HTTP — ассимитричный протокол. (правило обмена данными между двумя приложениями) (ассимитричный – если в одну сторону идут сообщения одного типа, обратно другого)

(на каждый запрос обязательно есть ответ, два подряд запроса быть не может)

Request:

Протокол и версия

Метод (get, post)

Uri

Набор заголовков (content-type: content-length)

Набор параметров (? В адресной строке, или через Input type через форму в submit)

Тело (ну впринципе может и не быть)

Response:

Статус (инфо, успех, переадресация, ошибка клиента , ошибка сервера)

Протокол+ версия

Заголовки (стандарные и те которые можно делать самим)

Тело

Web приложение приложение взаимодейстрвующее по протоколу http

Назначение сервера — получить запрос, сформировать request и response, передать нашему приложению, которое отвечает на запрос и заполняет response и передает серверу который передает сформированую битовую последоватльность обратно клиенту.

Clr (common language runtime (программа .dll написанная по спецификации com)) + fcl (foundation class library (стандартная библиотека классов))

CTS (Common Type System), VES (virtual execution System)

CLR:

MSIL(результат компиляции (написанный с соответствии со спецификацией CLI(infrastructure)))

JIT – на лету компилирует код(

В приложении есть много классов и jit компилирует по типам (код компилируется частично)

) любое .net приложение «разгоняется»

ASP.NET – технология предназначенная для разработки распределенных приложений.

NET FRAMEWORK (РЕАлизация спецификации mono под windows)

Iisapi.dll

Handler — набор .net классов предназначенных для первичной обработки объектов request

**ASP.NET**

В основе любого приложения лежит некоторая страница с расширением.

Страница .aspx

Движки для обработки страниц (aspx, razor)

Runat=”server” обязывает создать объект на сервере и сохранить его состояние.

9/19/2017

Ресурс то к чему можно добраться по url или uri.

Сессия – объект хранящий информацию о соединении с клиентом, создается при первом обращщении (session id, timeout)

Реализация сесси осуществляется через куки.

Серверные элементы сохраняют свое состояние.

Написали uri в браузере :

1. url
2. / something

Запрос отправляется на сервер адрес которого указан в uri

Библиотека ASPIISAPI + Набор HTTP обработчиков (системные и пользовательские)

хелперы - то что пишется в .cshtml через @

@Html.ActionLink - конвертится в anchord - есть перегрузки

@Html.BeginForm;

@Html.EditorFor(model => model.Surname);

**строготипизированное** представление - при создании необходимо указать класс модели (ну по идее можно ваще любой класс), ообъект этого класса будет передаваться в качестве параметра во veiw; теперь можно из view писать mode.some\_property и получать данные)); и теперь при вызове View нужно в качестве параметра объект указанного класса, если не передать по получиться что model = null;

если во Veiw не указано ни контроллера ни акции, то запрос вернется туда же.