Основы Интернет-технологий

Введение в Интернет-технологии

Лекция 1

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Предмет и задачи курса Основы Интернет-технологий

Предметом данного курса являются технологии глобальной сети World Wide Web

В рамках курса будут рассмотрены такие вопросы как:

- Структура и принципы Веб (базовые понятия, архитектура, стандарты и протоколы);
- Технологии сети Веб (языки разметки и программирования веб-страниц – HTML, CSS и JavaScript, PHP, инструменты разработки и управления веб-контента и приложений для Веб, средства интеграции веб-контента и приложений в Веб).

Сеть Веб представляет собой глобальное информационное пространство, основанное на физической инфраструктуре Интернета и протоколе передачи данных HTTP. Зачастую, говоря об Интернете, подразумевают именно сеть Веб.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

История развития Интернет

Хронология развития Интернета (с 1966 по 2000 г.)	
Год	**Событие**
1966	Эксперимент с коммутацией пакетов управления ARPA
1969	Первые работоспособные узлы сети ARPANET
1972	Изобретение распределенной электронной почты
1973	Первые компьютеры, подключенные к сети ARPANET за пределами США
1975	Сеть ARPANET передана в ведение управления связи министерства обороны США
1980	Начинаются эксперименты с TCP/IP
1981	Каждые 20 дней к сети добавляется новый хост
1983	Завершен переход на TCP/IP
1986	Создана магистраль NSFnet
1990	Сеть ARPANET прекратила существование
1991	Появление Gopher
1991	Изобретение Всемирной паутины. Выпущена система PGP. Появление Mosaic
1995	Приватизация магистрали Интернета
1996	Построена магистраль OC-3 (155 Мбит/с)
1998	Число зарегистрированных доменных имен превысило 2 млн.
2000	Количество индексируемых веб-страниц превысило 1 млрд.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Стандартизация в Интернет

Результат работы по стандартизации воплощается в документах RFC.

RFC (англ. Request for Comments) — документ из серии пронумерованных информационных документов Интернета, содержащих технические

Примеры популярных RFC-документов.

Номер RFC	Тема
RFC 768	UDP
RFC 791	IP
RFC 793	TCP
RFC 822	Формат электронной почты, заменен RFC 2822
RFC 959	FTP
RFC 1034	DNS — концепция
RFC 1035	DNS — внедрение
RFC 1591	Структура доменных имен
RFC 1738	URL
RFC 1939	Протокол POP версии 3 (POP3)
RFC 2026	Процесс стандартизации в Интернете
RFC 2045	MIME
RFC 2231	Кодировка символов
RFC 2616	HTTP
RFC 2822	Формат электронной почты
RFC 3501	IMAP версии 4 издание 1 (IMAP4rev1)

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Консорциум W3C

Консорциум W3C — организация, разрабатывающая и внедряющая технологические стандарты для Интернета и WWW. Миссия W3C формулируется следующим образом: "Полностью раскрыть потенциал Всемирной паутины путем создания протоколов и принципов, гарантирующих долгосрочное развитие Сети". Две другие важнейшие задачи Консорциума — обеспечить полную "интернационализацию Сети" и сделать ее доступной для людей с ограниченными возможностями.

W3C разрабатывает для WWW единые принципы и стандарты, называемые "***Рекомендациями***", которые затем внедряются разработчиками программ и оборудования. Благодаря *Рекомендациям* достигается совместимость между программными продуктами и оборудованием различных компаний, что делает сеть WWW более совершенной, универсальной и удобной в использовании.

Все *Рекомендации* W3C открыты, то есть, не защищены патентами и могут внедряться любым человеком без каких-либо финансовых отчислений Консорциуму.

Для удобства пользователей Консорциумом созданы специальные ***программы-валидаторы*** (англ. Online Validation Service), которые доступны по сети и могут за несколько секунд проверить документы на соответствие популярным Рекомендациям W3C.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

STANDARDS

- Web Design and Applications
- Web Architecture
- Semantic Web
- XML Technology
- Web of Services
- Web of Devices
- Browsers and Authoring Tools

... or view all

WEB FOR ALL

- W3C A to Z
- Accessibility
- Internationalization
- Mobile Web

Workshop Report: Linked Enterprise Data Workshop

18 January 2012 | Archive

W3C today published the final report of the Linked Enterprise Data Workshop, hosted by W3C on t December in Cambridge, MA, USA. This workshop provided a way for the community to meet and d some of the challenges when deploying application relying on the principles of Linked Data. The presentations covered many different topics, ranging from the benefits a set of additional conventio would bring to specific technical issues such as the challenges of dealing with the reality that URLs change sometimes, as well as the need for a more robust security model, and specific gaps in the c set of standards.

Participants of the Workshop agreed that W3C should create a Working Group to define a "Linked Platform". This is expected to be an enumeration of specifications which constitute Linked Data, wit small additional specifications to cover specific functionality such as pagination. We anticipate a dra charter will be available in the coming weeks.

Group Note: MMI interoperability test report

24 January 2012 | Archive

Last Call: XML processor profiles

24 January 2012 | Archive

W3C Invites Implementer Feedback on XML Security 2.0 Specifications

Структура и принципы WWW

Сеть WWW образуют миллионы ***веб-серверов***, расположенных по всему миру. *Веб-сервер* является программой, запускаемой на подключенном к сети компьютере и передающей данные по протоколу HTTP.

Для идентификации ресурсов (зачастую файлов или их частей) в WWW используются идентификаторы ресурсов ***URI*** (Uniform Resource Identifier). Для определения местонахождения ресурсов в этой сети используются локаторы ресурсов ***URL*** (Uniform Resource Locator). Такие URL-локаторы представляют собой комбинацию URI и системы DNS.

Доменное имя (или IP-адрес) входит в состав URL для обозначения компьютера (его сетевого интерфейса), на котором работает программа веб-сервер.

На клиентском компьютере для просмотра информации, полученной от веб-сервера, применяется специальная программа — *веб-браузер*. Основная функция веб-браузера - отображение гипертекстовых страниц (веб-страниц). Для создания гипертекстовых страниц в WWW изначально использовался язык HTML. Множество веб-страниц образуют *веб-сайт*.

Доцент кафедри інформатики ХТУРЕ, начальник відділу контроля якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Прокси-серверы

Прокси-сервер **(proxy-server)** — служба в компьютерных сетях, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам.

Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс, расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного *кеша* (если имеется). В некоторых случаях запрос клиента или ответ сервера может быть изменен прокси-сервером в определенных целях. Также прокси-сервер позволяет защищать клиентский компьютер от некоторых сетевых атак.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Прокси-серверы применяются для следующих целей:

- обеспечение доступа с компьютеров локальной сети в Интернет;
- кеширование данных: если часто происходят обращения к одним и тем же внешним ресурсам, то можно держать их копию на прокси-сервере и выдавать по запросу, снижая тем самым нагрузку на канал во внешнюю сеть и ускоряя получение клиентом запрошенной информации.
- сжатие данных: прокси-сервер загружает информацию из Интернета и передает информацию конечному пользователю в сжатом виде.
- защита локальной сети от внешнего доступа: например, можно настроить прокси-сервер так, что локальные компьютеры будут обращаться к внешним ресурсам только через него, а внешние компьютеры не смогут обращаться к локальным вообще (они "видят" только прокси-сервер).
- ограничение доступа из локальной сети к внешней: например, можно запретить доступ к определенным веб-сайтам, ограничить использование интернета каким-то локальным пользователям, устанавливать квоты на трафик или полосу пропускания, фильтровать рекламу и вирусы.
- анонимизация доступа к различным ресурсам. Прокси-сервер может скрывать сведения об источнике запроса или пользователе. В таком случае целевой сервер видит лишь информацию о прокси-сервере, например, IP-адрес, но не имеет возможности определить истинный источник запроса. Существуют также искажающие прокси-серверы, которые передают целевому серверу ложную информацию об истинном пользователе.

Доцент Кафедри інформатики ХНУРЕ, начальник Відділу Контроля Якості ПВ INFOSTROY LTD Матат О.О.
E-mail: [illegible]@infostroy.com.ua

Протоколы Интернет прикладного уровня

Самый верхний уровень в иерархии протоколов Интернет занимают следующие протоколы прикладного уровня:

- ***DNS*** - распределенная система доменных имен, которая по запросу, содержащему доменное имя хоста сообщает IP адрес;
- ***HTTP*** - протокол передачи гипертекста в Интернет;
- ***HTTPS*** - расширение протокола HTTP, поддерживающее шифрование;
- ***FTP*** (File Transfer Protocol - RFC 959) - протокол, предназначенный для передачи файлов в компьютерных сетях; FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверам; FTP позволяет обмениваться файлами и выполнять операции над ними через TCP-сети. Данный протокол работает независимо от операционных систем.
- ***Telnet*** (TELecommunication NETwork - RFC 854) - сетевой протокол для реализации текстового интерфейса по сети; Протокол *telnet* работает в соответствии с принципами архитектуры "клиент-сервер" и обеспечивает эмуляцию алфавитно-цифрового терминала, ограничивая пользователя режимом командной строки. Приложение *telnet* предоставило язык для общения терминалов с удаленными компьютерами.
- ***SSH*** (Secure Shell - RFC 4251) - протокол прикладного, позволяющий производить удаленное управление операционной системой и передачу файлов. В отличие от Telnet шифрует весь трафик; Сходен по функциональности с протоколами telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH-клиенты и SSH-серверы имеются для большинства операционных систем.

Доцент кафедри інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Почтовые протоколы.

- ***POP3***(Post Office Protocol Version 3 - RFC 1939) – протокол почтового клиента, который используется почтовым клиентом для **получения** сообщений электронной почты с сервера;
- ***IMAP*** (Internet Message Access Protocol - RFC 3501) - протокол **доступа** к электронной почте в Интернет. Аналогичен POP3, однако предоставляет пользователю богатые возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без необходимости постоянной пересылки с сервера и обратно файлов с полным содержанием писем;
- ***SMTP***(Simple Mail Transfer Protocol — RFC 2821) – протокол, который используется для **отправки** почты от пользователей к серверам и между серверами для дальнейшей пересылки к получателю. Для приема почты почтовый клиент должен использовать протоколы POP3 или IMAP;

.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Протокол HTTP

HTTP (HyperText Transfer Protocol - RFC 1945, RFC 2616) - протокол прикладного уровня для передачи гипертекста.

Все программное обеспечение для работы с протоколом HTTP разделяется на три основные категории:

- ***Серверы*** - поставщики услуг хранения и обработки информации (обработка запросов).
- ***Клиенты*** - конечные потребители услуг сервера (отправка запросов).
- ***Прокси-серверы*** для поддержки работы транспортных служб.

Основными клиентами являются ***браузеры*** например: Internet Explorer, Opera, Mozilla Firefox, Google Chrome, Safari и другие. Наиболее популярными реализациями веб-серверов являются: Internet Information Services (IIS), Apache, lighttpd, nginx. Наиболее известные реализации прокси-серверов: Squid, UserGate, Multiproxy, Naviscope.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Схема HTTP-сеанса

Установление TCP-соединения.

- Запрос клиента.
- Ответ сервера.
- Разрыв TCP-соединения.

Таким образом, клиент посылает серверу запрос, получает от него ответ, после чего взаимодействие прекращается. Обычно запрос клиента представляет собой требование передать HTML-документ или какой-нибудь другой ресурс, а ответ сервера содержит код этого ресурса.

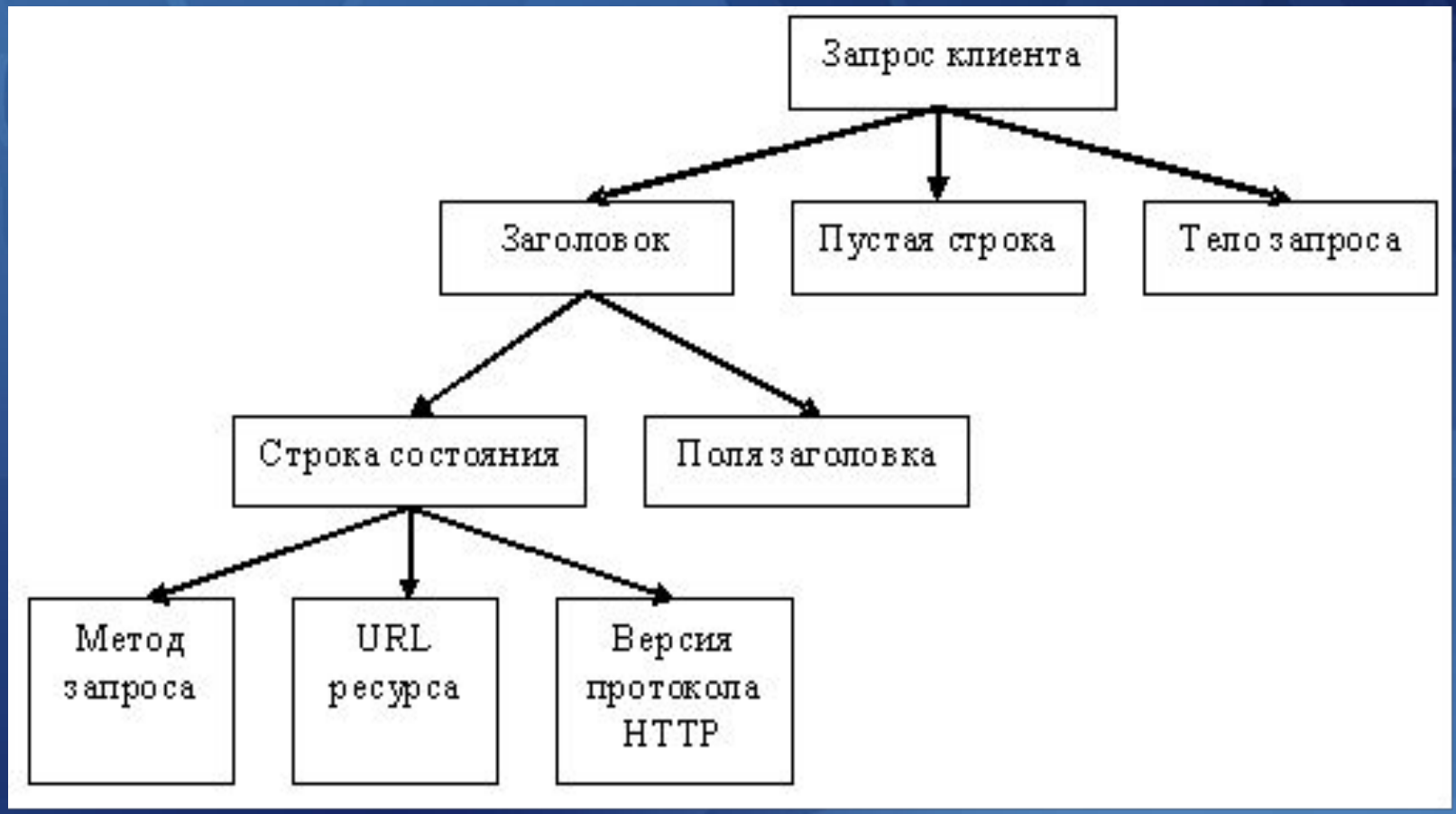
В *состав HTTP-запроса*, передаваемого клиентом серверу, входят следующие компоненты.

- Строка состояния (строка-статус или строка запроса).
- Поля заголовка.
- Пустая строка.
- Тело запроса.

Строку состояния вместе с *полями заголовка* иногда называют также *заголовком запроса*.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Структура запроса клиента.



Строка состояния имеет следующий формат:

метод_запроса URL_ресурса версия_протокола_HTTP

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Методы запроса

Метод, указанный в строке состояния, определяет способ воздействия на ресурс, URL которого задан в той же строке. Метод может принимать значения GET, POST, HEAD, PUT, DELETE и т.д. Несмотря на обилие методов, для веб-программиста по-настоящему важны лишь два из них: GET и POST.

- **GET -** предназначается ***для получения ресурса с указанным URL***. Получив запрос GET, сервер должен прочитать указанный ресурс и включить код ресурса в состав ответа клиенту. Ресурс, URL которого передается в составе запроса, не обязательно должен представлять собой HTML-страницу, файл с изображением или другие данные. URL ресурса может указывать на исполняемый код программы, который, при соблюдении определенных условий, должен быть запущен на сервере. В этом случае клиенту возвращается не код программы, а данные, сгенерированные в процессе ее выполнения. Несмотря на то что, по определению, метод GET предназначен для получения информации, он может применяться и в других целях. Метод GET вполне подходит для передачи небольших фрагментов данных на сервер.
- **POST -** основное назначение - ***передача данных на сервер***. Однако, подобно методу GET, метод POST может применяться по-разному и нередко используется ***для получения информации с сервера***. Как и в случае с методом GET, URL, заданный в строке состояния, указывает на конкретный ресурс. Метод POST также может использоваться для запуска процесса.

Методы **HEAD** и **PUT** являются модификациями методов GET и POST.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Элементы заголовка запроса (продолжение)

Версия протокола HTTP, как правило, задается в следующем формате:

HTTP/версия.модификация

Поля заголовка, следующие за строкой состояния, позволяют уточнять запрос, т.е. передавать серверу дополнительную информацию.

Поле заголовка имеет следующий формат:

Имя_поля: Значение

Назначение поля определяется его именем, которое отделяется от значения двоеточием.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Поля заголовка запроса HTTP.

Поля заголовка HTTP-запроса	Значение
Host	Доменное имя или IP-адрес узла, к которому обращается клиент
Referer	URL документа, который ссылается на ресурс, указанный в строке состояния
From	Адрес электронной почты пользователя, работающего с клиентом
Accept	MIME-типы данных, обрабатываемых клиентом. Это поле может иметь несколько значений, отделяемых одно от другого запятыми. Часто поле заголовка Accept используется для того, чтобы сообщить серверу о том, какие типы графических файлов поддерживает клиент
Accept-Language	Набор двухсимвольных идентификаторов, разделенных запятыми, которые обозначают языки, поддерживаемые клиентом
Accept-Charset	Перечень поддерживаемых наборов символов
Content-Type	MIME-тип данных, содержащихся в теле запроса (если запрос не состоит из одного заголовка)
Content-Length	Число символов, содержащихся в теле запроса (если запрос не состоит из одного заголовка)
Range	Присутствует в том случае, если клиент запрашивает не весь документ, а лишь его часть
Connection	Используется для управления TCP-соединением. Если в поле содержится Close, это означает, что после обработки запроса сервер должен закрыть соединение. Значение Keep-Alive предлагает не закрывать TCP-соединение, чтобы оно могло быть использовано для последующих запросов
User-Agent	Информация о клиенте

Пример HTML-запроса, сгенерированного браузером

GET http://oak.oakland.edu/ HTTP/1.0

Connection: Keep-Alive

User-Agent: Mozilla/4.04 [en] (Win95; I)

Host: oak.oakland.edu

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*

Accept-Language: en

Accept-Charset: iso-8859-l,*,utf-8

Получив от клиента запрос, сервер должен ответить ему. Знание структуры ответа сервера необходимо разработчику веб-приложений, так как программы, которые выполняются на сервере, должны самостоятельно формировать ответ клиенту.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Ответ сервера

Ответ сервера также состоит из четырех перечисленных ниже компонентов.

- Строка состояния.
- Поля заголовка.
- Пустая строка.
- Тело ответа.

Ответ сервера клиенту начинается со строки состояния, которая имеет следующий формат:

Версия_протокола Код_ответа Пояснительное_сообщение

Версия_протокола задается в том же формате, что и в запросе клиента, и имеет тот же смысл.

Код_ответа - это трехзначное десятичное число, представляющее в закодированном виде результат обслуживания запроса сервером.

Пояснительное_сообщение дублирует код ответа в символьном виде. Это строка символов, которая не обрабатывается клиентом. Она предназначена для системного администратора или оператора, занимающегося обслуживанием системы, и является расшифровкой кода

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@kture.kharkov.ua

Код ответа сервера

Из трех цифр, составляющих код ответа, первая (старшая) определяет класс ответа, остальные две представляют собой номер ответа внутри класса. *Например*, если запрос был обработан успешно, клиент получает следующее сообщение:

HTTP/1.0 200 OK

За версией протокола HTTP 1.0 следует код 200. В этом коде символ 2 означает успешную обработку запроса клиента, а остальные две цифры (00) — номер данного сообщения.

В используемых в настоящее время реализациях протокола HTTP первая цифра не может быть больше 5 и определяет следующие ***классы ответов***:

1 - специальный класс сообщений, называемых информационными. Код ответа, начинающийся с 1, означает, что сервер продолжает обработку запроса. При обмене данными между HTTP-клиентом и HTTP-сервером сообщения этого класса используются достаточно редко.

2 - успешная обработка запроса клиента.

3 - перенаправление запроса. Чтобы запрос был обслужен, необходимо предпринять дополнительные действия.

4 - ошибка клиента. Как правило, код ответа, начинающийся с цифры 4, возвращается в том случае, если в запросе клиента встретилась синтаксическая ошибка.

5 - ошибка сервера. По тем или иным причинам сервер не в состоянии выполнить запрос.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Классы кодов ответов сервера

Код	Расшифровка	Интерпретация
100	Continue	Часть запроса принята, и сервер ожидает от клиента продолжения запроса
200	OK	Запрос успешно обработан, и в ответе клиента передаются данные, указанные в запросе
201	Created	В результате обработки запроса был создан новый ресурс
202	Accepted	Запрос принят сервером, но обработка его не окончена. Данный код ответа не гарантирует, что запрос будет обработан без ошибок.
206	Partial Content	Сервер возвращает часть ресурса в ответ на запрос, содержавший поле заголовка Range
301	Multiple Choice	Запрос указывает более чем на один ресурс. В теле ответа могут содержаться указания на то, как правильно идентифицировать запрашиваемый ресурс
302	Moved Permanently	Затребованный ресурс больше не располагается на сервере
302	Moved Temporarily	Затребованный ресурс временно изменил свой адрес
400	Bad Request	В запросе клиента обнаружена синтаксическая ошибка
403	Forbidden	Имеющийся на сервере ресурс недоступен для данного пользователя
404	Not Found	Ресурс, указанный клиентом, на сервере отсутствует
405	Method Not Allowed	Сервер не поддерживает метод, указанный в запросе
500	Internal Server Error	Один из компонентов сервера работает некорректно
501	Not Implemented	Функциональных возможностей сервера недостаточно, чтобы выполнить запрос клиента
503	Service Unavailable	Служба временно недоступна
505	HTTP Version not Supported	Версия HTTP, указанная в запросе, не поддерживается сервером

Поля заголовка ответа веб-сервера.

Имя поля	Описание содержимого
Server	Имя и номер версии сервера
Age	Время в секундах, прошедшее с момента создания ресурса
Allow	Список методов, допустимых для данного ресурса
Content-Language	Языки, которые должен поддерживать клиент для того, чтобы корректно отобразить передаваемый ресурс
Content-Type	MIME-тип данных, содержащихся в теле ответа сервера
Content-Length	Число символов, содержащихся в теле ответа сервера
Last-Modified	Дата и время последнего изменения ресурса
Date	Дата и время, определяющие момент генерации ответа
Expires	Дата и время, определяющие момент, после которого информация, переданная клиенту, считается устаревшей
Location	В этом поле указывается реальное расположение ресурса. Оно используется для перенаправления запроса
Cache-Control	Директивы управления кэшированием. Например, no-cache означает, что данные не должны кэшироваться

E-mail: helen.matat@infostroy.com.ua

Пример ответа на запрос

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
X-Powered-By: ASP.NET
Date: Mon, 20 OCT 2008 11:25:56 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Sat, 18 Oct 2008 15:05:44 GMTE
Tag: "b66a667f948c92:8a5«
Content-Length: 426
<html><body><form action='http://localhost/Scripts/test.pl'>
Operand1: <input type='text' name='A'>
Operand2: <input type='text' name='B'>
Operation:
<select name='op'>
<option value='+'>+</option>
<option value='-'>-</option>
<option value='*'>*</option>
<option value='/'>/</option>
<select>
<input type='submit' value='Calculate!'>
</from>
</body>
</html>

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Безопасность передачи данных HTTP

Протокол HTTP предназначен для передачи символьных данных в открытом (незашифрованном) виде. Для защиты передаваемых данных от несанкционированного доступа исполюзуют расширения протокола, самым простейшим из которых является расширение ***HTTPS***, при котором данные, передаваемые по протоколу HTTP, "упаковываются" в криптографический протокол SSL или TLS, тем самым обеспечивая защиту этих данных. В отличие от HTTP, для HTTPS по умолчанию используется TCP-порт 443. Чтобы подготовить веб-сервер для обработки HTTPS соединений, администратор должен получить и установить в систему сертификат для этого веб-сервера.

SSL (Secure Sockets Layer) - криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. При его использовании создается защищенное соединение между клиентом и сервером. SSL изначально разработан компанией Netscape Communications. Впоследствии на основании протокола SSL 3.0 был разработан и принят стандарт RFC, получивший название TLS. Этот протокол использует шифрование с открытым ключом для подтверждения подлинности передатчика и получателя. Поддерживает надежность передачи данных за счет использования корректирующих кодов и безопасных хэш-функций. На нижнем уровне многоуровневого транспортного протокола (например, TCP) он является протоколом записи и используется для инкапсуляции различных протоколов (POP3, IMAP, SMTP или HTTP).

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

3 типа аутентификации

Для доступа к веб-страницам, защищенным протоколом SSL, в URL вместо схемы http, как правило, подставляется схема https, указывающая на использование SSL-соединения. Стандартный TCP-порт для соединения по протоколу https — 443. Для работы SSL требуется, чтобы на сервере имелся ***SSL-сертификат***.

В сети Веб поддерживаются 3 типа аутентификации при клиент-серверных взаимодействиях:

Basic - базовая аутентификация: имя пользователя и пароль передаются в заголовках http-пакетов. Пароль не шифруется и присутствует в чистом виде в кодировке base64. Для ***Basic*** использование SSL является обязательным.

Digest- дайджест-аутентификация: пароль пользователя передается в хешированном виде. По уровню конфиденциальности паролей этот тип мало чем отличается от предыдущего, так как атакующему все равно, действительно ли это настоящий пароль или только хеш от него: перехватив удостоверение, он все равно получает доступ к конечной точке. Для ***Digest*** использование SSL является обязательным.

Integrated - интегрированная аутентификация, при которой клиент и сервер обмениваются сообщениями для выяснения подлинности друг друга с помощью протоколов NTLM или Kerberos. Этот тип аутентификации защищен от перехвата удостоверений пользователей, поэтому для него не требуется протокол SSL. Только при использовании данного типа аутентификации можно работать по схеме http, во всех остальных случаях необходимо использовать

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
Email: [redacted]

Cookie

Поскольку HTTP-сервер не помнит предыстории запросов клиентов, то каждый запрос обрабатывается независимо от других, и у сервера нет возможности определить, исходят ли запросы от одного клиента или разных клиентов.

Механизм ***cookie*** позволяет серверу хранить информацию на компьютере клиента и извлекать ее оттуда.

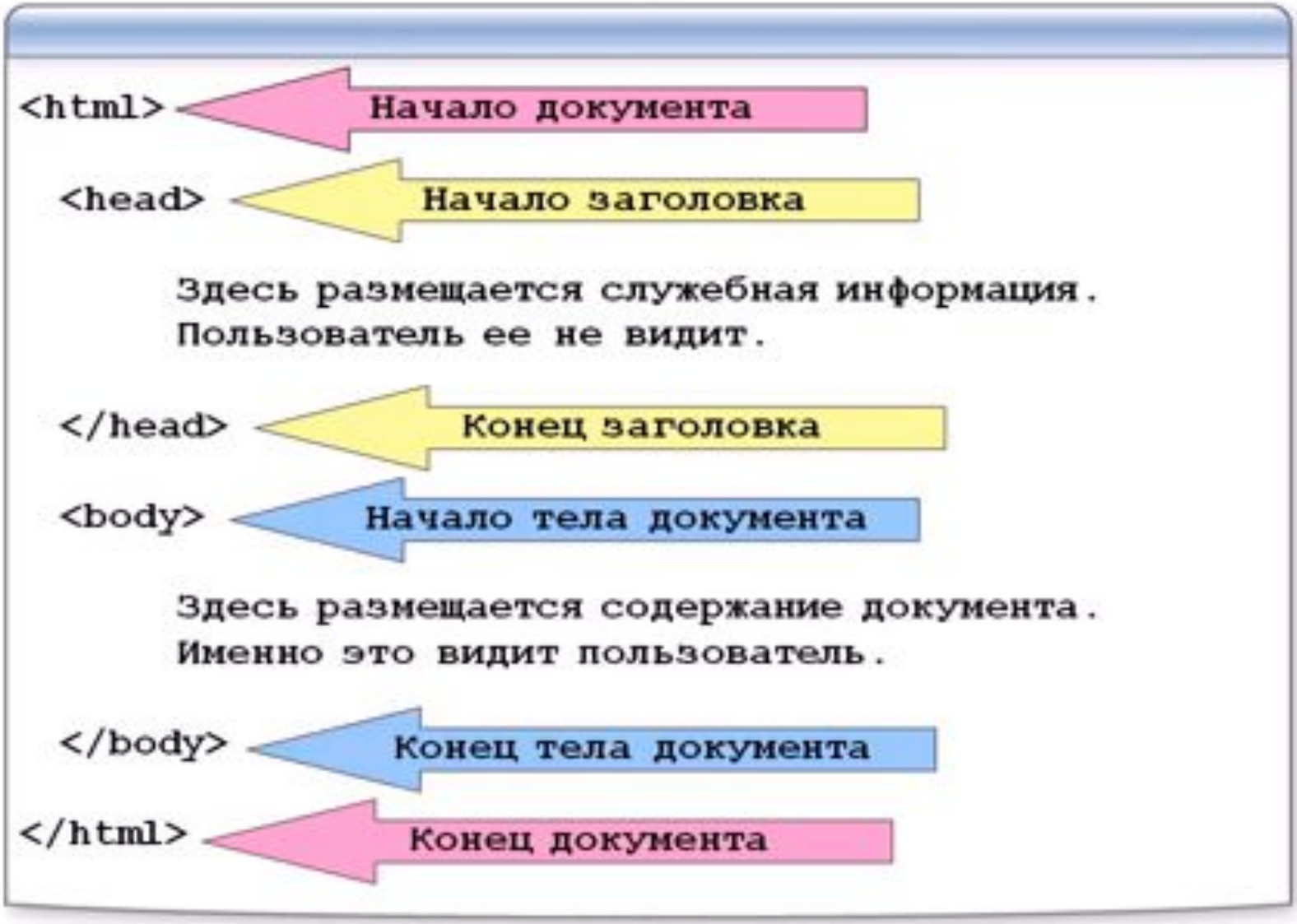
Инициатором записи cookie выступает сервер. Если в ответе сервера присутствует поле заголовка ***Set-cookie***, клиент воспринимает это как команду на запись ***cookie***. В дальнейшем, если клиент обращается к серверу, от которого он ранее принял поле заголовка ***Set-cookie***, помимо прочей информации он передает серверу данные cookie. Для передачи указанной информации серверу используется поле заголовка ***Cookie***.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Пример обмена данными Cookie

Предположим, что клиент передает запросы на серверы **A**, **B** и **C**. Предположим также, что сервер **B**, в отличие от **A** и **C**, передает клиенту команду записать cookie. Последовательность запросов клиента серверу и ответов на них будет выглядеть приблизительно следующим образом.

1. Передача запроса серверу **A**.
2. Получение ответа от сервера **A**.
3. Передача запроса серверу **B**.
4. Получение ответа от сервера **B**. В состав ответа входит поле заголовка SetCookie. Получив его, клиент записывает cookie на диск.
5. Передача запроса серверу **C**. Несмотря на то что на диске хранится запись cookie, клиент не предпринимает никаких специальных действий, так как значение cookie было записано по инициативе другого сервера.
6. Получение ответа от сервера **C**.
7. Передача запроса серверу **A**. В этом случае клиент также никак не реагирует на тот факт, что на диске хранится cookie.
8. Получение ответа от сервера **A**.
9. Передача запроса серверу **B**. Перед тем как сформировать запрос, клиент определяет, что на диске хранится запись cookie, созданная после получения ответа от сервера **B**. Клиент проверяет, удовлетворяет ли данный запрос некоторым требованиям, и, если проверка дает положительный результат, включает в заголовок запроса поле Cookie.

Доцент Кафедри Інформатики ХНУРЕ, начальник Відділу Контроля Якості ПЗ INFOSTROY LTD Матат О.О.
E-mail: helen.matat@infostroy.com.ua

Структура HTML документа