

# Лабораторная работа 8: Работа с фреймворком Express

В этой работе мы создадим и будем работать самым популярным Nodejs-фреймворком Express

0. Откройте в браузере произвольную страницу в сети, зайдите в панель разработчика, выберите *Network* и обновите страницу - будут видны HTTP-заголовки

1. Откройте в консоли папку **08-lab-express** и инициализируйте её командой

```
npm init -y
```

2. Установите фреймворк **Express** и модули **pug**, **query-string**, **nodemon**:

```
npm i express pug query-string
```

```
npm i nodemon --save-dev
```

3. Убедитесь, что установка прошла успешно - посмотрите `npm list` или загляните в **08-lab-express/package.json**

4. Добавьте в **package.json** указанием на работу с ES6-модулями `"type": "module"` и запуск приложения через `npm start`:

```
{
  "name": "08-lab-express",
  "version": "1.0.0",
  "description": "",
  "main": "test.js",
  "type": "module",
  "scripts": {
    "start": "nodemon app.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2",
    "pug": "^3.0.2",
    "query-string": "^8.1.0"
  },
  "devDependencies": {
    "nodemon": "^2.0.20"
  }
}
```

5. Создайте файл **08-lab-express/app.js**:

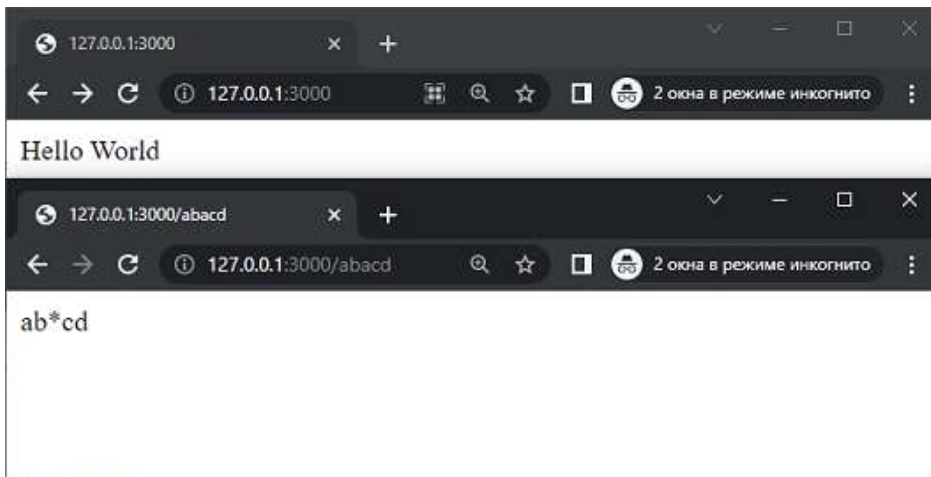
```
import express from 'express'
import queryString from 'query-string';
const app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

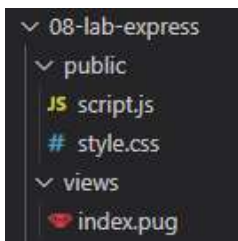
app.get('/ab*cd', function(req, res) {
  res.send('ab*cd');
});

app.listen(3000)
```

6. Запустите сервер `npm start` и откройте в браузере адреса `http://127.0.0.1:3000/` и `http://127.0.0.1:3000/abacd` :



7. Создайте папки **08-lab-express/public** и **08-lab-express/views**, поместите в неё файлы:



**index.pug:**

```
doctype html
head
  meta(charset='utf-8')
  meta(name='viewport' content='width=device-width')
  title App
  link(rel='stylesheet' href='style.css')
ul
  each item in items
    li= item
script(src='script.js')
```

**script.js:**

```
console.log('Проверка вывода')
```

## style.css:

```
* {  
  font-family: sans-serif;  
  box-sizing: border-box;  
}
```

8. В файле приложения **08-lab-express/app.js** укажите использование статических файлов:

```
app.use(express.static('public'));
```

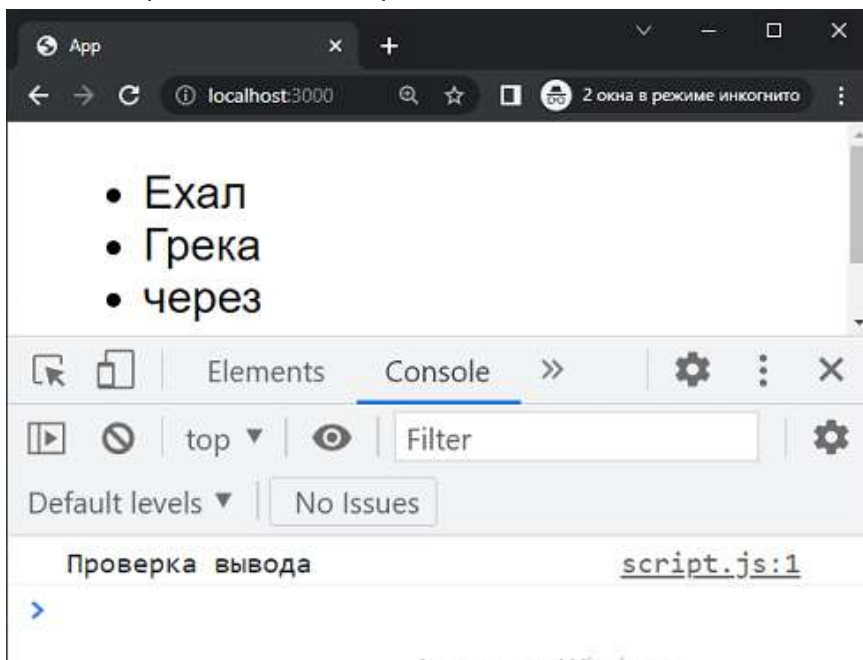
9. Настройте каталог для шаблонов и укажите в качестве движка шаблонов Pug:

```
app.set('views', './views')  
app.set('view engine', 'pug')
```

10. Перепишите обработчик для корневой страницы

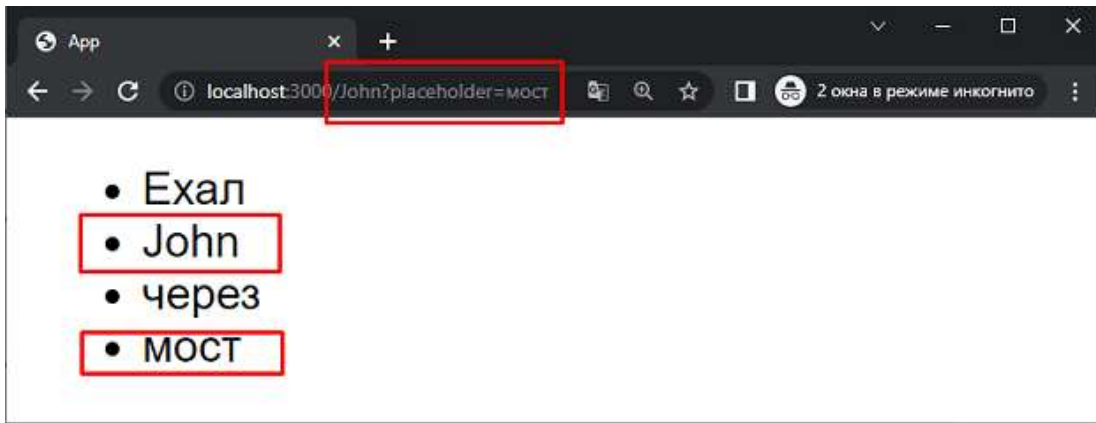
```
app.get('/', function (req, res) {  
  res.render('index', { items: ['Ехал', 'Грека', 'через', 'реку'] });  
});
```

11. Откройте/обновите в браузере страницу <http://localhost:3000/> : вы увидите использование стилей и вывод строки в консоли через JS



12. Получите параметр из маршрута и GET-параметр:

```
app.get('/:name', function (req, res) {  
  console.log( req.params )  
  console.log( req.query )  
  const firstName = req.params.name ?? 'Грека'  
  const placeholder = req.query.placeholder ?? 'реку'  
  res.render('index', { items: ['Ехал', firstName, 'через', placeholder] });  
});
```



13\*. Создайте `feedback.pug` с формой обратной связи из многострочного текстового поля и кнопки отправки формы методом **POST**. Опишите обработчики запросов для маршрута `/feedback` методами GET и POST. Для метода GET должна показываться форма из `feedback.pug`. Для метода POST должна показываться надпись "Спасибо за обращение". При этом, при методе POST через 10 секунд должно происходить перенаправление на маршрут `/feedback` методом GET. Информация из формы должна записываться в файл **messages.json**. *Примечание: можно создать обычную форму, а потом найти в поисковике по запросу 'html to pug' конвертер в формат pug*

14\*. Разберитесь самостоятельно с модулем **express-session** (<https://www.npmjs.com/package/express-session>). Создайте приложение, в котором можно регистрироваться, логиниться и разлогиниваться пользователям. Данные учётных записей пользователей можно хранить в JSON-файле. *Примечание: для этого приложения лучше создать отдельную папку*

15\*. Установите, настройте и запустите **json-server** (<https://www.npmjs.com/package/json-server>). Создайте новое Express-приложение, которое будет выполнять GET, POST, PUT и DELETE запросы к **json-server**.