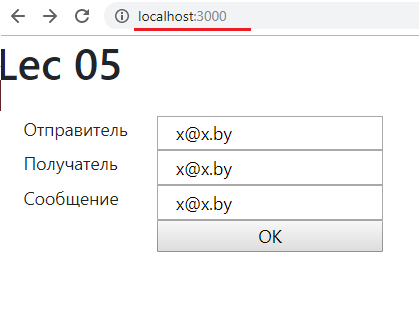
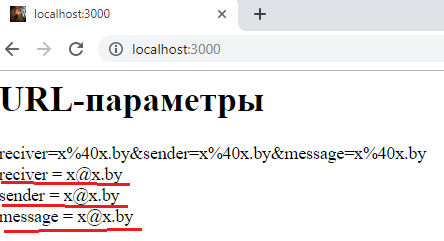
1. **HTTP-сервер:** обработка GET-параметров

|  |  |
| --- | --- |
| Модуль | Функции, классы |
| url | Parse/format, classes: URL, URLSearchParams |
| querystring | parse/stribgify, escape/unescspe |

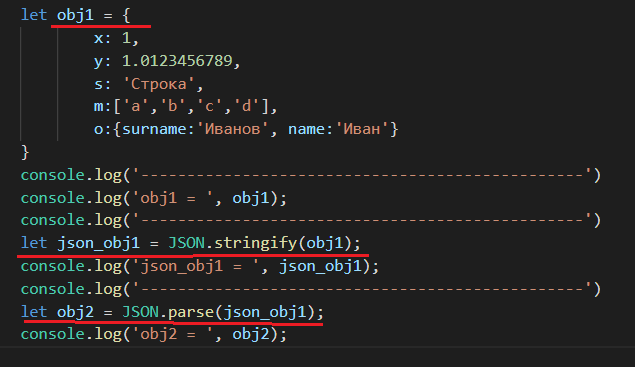
1. **HTTP-сервер:** простейший сервер, POST-параметры

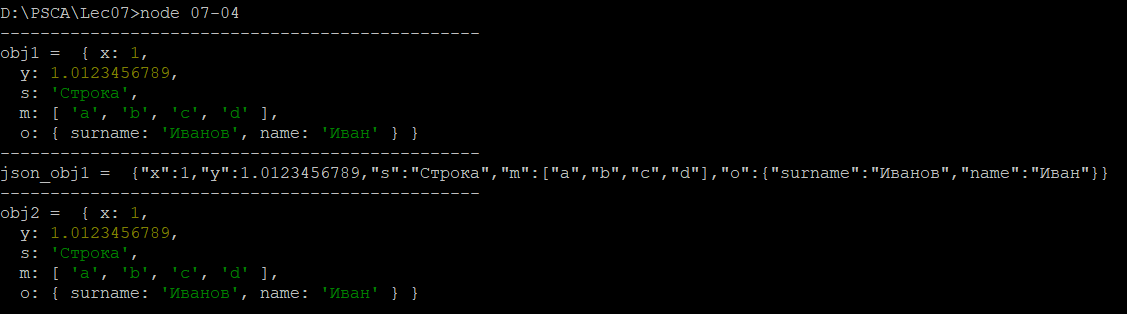
****

****

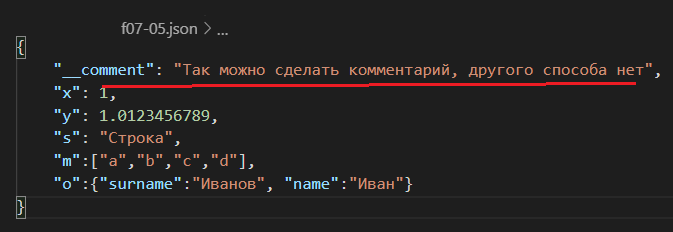
****

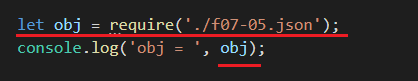
1. **HTTP-сервер:** простейший сервер, JSON-формат
2. **JSON:** JavaScript Object Notation, текстовый формат передачи данных, автор: Дуглас Крокфорд,   
   

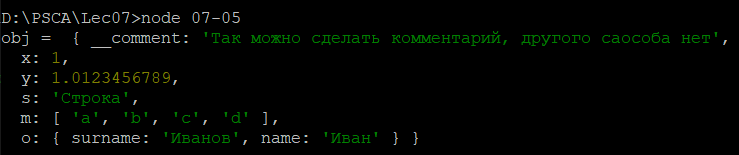
****

****

1. **JSON: require**:часто применяется для конфигурационных файлов

****

****

****

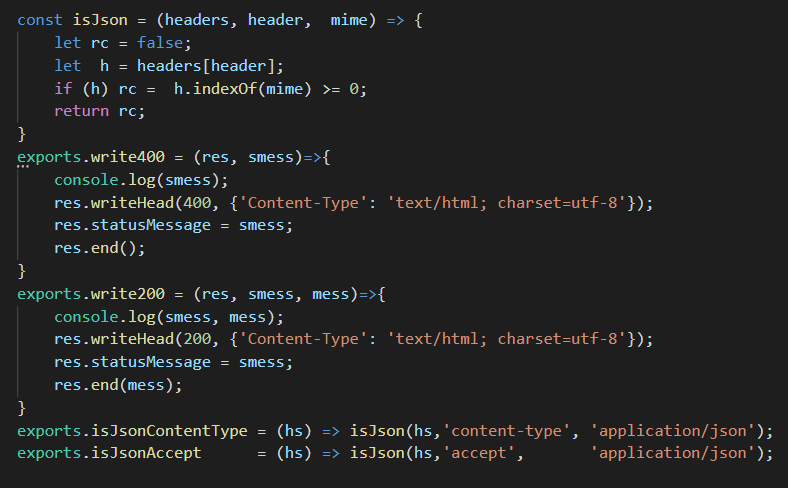
1. **JSON: MIME:**Multipurpose Internet Mail Extensions**,**

***application/json*** (RFC 4627), Content-Type, Accept

****

1. **HTTP-сервер:** простейший сервер, JSON-формат

****

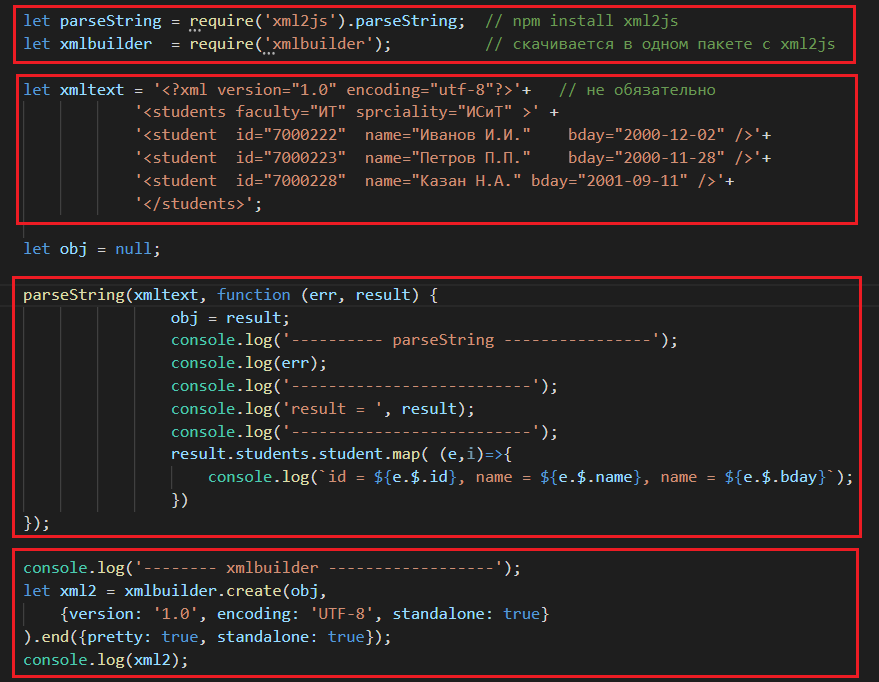
****

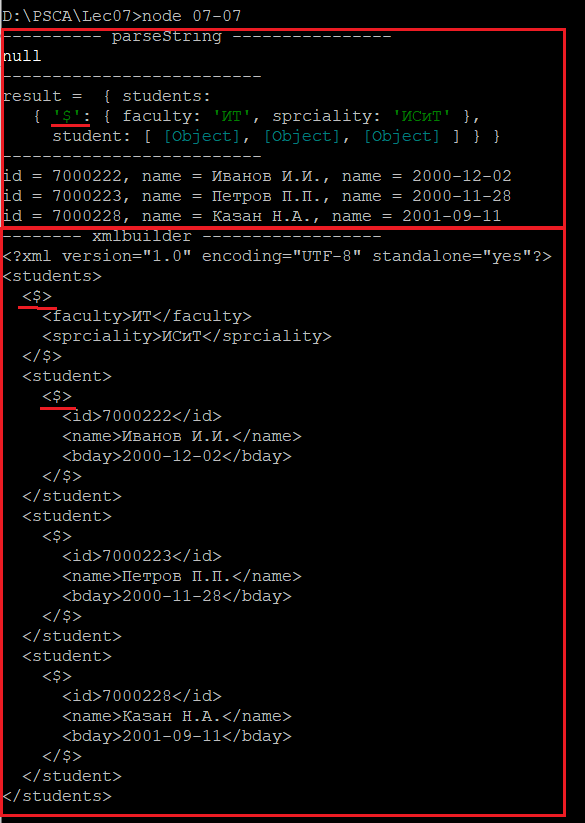
1. **HTTP-сервер:** простейший сервер, XML-формат, MIME:

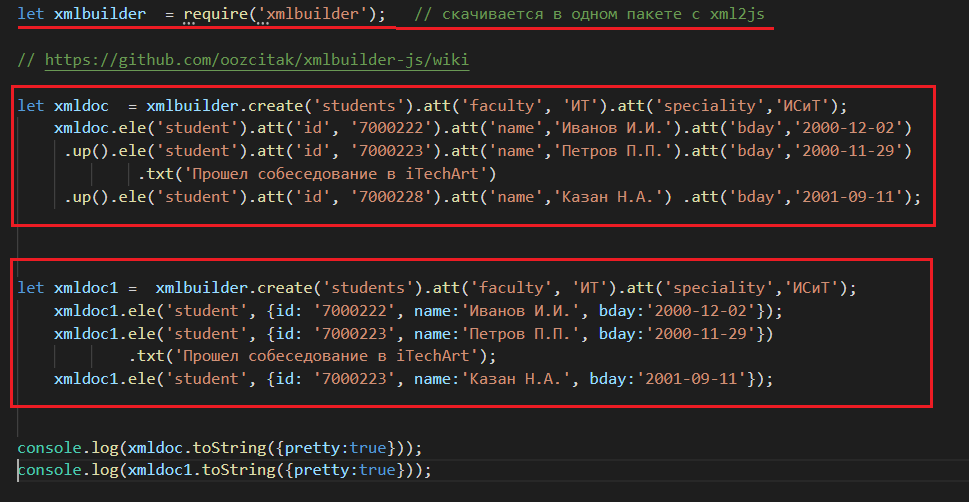
application/xml,

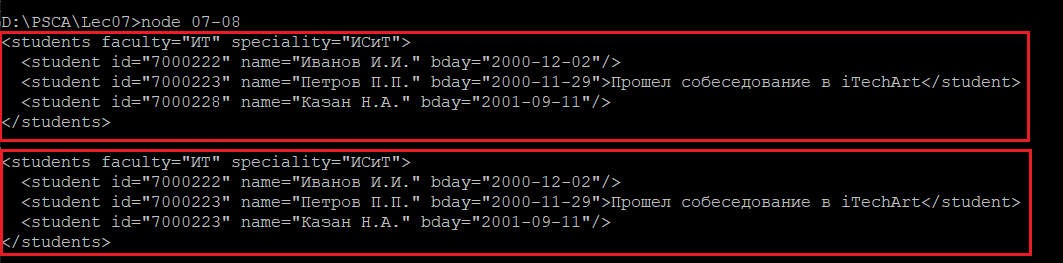
text/xml

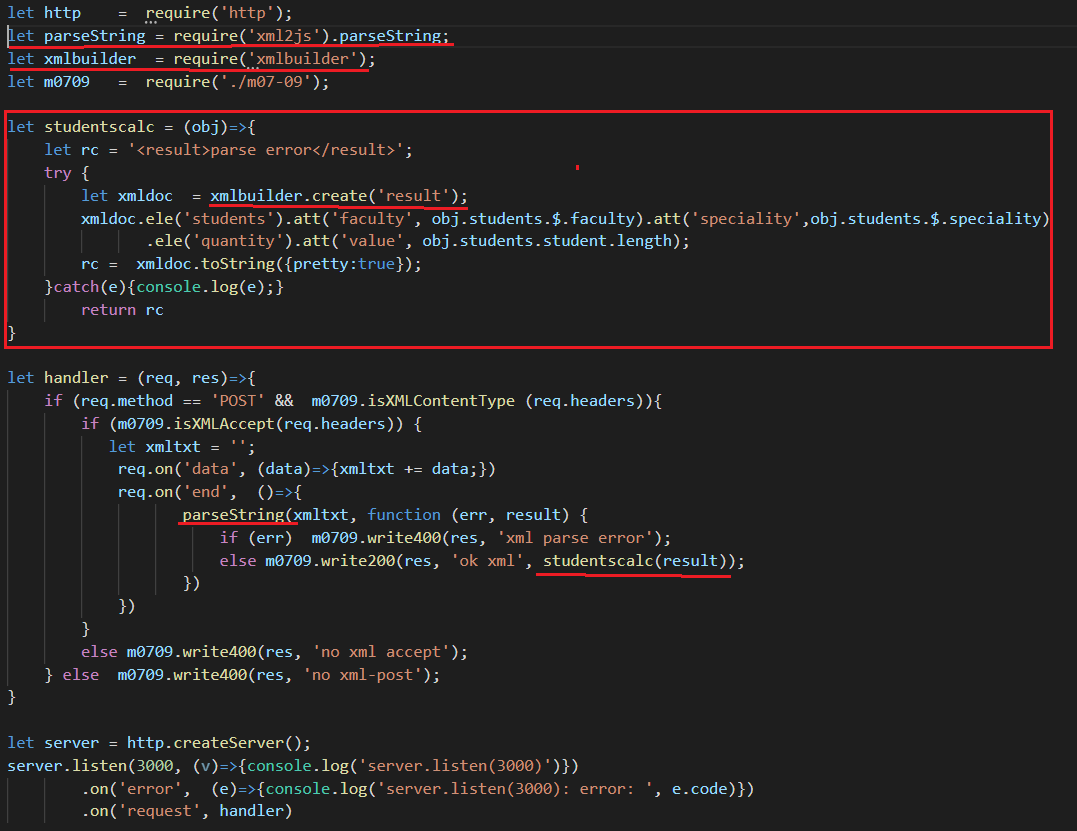
****

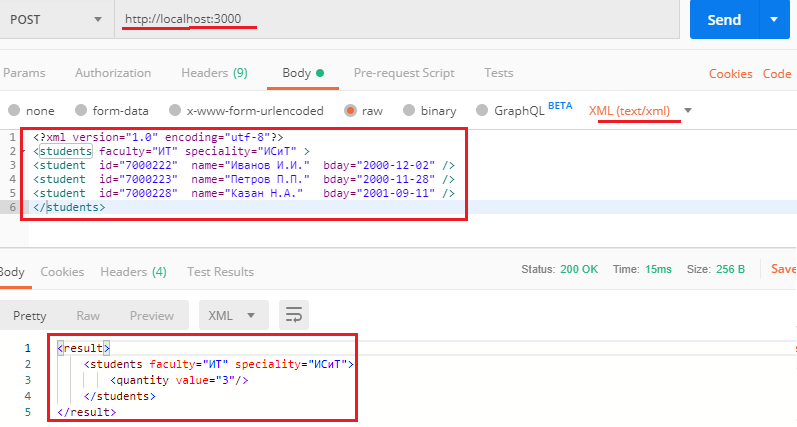
****

****

****

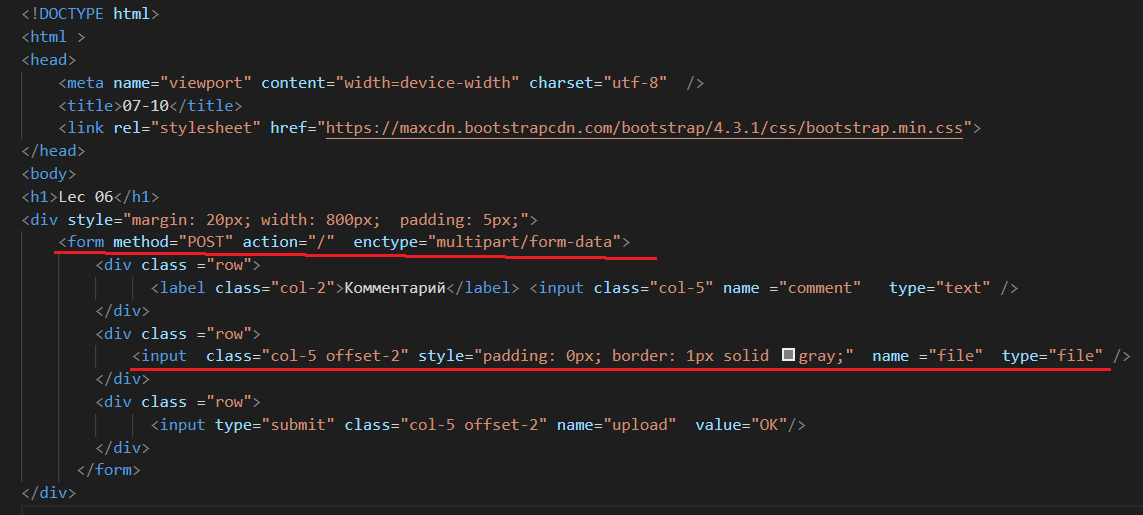
****

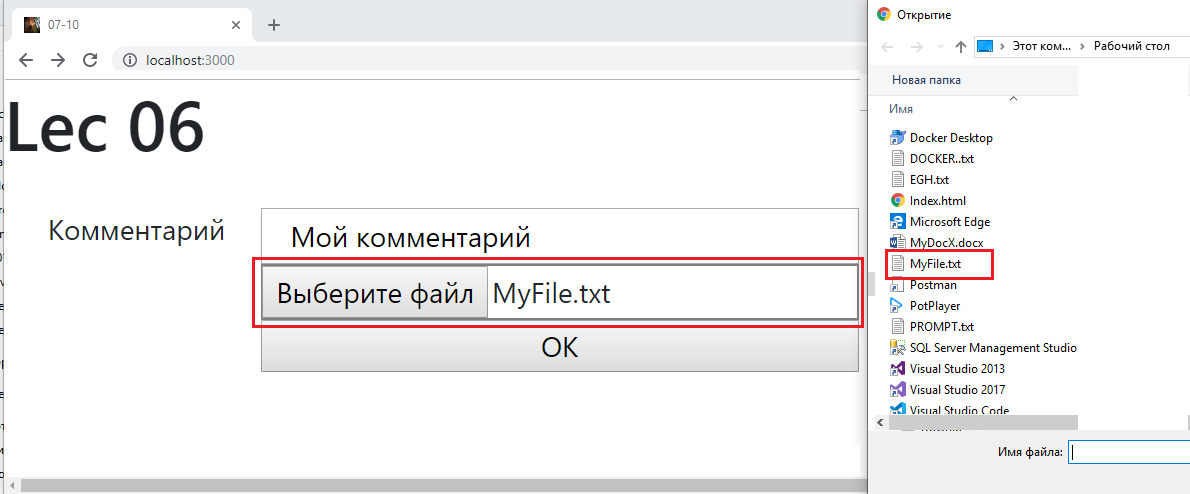
****

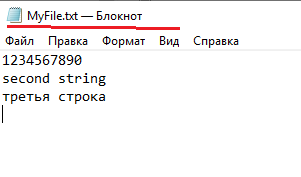
****

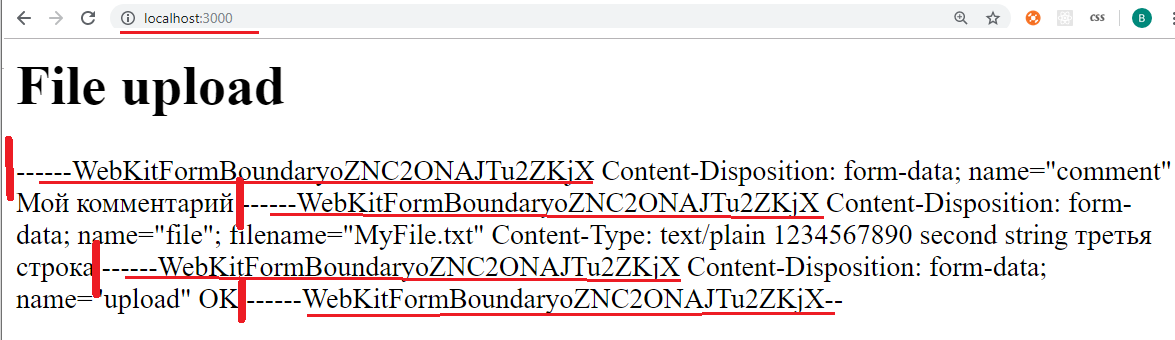
1. **HTTP-сервер:** простейший сервер, file upload

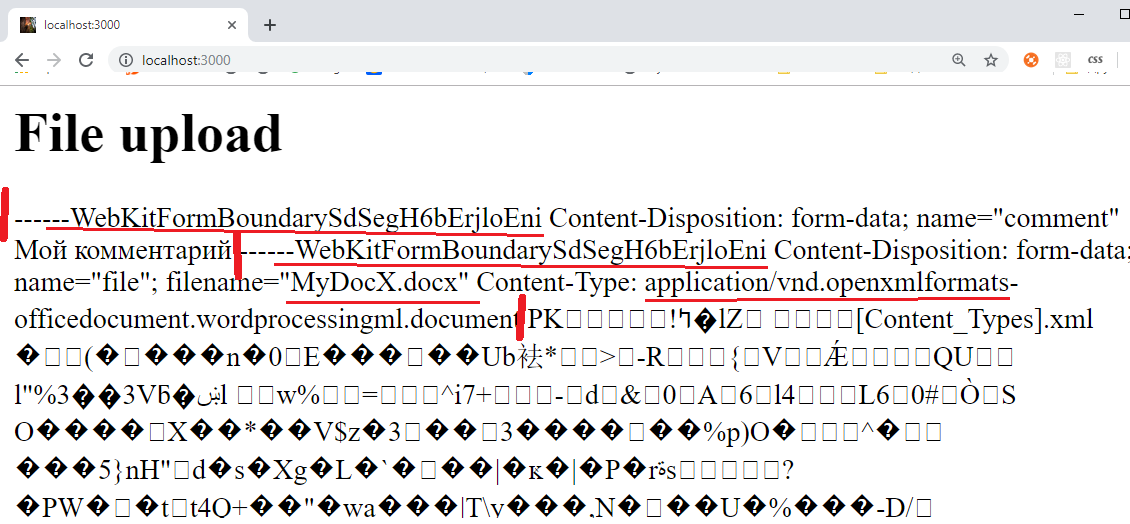




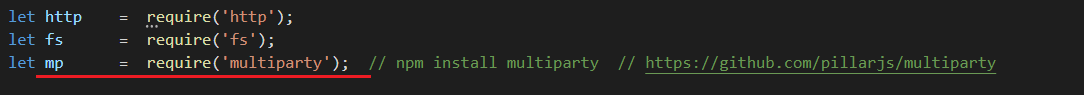


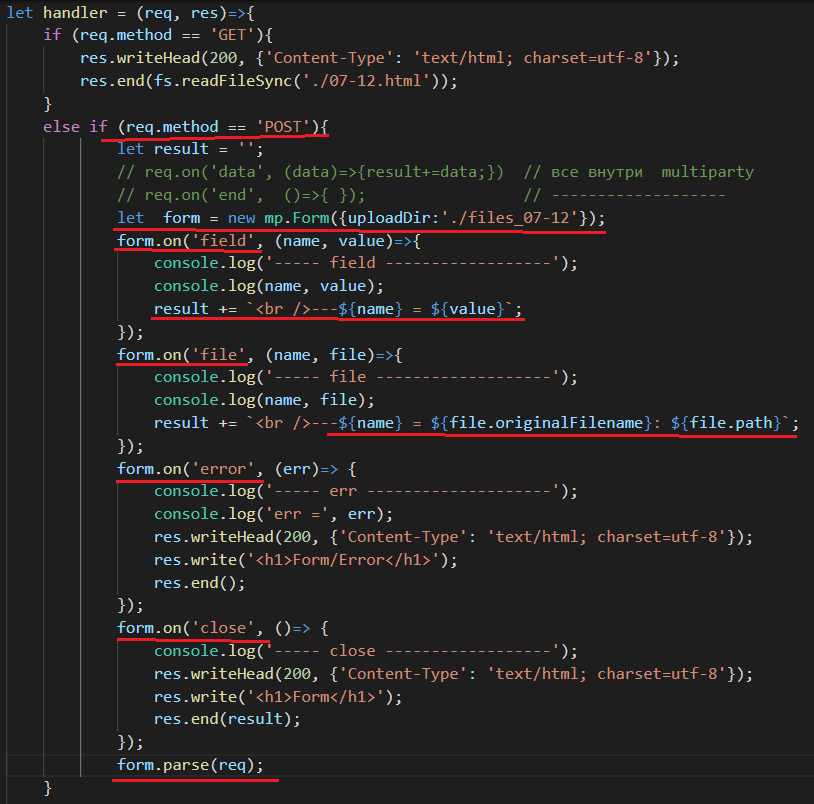


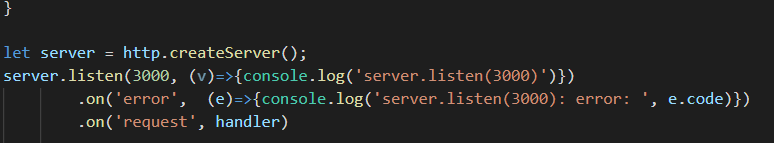


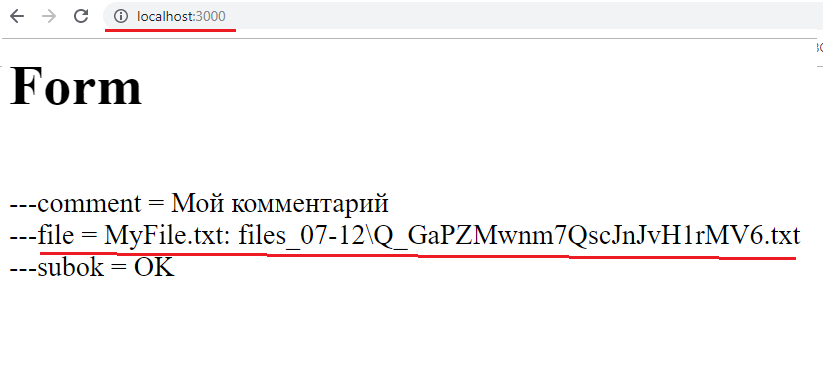


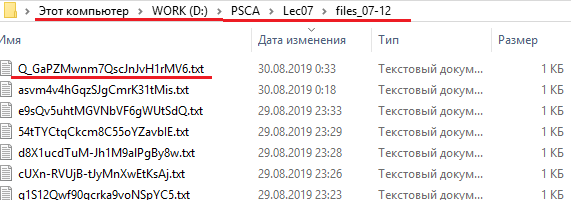
1. **HTTP-сервер:** простейший сервер, file upload, multiparty











http.STATUS\_CODES

{ '100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Moved Temporarily',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Time-out',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Request Entity Too Large',

'414': 'Request-URI Too Large',

'415': 'Unsupported Media Type',

'416': 'Requested Range Not Satisfiable',

'417': 'Expectation Failed',

'418': 'I\'m a teapot',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Unordered Collection',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Time-out',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required' }