

ОБРАЗОВАТЕЛЬНЫЙ ОНЛАЙН-КУРС

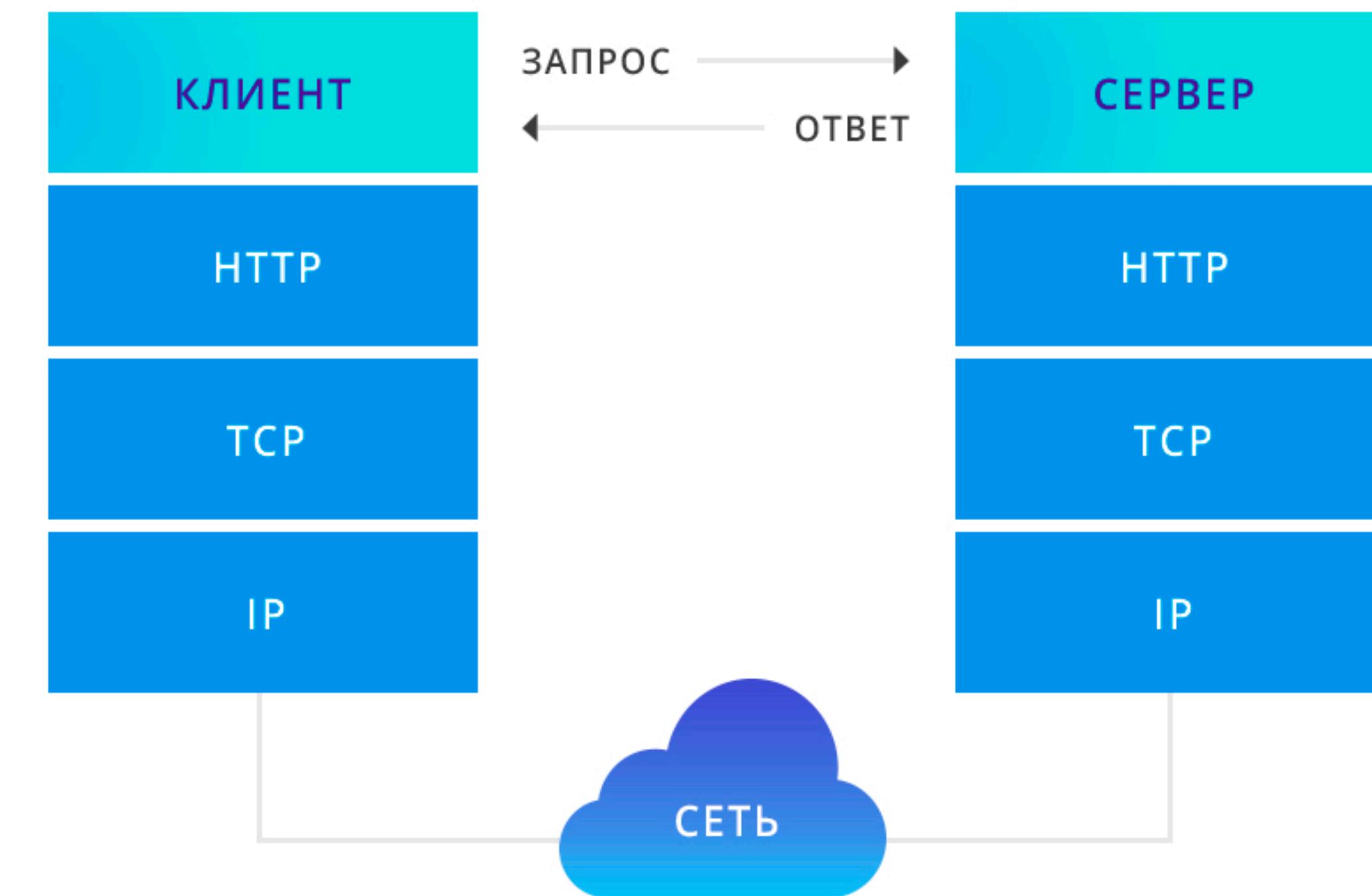
PHP-разработчик с нуля до PRO

КУРС 1. МОДУЛЬ 2.

■ Протокол HTTP/HTTPS —

Протокол передачи данных HTTP

HTTP (англ. *HyperText Transfer Protocol* – протокол передачи гипертекста) –



Структура протокола HTTP

1. **Стартовая строка** (англ. *Starting line*) – определяет тип сообщения
2. **Заголовки** (англ. *Headers*) – характеризуют тело сообщения, параметры передачи и прочие сведения
3. **Тело сообщения** (англ. *Message Body*) – непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой

The diagram illustrates the structure of an HTTP request message. It is divided into three main sections, each marked with a number:

- 1**: The first section contains the start line: `POST /security/config HTTP/1.1`.
- 2**: The second section contains the headers:

```
Host: www.defence.gov
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Referer: https://www.qsoft.ru/hack-pentagon/
Cookie: PHPSESSID=m16ugmdp26r2539ujssucreil0
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 83
```
- 3**: The third section contains the message body, which is highlighted with a red border and contains the parameter `form_id=Security-ReserveEmailHandlerAuthForm&token=8455d5c67b5ab77049c201044a77be0d`.

Стартовая строка

Стартовая строка составляется по схеме: **Метод URI HTTP/Версия**

Метод (англ. *Method*) – определяет операцию, которую нужно осуществить с указанным ресурсом

▶ POST/security/config HTTP/1.1

URL (*Uniform Resource Identifier*, унифицированный идентификатор ресурса) – путь до ресурса на web-сервере

▶ POST/example/index.php?param=value HTTP/1.1

Версия – используемая версия протокола, совпадает с версией указанной в запросе

▶ POST/security/config HTTP/1.1

Методы протокола

- Безопасные
- Идемпотентные
- Кэшируемые



Список HTTP методов

- **GET** – используется для запроса содержимого указанного ресурса
- **POST** – применяется для передачи пользовательских данных заданному ресурсу
- **OPTIONS, HEAD, PUT, DELETE** и т.д.



Заголовки HTTP

1. **Основные заголовки** (англ. *General Headers*) – должны включаться в любое сообщение клиента и сервера
2. **Заголовки запроса** (англ. *Request Headers*) – используются только в запросах клиента
3. **Заголовки ответа** (англ. *Response Headers*) – только для ответов от сервера
4. **Заголовки сущности** (англ. *Entity Headers*) – сопровождают каждую сущность сообщения

Основные заголовки HTTP-сообщений

Заголовок	Группа	Назначение	Пример
Accept	Заголовок запроса	Предназначен для информирования сервера о типах данных, которые поддерживаются клиентом	Accept: text/html, text/plain, image/jpeg
Content-type	Заголовок сущности	Предназначен для идентификации типа передаваемых данных	Content-Type: text/html; charset=utf-8
Content-length	Заголовок сущности	Содержит строку, в которой записана длина передаваемых данных в байтах при использовании метода передачи POST	Content-Length: 1348
Cookie	Заголовок запроса	В этом заголовке хранятся все Cookies	Cookie: version=1; firstVisit=false;
Set-Cookie	Заголовок ответа	Используется для установки Cookie	Set-Cookie: SM_GUEST_ID=146450; expires=Sat, 05-Jan-2019 18:40:26 GMT; path=/; domain=example.com
Location	Заголовок ответа	Полный или относительный URI по которому клиенту следует перейти	Location: http://example.com/

Тело сообщения

ТЕЛО ЗАПРОСА

Если для запроса используется метод POST, в теле сообщения содержатся любые данные отправляемые серверу, например данные введенные пользователем в HTML форму

ТЕЛО ОТВЕТА

Если запрос выполнен успешно, то тело ответа содержит данные в формате указанном в заголовке Content-type и кодировке указанной в заголовке Transfer-Encoding.

Обычно тело ответа содержит HTML-код запрошенного ресурса



Коды состояния

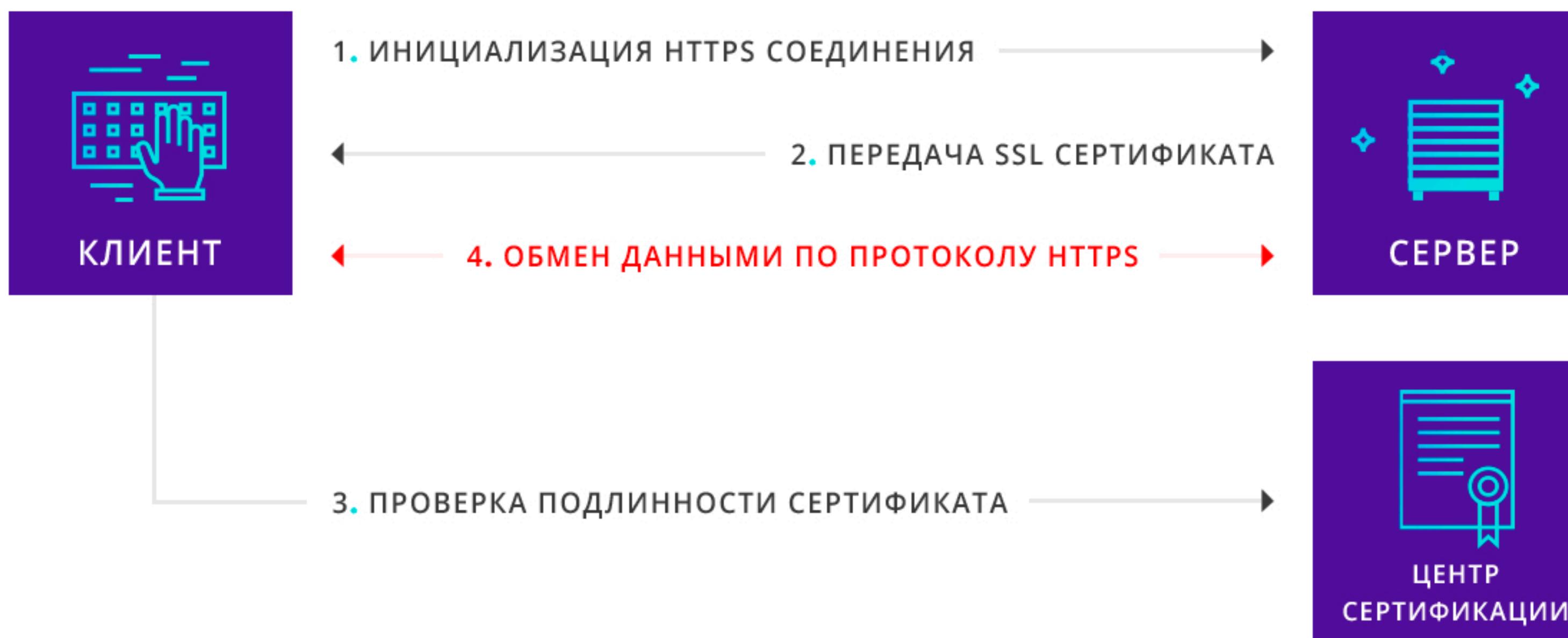
Код	Класс	Назначение
1xx	Информационный (англ. Informational)	Используются чтобы сообщить клиенту о процессе обработки запроса. Ответа от клиента на данные коды не требуется.
2xx	Успех (англ. Success)	В этом классе находятся коды успешного выполнения запроса.
3xx	Перенаправление (англ. Redirection)	Используются для уведомления клиента о том, что для завершения запроса необходимо выполнить другой запрос.
4xx	Ошибка клиента (англ. Client Error)	Отправляются клиенту, если сервер не смог обработать запрос из за ошибки со стороны клиента.
5xx	Ошибка сервера (англ. Server Error)	Информирование о случаях неудачного выполнения операции по вине сервера.

Часто используемые HTTP-коды

HTTP-код	Описание
200 OK	Сервер успешно обработал запрос. Если клиентом были запрошены какие-либо данные, то они находятся в заголовке и/или теле сообщения
301 Moved Permanently	Запрашиваемый ресурс на сервере отсутствует. Для переадресации клиента на новый URL отправляется HTTP-заголовок <i>Location</i> .
304 Not Modified	Сервер использует этот HTTP-код в соответствии с HTTP-заголовком <i>If-Modified-Since</i> или <i>If-None-Match</i> . Это означает, что запрашиваемый документ не модифицировался с указанной даты
400 Bad Request	В запросе клиента обнаружена синтаксическая ошибка
401 Unauthorized	Запрос требует аутентификации клиента. Для уточнения типа аутентификации и области запрашиваемого ресурса сервер может отправить HTTP-заголовок <i>WWW-Authenticate</i> . Клиент может повторить запрос, включив в заголовок сообщения поле <i>Authorization</i> с требуемыми для аутентификации данными
403 Forbidden	Доступ к запрашиваемому ресурсу запрещен
404 Not Found	Запрашиваемый документ отсутствует на сервере
500 Internal Server Error	Ошибка конфигурации сервера или внешней программы
502 Bad Gateway	Неверный ответ вышестоящего сервера или прокси-сервера
504 Gateway Time-Out	Шлюз или прокси-сервер временно недоступны

Безопасность протокола

HTTPS (англ. HyperText Transfer Protocol Secure – безопасный протокол передачи гипертекста). Защита данных в протоколе HTTPS обеспечивается с использованием криптографических протоколов **SSL** (англ. Secure Sockets Layer – уровень защищённых сокетов) или **TSL** (англ. Transport Layer Security – Протокол защиты транспортного уровня)



Сегодня мы узнали

- Структура протокола HTTP
- Методы протокола HTTP
- Заголовки протокола HTTP
- Что такое тело сообщения
- HTTP коды состояния
- Безопасный протокол HTTPS