

# Чек-лист “Как стать Back-end Разработчиком”

## 1. неВведение

### 1.1 А оно вам вообще надо?

Бэкенд-разработка предлагает множество возможностей для карьерного роста и хорошего заработка. Однако, она требует постоянного обучения и адаптации к новым технологиям. Работа может быть сложной и монотонной, но в то же время предоставляет возможность решать интересные задачи и работать в команде с другими профессионалами.

<https://online.lemon.school/blog/stoit-li-stanovitsya-programmistom-v-2023-godu-plyusy-i-minusy-professii>

<https://youtu.be/6SkKPpF5bjo>

<https://youtu.be/8lCdObsYjP0>

### 1.2 Самый главный навык программиста - умение самообучаться

Умение самообучаться - ключевой навык для бэкенд-разработчика. Это включает использование поисковых систем, чтение документации, участие в сообществах разработчиков и использование инструментов, таких как ChatGPT, для получения помощи и рекомендаций.

<https://youtu.be/4T5eQQ4FO98>

<https://youtu.be/ruNMS07ru8c>

<https://vc.ru/education/416875-10-sovetov-o-tom-kak-pravilno-guglit>

<https://rb.ru/story/kak-pisat-zapros-v-chatgpt/>

## 2. Обучение

### 2.0 Полезные ресурсы

<https://roadmap.sh/backend>

<https://habr.com/ru/articles/593801/>

<https://blog.skillfactory.ru/kak-stat-behkend-razrabotchikom-s-nulya-poshagovoe-rukovodstvo-dlya-novichkov/>

<https://sky.pro/media/backend-razrabotchik-kto-eto-takoj-i-chem-on-zanimaetsya/>

<https://youtu.be/VR9EPKz8aXk>

## 2.1 Интернет

Изучение принципов работы интернета, включая протоколы (HTTP, HTTPS, TCP/IP), браузеры, DNS, домены и хостинг, является основой для понимания бэкенд-разработки.

<https://cctld.ru/help/wiki/kak-ustroen-internet/>

<https://habr.com/ru/companies/ruvds/articles/720704/>

<https://youtu.be/vobsAhMBPVw>

## 2.2 Фронт-енд (база)

Основы фронт-енда (HTML, CSS, JavaScript) необходимы для взаимодействия с клиентской стороной и понимания того, как данные передаются между клиентом и сервером.

<https://stepik.org/course/38218/promo>

<http://htmlbook.ru/>

<https://youtu.be/LXb1CLGtlV8>

## 2.3 OS и фундаментальные знания

Основы работы с операционными системами, особенно с Linux, важны для бэкенд-разработчика. Знание командной строки, файловой системы и процессов поможет вам в разработке и развертывании приложений.

<https://youtu.be/W1pF8nglj1U>

<https://youtu.be/wENbIPiO9mk>

<https://blog.skillfactory.ru/glossary/linux/>

<https://stepik.org/course/73/promo>

## 2.4 Выбор и изучение языка

Выберите язык программирования, который вам больше всего подходит (PHP, Python, Java, Go, C#, Ruby, Node.js) и изучите его основы, синтаксис и стандартные библиотеки.

<https://youtu.be/wZ4FyZ7LzKE>

<https://youtu.be/TVq6dBklXxc>

<https://stepik.org/>

## 2.5 Алгоритмы и структуры данных

Основные алгоритмы (сортировка, поиск) и структуры данных (массивы, списки, стеки, очереди, деревья, графы) являются фундаментальными знаниями для решения задач и оптимизации производительности.

<https://youtu.be/U0g-dbrVTV8>

<https://stepik.org/>  
<https://yandex.ru/yaintern/algorithm-training>  
[https://yandex.ru/yaintern/algorithm-training\\_1](https://yandex.ru/yaintern/algorithm-training_1)  
[https://yandex.ru/yaintern/algorithm-training\\_2](https://yandex.ru/yaintern/algorithm-training_2)

## 2.6 JS

Основы JavaScript необходимы для взаимодействия с фронт-ендом и понимания асинхронного обмена данными между клиентом и сервером.

<https://stepik.org/course/2223/promo>

## 2.7 Система контроля версий

Git - основной инструмент для контроля версий, позволяющий отслеживать изменения кода, ветвление и слияние. Он является стандартом в индустрии и необходим для работы в команде.

[https://youtu.be/zZBiln\\_2FhM](https://youtu.be/zZBiln_2FhM)  
[https://github.com/Gnurian/UDEMY\\_GIT](https://github.com/Gnurian/UDEMY_GIT)  
<https://git-scm.com/book/ru/v2>

## 2.8 Базы данных

Реляционные (MySQL, PostgreSQL) и NoSQL (MongoDB, Redis) базы данных используются для хранения и обработки информации. Знание принципов работы с ними является важным навыком бэкенд-разработчика.

<https://stepik.org/course/1240/promo>  
<https://stepik.org/course/63054/>

## 2.9 REST, API

Создание и использование RESTful API позволяет обмениваться данными между различными системами и сервисами. Это важный аспект взаимодействия с фронт-ендом и другими бэкенд-сервисами.

<https://youtu.be/BRPismGYR6o>  
<https://habr.com/ru/articles/38730/>

## 2.10 Библиотеки и Фреймворки

Фреймворки упрощают разработку, предоставляя готовые решения для распространенных задач. Изучение популярных фреймворков для выбранного языка (например, Django для Python) ускорит ваше обучение и повысит вашу продуктивность.

PHP:

- Базовые библиотеки: PDO, OpenSSL, XMLReader, JSON, SPL
- Библиотеки для работы с базами данных: MySQLi, PDO, Memcached, Redis
- Библиотеки для работы с веб-сервером: cURL, DOM, Gd, ZIP
- Библиотеки для работы с сетью: sockets, imap, ftp, soap
- Библиотеки для работы с файлами: Fileinfo, FileInfo, ZIP
- Библиотеки для работы с XML: SimpleXML, XMLReader, XMLWriter

#### Python:

- Базовые библиотеки: os, sys, datetime, sqlite3, re, urllib, requests, json, argparse, logging
- Библиотеки для работы с веб-сервером: Flask, Django, Pyramid, Tornado, FastAPI
- Библиотеки для работы с базами данных: SQLAlchemy, PyMySQL
- Библиотеки для работы с сетью: socket, asyncore, asyncssh, paramiko
- Библиотеки для работы с алгоритмами: NumPy, Pandas, SciPy, Scikit-learn
- Библиотеки для работы с графикой: PIL, matplotlib, pygame

#### Java:

- Базовые библиотеки: java.util, [java.io](#), [java.net](#), java.lang, java.sql, java.time, java.regex
- Библиотеки для работы с веб-сервером: Spring, Hibernate, Struts, JSF, GWT
- Библиотеки для работы с базами данных: JDBC, JPA, Hibernate, MyBatis
- Библиотеки для работы с сетью: Java Sockets, Apache HttpComponents, Netty
- Библиотеки для работы с алгоритмами: Apache Commons Math, Weka
- Библиотеки для работы с графикой: JavaFX

#### Go:

- Базовые библиотеки: fmt, os, net, time, log, flag
- Библиотеки для работы с веб-сервером: Gorilla Mux, Gin, Echo, Revel

- Библиотеки для работы с базами данных: PostgreSQL, MySQL, MongoDB
- Библиотеки для работы с сетью: net/http, net/smtp
- Библиотеки для работы с алгоритмами: gonum, golearn
- Библиотеки для работы с графикой: go-chart

#### C#:

- Базовые библиотеки: System, System.Collections, System.IO, System.Net, System.Text, System.Xml
- Библиотеки для работы с веб-сервером: ASP.NET, MVC, WebForms
- Библиотеки для работы с базами данных: ADO.NET, EF, LINQ
- Библиотеки для работы с сетью: sockets, HttpWebRequest, WebClient
- Библиотеки для работы с алгоритмами: Math.NET, AForge.NET
- Библиотеки для работы с графикой: System.Drawing

#### Ruby:

- Базовые библиотеки: time, json, yaml, fileutils, uri, net, socket
- Библиотеки для работы с веб-сервером: Ruby on Rails, Sinatra, Hanami
- Библиотеки для работы с базами данных: ActiveRecord, Mongoid
- Библиотеки для работы с сетью: net/http, net/scp, snmp
- Библиотеки для работы с алгоритмами: SciRuby
- Библиотеки для работы с графикой: RMagick, Cairo

#### Node.js:

- Базовые библиотеки: http, path, fs, os, https, util
- Библиотеки для работы с веб-сервером: Express, Koa, Hapi, Meteor
- Библиотеки для работы с базами данных: MongoDB, Mongoose, MySQL, PostgreSQL
- Библиотеки для работы с сетью: net, dgram, tls, request
- Библиотеки для работы с алгоритмами: Math.js, TensorFlow.js

- Библиотеки для работы с графикой: Chart.js, D3.js, Canvas

### 2.11 Регулярные выражения

Регулярные выражения - это мощный инструмент для работы с текстом, который позволяет искать, заменять и манипулировать строками на основе заданных шаблонов. Back-end разработчики могут использовать регулярные выражения для валидации входных данных, поиска и извлечения информации из текста, а также для реализации сложных операций со строками. Знание регулярных выражений поможет разработчикам эффективно работать с текстовыми данными и улучшить качество кода.

<https://stepik.org/course/107335/syllabus>

<https://youtu.be/xaAWyLC5ORM>

### 2.12 Кэширование

Кэширование - это процесс хранения результатов запросов или вычислений для быстрого доступа при повторном использовании. В контексте back-end разработки, кэширование может использоваться для ускорения загрузки страниц, снижения нагрузки на сервер и базу данных. Разработчики могут использовать различные стратегии кэширования, такие как кэширование на стороне сервера, кэширование базы данных или кэширование с использованием сервисов, таких как Redis или Memcached.

<https://youtu.be/ZPPKUEpxAVE>

### 2.13 Аудит безопасности

Безопасность является критически важным аспектом разработки веб-приложений. Back-end разработчики должны знать основы авторизации, аутентификации и идентификации. Это включает в себя создание безопасных систем хранения паролей, использование токенов для доступа к защищенным ресурсам и применение принципа наименьших привилегий для ограничения доступа к данным и функциональности.

<https://www.kaspersky.ru/blog/identification-authentication-authorization-difference/29123/>

<https://youtu.be/SQPdxnFnpZY>

<https://habr.com/ru/articles/720842/>

### 2.14 Тестирование

Тестирование - это процесс проверки корректности работы программного обеспечения. Важными видами тестирования для back-end разработчиков

являются unit-тестирование и интеграционное тестирование. Unit-тесты проверяют отдельные функции или компоненты кода, а интеграционные тесты проверяют взаимодействие между различными частями системы.

Тестирование помогает обнаружить и исправить ошибки, а также улучшить качество кода.

<https://habr.com/ru/articles/169381/>

<https://www.zaptest.com/ru/бэкенд-тестирование-глубокое-погруж>

## 2.15 CI/CD

Непрерывная интеграция (CI) и непрерывная доставка (CD) - это практики, которые позволяют автоматизировать процесс сборки, тестирования и развертывания приложений. CI/CD упрощает разработку, обеспечивая быстрое обнаружение и исправление ошибок, а также сокращая время между написанием кода и его развертыванием на продакшене. Разработчики должны знать, как настроить и использовать инструменты CI/CD, такие как Jenkins, GitLab CI или GitHub Actions.

<https://habr.com/ru/companies/southbridge/articles/649027/>

<https://youtu.be/ANj7qUgzNq4>

## 2.16 Принципы проектирования и разработки

Принципы проектирования и разработки, такие как SOLID, DRY, KISS и YAGNI, помогают создавать чистый, модульный и легко поддерживаемый код. SOLID - это набор принципов объектно-ориентированного программирования, DRY (Don't Repeat Yourself) означает избегание дублирования кода, KISS (Keep It Simple, Stupid) подразумевает простоту и ясность кода, а YAGNI (You Aren't Gonna Need It) предлагает избегать добавления функциональности, которая может быть не нужна.

<https://habr.com/ru/articles/589489/>

<https://otus.ru/nest/post/2503/>

## 2.17 Архитектурные паттерны

Архитектурные паттерны - это проверенные временем решения для общих проблем проектирования систем. Они помогают создавать масштабируемые, гибкие и поддерживаемые приложения. Некоторые популярные архитектурные паттерны включают MVC (Model-View-Controller), MVVM (Model-View-ViewModel) и CQRS (Command Query Responsibility Segregation). Знание этих паттернов поможет разработчикам принимать обоснованные решения при проектировании системы.

<https://habr.com/ru/companies/alconost/articles/522662/>

<https://tproger.ru/translations/top-5-arhitekturnyh-patternov-dlja-raspredelejnyh-sistem/>

### 2.18 Message brokers

Message brokers, такие как RabbitMQ и Kafka, обеспечивают надежную и асинхронную передачу сообщений между компонентами системы. Они могут использоваться для распределения задач, обработки событий и обмена данными между микросервисами. Знание работы с message brokers поможет разработчикам создавать масштабируемые и отказоустойчивые приложения.

<https://habr.com/ru/companies/sberbank/articles/669456/>

[https://ru.wikipedia.org/wiki/Брокер\\_сообщений](https://ru.wikipedia.org/wiki/Брокер_сообщений)

### 2.19 Docker

Docker - это платформа для разработки, доставки и запуска приложений в контейнерах. Контейнеры позволяют упаковать приложение со всеми его зависимостями, что облегчает развертывание и управление приложениями. Знание работы с Docker поможет разработчикам создавать и поддерживать однородные среды разработки и продакшена.

<https://stepik.org/course/123300/promo>

<https://youtu.be/KS80Knz-1Z4>

### 2.20 GraphQL

GraphQL - это язык запросов и среда выполнения для API, который предоставляет гибкий и эффективный способ получения данных с сервера. В отличие от REST, GraphQL позволяет клиентам запрашивать только те данные, которые им нужны, и объединять запросы для разных ресурсов. Знание работы с GraphQL может быть полезным для создания современных и производительных API.

<https://habr.com/ru/articles/326986/>

### 2.21 Websockets

Websockets - это протокол, который обеспечивает двунаправленное взаимодействие между клиентом и сервером через постоянное соединение. Он позволяет серверу отправлять данные клиенту без необходимости постоянных запросов со стороны клиента. Знание работы с веб-сокетами поможет разработчикам создавать приложения с непосредственным обменом



данными в реальном времени, такие как чаты или онлайн-игры.

<https://ru.hexlet.io/blog/posts/chto-takoe-websocket-i-kak-oni-voobsche-rabotayut>

## 2.22 Веб серверы

Веб-серверы, такие как Apache и Nginx, являются основой для размещения и обслуживания веб-приложений. Back-end разработчики должны знать, как настраивать и оптимизировать веб-серверы для обеспечения безопасности, производительности и надежности приложений. Это включает в себя управление конфигурацией, настройку SSL/TLS, балансировку нагрузки и мониторинг производительности.

<https://youtu.be/uFWyAn6E4BU>

## 3. Резюме

Резюме Back-end разработчика должно содержать информацию о вашем опыте работы, навыках и достижениях в области разработки веб-приложений. Начните с краткого описания вашей профессиональной квалификации и опыта работы, указав, сколько лет вы уже работаете в этой области.

Затем перечислите языки программирования, фреймворки, технологии и инструменты, которыми вы владеете и которые использовали в своих проектах. Опишите свой опыт работы с базами данных, системами управления контентом и другими технологиями, которые вы использовали в своей работе. Укажите свои достижения, такие как создание и оптимизация высоконагруженных систем, разработка архитектуры приложений и опыт работы с DevOps. Также укажите свои навыки в управлении проектами, коммуникации и работе в команде.

Важно, чтобы ваше резюме было лаконичным и содержало только самую важную информацию. Используйте ключевые слова, которые соответствуют вакансии, на которую вы подаете заявку. Это поможет привлечь внимание работодателя и увеличить шансы на получение желаемой должности. Не забывайте также указывать контактную информацию для связи с вами.

<https://youtu.be/6XK2OvWHSsg>

<https://tproger.ru/experts/do-programmers-need-cv/>

## 4. Итог

Back-end разработка - это важная и сложная область, которая требует глубоких знаний и навыков. Чтобы стать успешным back-end разработчиком, необходимо освоить множество технологий и понимать, как они

взаимодействуют друг с другом. Кроме того, разработчики должны быть готовы к постоянному обучению и развитию, так как индустрия программного обеспечения постоянно меняется и предлагает новые инструменты и подходы. Важно также развивать навыки коммуникации и управления временем, чтобы успешно работать в команде и справляться с множеством задач и проектов.