Web Api questions and terms

# What is REST and RESTful

* Representational State Transfer
* RESTful refers to web services which use the REST architecture
* REST services focus on resources and how state should be transported over HTTP
* It supports the standards http verbs: Get/Put/Delete/Post for perform CRUD operation

Key terms:

* Resource-based
* Resources should be nouns, not verbs
* Representations of data
* Stateless
* Self-descriptive
* Cacheable
* Decoupled architecture and separate of concerns
* Http Verbs for CRUD operations
* Http response codes (100 Continue, 200 OK, 201 Created, 204 No Content, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error,

REST architecture defines **six constraints**

* ***Uniform interface*** – which describes the interface between client and server, which simplifies and decouples the architecture. Four principles of the uniform interface :
  + Resource-based:
    - the resource is identified in requests ***using URIs as resource identifiers***
    - The resources are separate from the representations returned to the client
  + Manipulation through representation
    - A client has enough information and metadata to modify/delete the resource
  + Self-descriptive messages
    - Each message includes enough information to describe how to process the message (i.e. media-type: JSON)
  + Hypermedia as the engine of Application State (HATEOAS)
    - Clients deliver state via body, query-string parameters, request headers, and the URI.
    - Services deliver state via response body, response codes, and response headers
* Stateless
* Cacheable
* Client-Server
* Layered System
  + A client doesn’t know if he is connected to an end server or an intermediary server
* Code on demand (optional)
  + Servers may extend the functionality of a client by transferring logic back to client which can be executed (i.e. Java applets, javascript ).

## Explain the REST architectural style for creating Web Api’s

* Simple URI represents the service address
* Http protocol for client/server communication
* XML/JSON as formatting language (i.e. Api return type)
* Stateless communication

## Testing tools

* SOAPUI for SOAP WS, Firefox Poster, Postman, Swagger

# RESTful Resource Naming

* Insert a new customer

POST <http://www.example.com/customers>

* To read a customer with Customer ID# 33245:  
  GET <http://www.example.com/customers/33245> (same URI for PUT/DELETE)
* Create a new order for a customer:
  + POST <http://www.example.com/customers/33245/orders>
* Get orders for a customer:
  + GET <http://www.example.com/customers/33245/orders>
* Get/Post line items for a customer order:
  + GET <http://www.example.com/customers/33245/orders/8769/lineitems>
  + <http://www.example.com/customers/33245/orders/8769/lineitems/1> (first line item)
  + POST <http://www.example.com/customers/33245/orders/8769/lineitems>