**Коллоквиум 1**

1. Сеть Интернет: 4 компонента.

**Интернет:** Всемирная компьютерная сеть, построенная на основе стека протоколов TCP/IP.



**Интернет** – 4 компонента:

* Стек протоколов TCP-IP — это основа Интернета.
* Интернет службы (DNS, DHCP, SMTP, POP3 и т.д.).
* Документация в формате RFC и STD.
* Система организаций, которые поддерживают сеть

1. Понятие службы Интернет.

**Internet-служба**: другое название Internet-сервис, один из видов Internet-ресурса, имеющий специальное назначение (DNS, WWW, E-mail, FTP, ICQ, Telnet)

1. Понятие ресурса Интернет.

**Internet-ресурс:** сущность в сети Internet, имеющая адрес (опубликованная в Internet сущность).

1. Основные организации управления Интернет.

 IETF (разрабатывает стандарты интернета в RFC)

 ICANN (распространяет доменные имена)

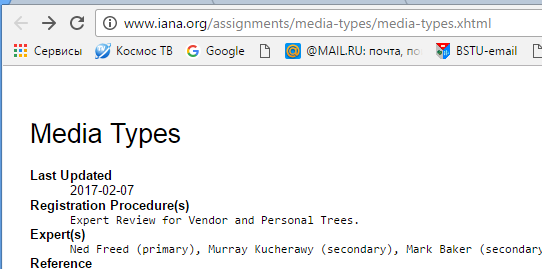
 IANA (регистрирует MIME)

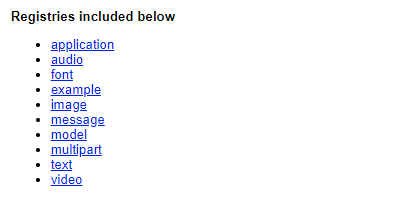
 ISOC (развитие, внедрение и распространение новых интернет-технологий)

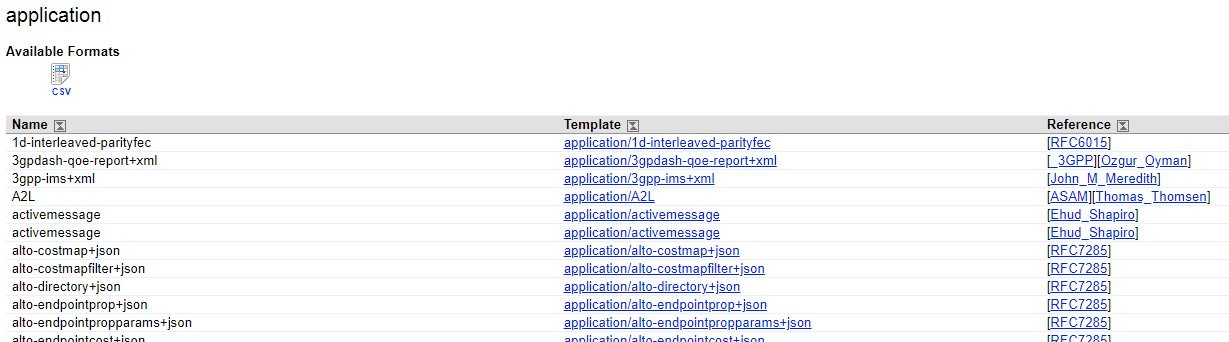
 W3C (WWW Consortium, некоммерческая организация для согласования стандартов) — согласовывают всё, что касается веб-программирования (HTTP, CSS, SVG, URI/URL, XML, PNG, JPEG, …). Также они изготавливают стандарты и регистрируют их в IETF

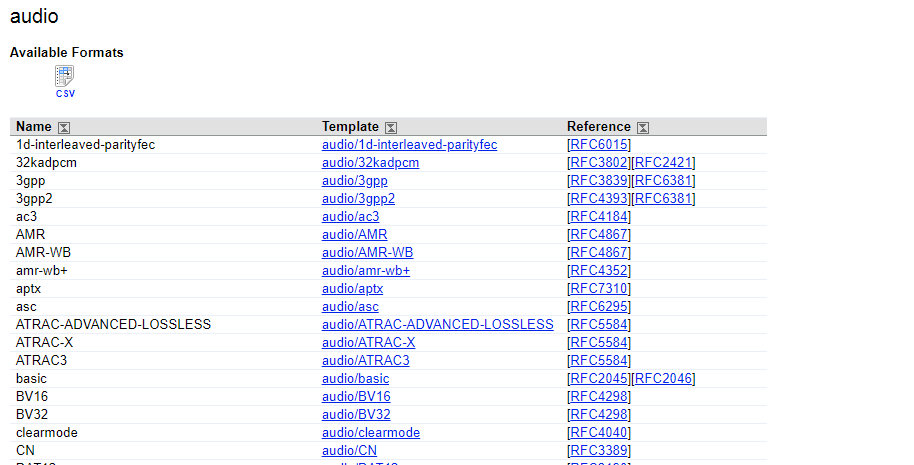
1. Система стандартизации Интернет. Основные стандартизующие организации.
2. Понятие MIME.

**Multipurpose Internet Mail Extensions** - многоцелевые расширения Internet-почты. Используется и как стандарт кодирования Internet-сообщений.









1. Определение web-приложения.

**Web-приложение:** клиент-серверное приложение, применяющее для обмена данными протокол HTTP; может быть просто web-приложением (HTML+HTTP) или web-службой (API, HTTP-транспорт, формат XML, JSON)

1. Протокол HTTP: основные свойства.

* версии HTTP/1.1 – действующий (текстовый), HTTP/2 – черновой (не распространен, бинарный);
* два типа абонентов: клиент и сервер;
* два типа сообщений: request и response;
* от клиента к серверу – request;
* от сервера к клиенту – response;
* на один request всегда один response, иначе ошибка;
* одному response всегда один request, иначе ошибка;
* TCP-порты: 80, 443;
* для адресации используется URI или URN;
* поддерживается W3C, описан в нескольких RFC.

1. Структура запроса.

Request:

1) метод

2) uri

3) версия протокола

4) заголовки

5) параметры

6) расширение

1. Основные методы HTTP-запроса.



**Методы**: **get** – запросы могут только извлекать данные; **post** – исп для отправки сущностей к ресурсу; **head** – запрашивает, как и get, но без тела ответа; **delete** – удаляет указанный ресурс; **connect** – устанавливает соединение к серверу, определенному ресурсу; **options** – исп для описания параметров соединения с ресурсом; **trace** – выполняет вызов возвращаемого тестового сообщения с ресурса; **patch(или put)** – исп для частичного изменения ресурса.

1. Структура ответа.

Response:

1) Код состояния

2) Пояснение к коду

3) Версия протокола

4) Заголовки

5) Расширения

1. Коды статуса ответа.

1xx: информационные

2хх: успешный ответ

3хх: переадресация

4хх: ошибка клиента

5хх: ошибка сервера

1. HTTP-заголовки: классификация, примеры.

* **General**: общие заголовки, используются в запросах и ответах;



* **Request**: используются только в запросах;



* **Response:** используются только в ответах;



* **Entity**: для сущности в ответах и запросах.



1. HTTP-переадресация: принцип работы, заголовки, статусы
2. HTTP: кеширование на стороне клиента, принцип, заголовки.
3. Понятия URI, URL, URN.

* **URI: Uniform Resource Identifier** – унифицированный идентификатор ресурса (документ, изображение, файл, служба, электронная почта,…).
* **URL: Uniform Resource Location** - унифицированный локатор ресурса , содержащий местонахождение ресурса и способ обращения (протокол) к ресурса, описывает множество URI.
* **URN: Uniform Resource Name** - унифицированное имя ресурса – URI, имя ресурса, не содержащее месторасположение и способ доступа к ресурсу. В будущем URN должен заменить URL (для решения проблем с перемещением ресурсов в Internet).
* **URI, URL, URN** –рекомендуется использовать термин URI

1. Понятие DOM.

Объектная Модель Документа (DOM) – это программный интерфейс (API) для HTML и XML документов. DOM предоставляет структурированное представление документа и определяет то, как эта структура может быть доступна из программ, которые могут изменять содержимое, стиль и структуру документа.

1. Понятие CSS.

Cascading Style Sheets (CSS) — это язык иерархических правил (таблиц стилей), используемый для представления внешнего вида документа, написанного на HTML или XML (включая различные языки XML, такие как SVG и XHTML. CSS описывает, каким образом элемент должен отображаться на экране, на бумаге, голосом или с использованием других медиа средств.

1. HTML: назначение, стандарты.

HTML (HyperText Markup Language — «язык гипертекстовой разметки») — самый базовый строительный блок Веба. Он определяет содержание и структуру веб-контента.

1. HTML: перечень тегов, генерирующих HTTP-запросы.

* адресная URI-строка (GET);
* HTML-тег: **<form>**;
* HTML-тег: **<a>** (GET);
* HTML-тег: **<img>** (GET);
* HTML-тег: **<script>** (GET);
* HTML-тег: **<link>** (GET);
* HTML-тег: **<audio>** (GET);
* HTML-тег: **<video>** (GET);
* HTML-тег: **<frame>** GET), не поддерживается в HTML5;
* Объект web-браузера:**XMLHTTPRequest**;
* JavaScript API: **web-сокеты**.

1. Основные объекты web-сервера (не менее 6).
2. HTTP-аутентификации: общая схема.



1. HTTP-аутентификации: отличие BASIC и DIGEST.





1. HTTP-аутентификации: FORMS.



1. Протокол TLS: уровень, назначение, рукопожатие.
2. Протокол TLS: X509-сертификаты, назначение.
3. Token-аутентификация: принцип работы.

**Принцип работы:**

* 1. Клиент запрашивает у сервера токен аутентификации.
  2. Сервер проверяет учетные данные пользователя и, если они верны, выдает клиенту токен аутентификации.
  3. Клиент отправляет запрос на защищенный ресурс с токеном аутентификации в заголовке Authorization.
  4. Сервер проверяет токен аутентификации и, если он действителен, разрешает доступ к ресурсу

1. Протокол OAuth 2.0: назначение, принципы применения.

1. Протокол WebDAV: назначение, принцип применения.
2. Протокол HTTP2: структура сообщений, основные отличия от протокола HTTP 1.1.
3. Протокол HTTP3: основные особенности.
4. Архитектура браузера, перечень и назначение основных компонентов.
5. Назначение и возможности IndexDB API.

СУБД, встроенная в web-browser, NoSQL, система индекированных хранилищ, формат данных JSON, JavaScript API, асинхронная работа, транзакционная модель, поддерживается курсор, использует события DOM об уведомлении javascrpt-приложения (error/succes).

1. WebRTC: назначение, перечень протоколов.