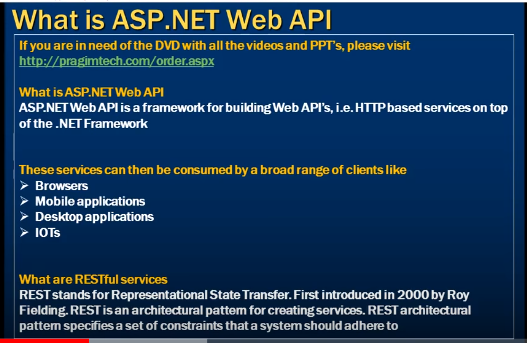
* Building restful services
* Can also be used for building non restful services
* Internet of things are devices that have ip address and can connect with internet and with other internet devices

Eg. Security system, electronic devices



* REST CONSTRAINTS

1. Client – request

2. Server – response

Independent implementation of server side as well as client side separation.

3. stateless – cannt store anything of client on server, every request treated indeplendtly

4. Cacheeable – lets client know how long is data required so that client don’t have to come to server for same data again.

Eg – list of employee

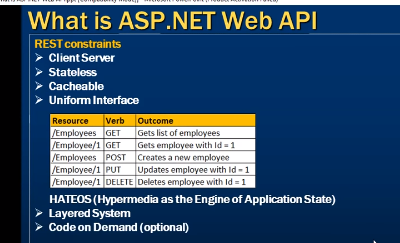
Avoides unnecessary processing and significantly improves performance

5. Uniform Interface – defines interface between client and server

Resourec – data entity (products, employess)

and http verbs(get,..delete)

Each resource is identitfy by a uniform URI



* WCF and WEB API

Windows communication foundation – can be used to create restful application then why we need web api?

* More configuration is required to turn a web service to a restful application
* 

Eg – we have a web service that can be used by Java client and other is .NET client

Java requires:

transport ptotocol : http

Message type : xml

.Net requires:

transport ptotocol : tcp

Message type : binary formate

http verbs:

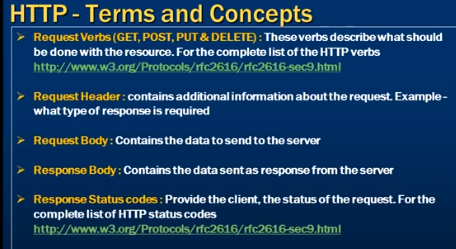


Put – we need particular id to update the part of particular employee



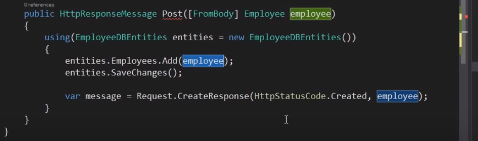
<https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

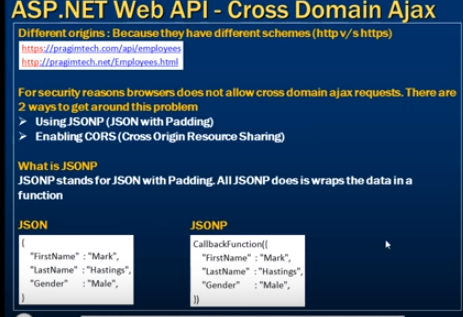
A request contains header and body

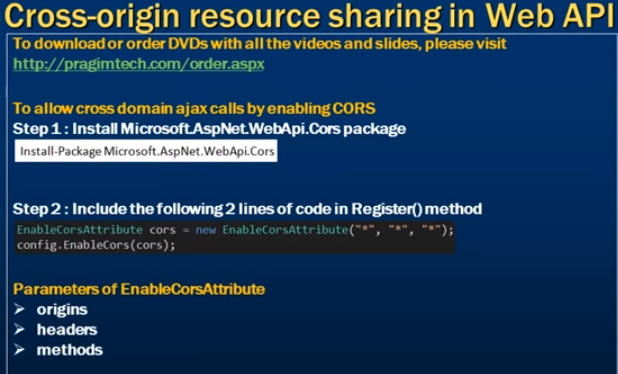
1. Request header: provides additional information eg. Want request in the form of xml or json
2. 

201 – created – to show http messes we can use HttpResponseMessage

204 – no content – void method

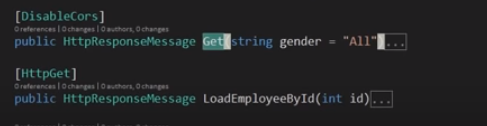








Cors is enable only for Employee controller



Disable cors for Get method

