# Esercitazione di Fine Settimana – Week 6

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Michael |
|  |  | Cognome | Locci |
|  |  | Data | 10/15/2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi.

1. Cos’è un Web Service?

Un web service è un sistema software progettato per supportare l’interoperabilità tra diversi elaboratori sulla stessa rete o in un constesto distribuito, è costituito da un’interfaccia attraverso la quale due dispositivi o applicazioni possono comunicare fra loro. Le caratteristiche principali sono l’essere multipiattaforma e la condivisione delle informazioni. Quando viene utilizzato un web service, il client invia una richiesta ad un server che ne provoca un’azione e successivamente il server invia una risposta al client. Lo scambio del messagio avviene utilizzando il protocollo HTTP.

1. Quali parti costituiscono una HTTP Response?

Una HTTP Response ha la seguente struttura:

* HTTP version ovvero psecifica la versione HTTP che si sta usando, ad esempio versione 1.1 o 2.0
* Status code ovvero il codice che defiisce lo stato della richiesta effettuata dal client al server e possiamo avere diversi stati fra cui:
  + 200 ok operazione andata a buon fine
  + 404 not found ovvero al risorsa cercata non è stata trovata
  + 400 bad request ovvero è tato inserito un valore per l’esecuzione di una operazione ma non era valido
  + 201 ovvero operazione di creazione di un oggetto andata a buon fine
* Headers ovvero delle intestazioni che caratterizzano la risposta, come:
  + Content type ovvero il tipo di applicazione e la codifica
  + Server
  + Data ovvero la data di risposta alla richiesta del client
  + Content length ovvero la lunghezza del messaggio
* Body ovvero il contenuto della risposta che il server invia in conseguenza ad una richiesta da parte del client ed è in formato Json.

1. Descrivere l’utilizzo degli attributi [DataContract] e [OperationContract]

Il Data Contract è un accord tra un servizio e un cliente che descrive in modo astratto i dati che vengono scambiati ad esempio nella nostra esercitazione abbiamo marca come [DataContract] la classe Employee e come [DataMember] le sua proprietà, cosi facendo abbiamo definito che questi dati possono essere scambiati. Analogamente per le operazioni abbiamo il marcatore [OperationContract] che definisce quali metodi un servizio espone all’esterno e vengono definiti in in’interfaccia .NET ed implementati nel servizio che li espone.

1. Come vengono utilizzati gli HTTP Methods in un servizio REST?

Gli HTTP Method vengono utilzzati nella costruzione della richiesta HTTPClient ovvero dopo aver istanziato un oggetto di tipo HTTPClient si procede alla creazione di un oggetto Request e nella sua definizione vengono passata due parametri:

* Method dove si utilizza HTTPMethod per definire il tipo di medodo a cui fa riferimento questa richiesta e può essere GET, POST, DELETE, PUT, PATCH, HEAD, quest’ultimi sono noti come HTTP Verb
* RequestUri ovvero l’uri a cui è reperibile la richiesta, ad esempio ("https://localhost:44372/api/employees")

É possible specificare il body della Request utilizzando la proprietà Content, successivamente si invoca l’API tramite il metodo SendAsync() ed infine si va a vedere l’esito della chiamata ad esempio con if(response.isSuccessStatusCode) e all’interno dell’if si esegue la de-serializzaione.

1. Come viene configurato un servizio realizzato con ASP.NET Core WebAPI?

Per la configurazione di un servizio realizzato con ASP.NET Core WebApi, si aggingono le dipendenze ai progetti di Core, dove abbiamo le entità, le interfacce ed il business layer, ed al progetto EF, dove vengono modellate le entità e sono presenti le migration per la configurazione del database. A questo punto si configura nel file StartUp.cs la dependency injection e la configurazione di EFCore, se vogliamo usare swagger, lo si deve configurare, sempre in questo file andando ad aggiungere anche l’ApplicationName e l’ApplicazionVersion . A questo punto possiamo creare i nostril controller, uno per ogni entità ed andare ad implementare le operazioni di CRUD ed eventuali altri operazioni su queste entità. È possible inserire anche altri Middleware oltre a quelli già presenti.

**Esercitazione Pratica**

* Realizzare un database per la Gestione degli Ordini e dei Clienti.
  + ***Cliente***
    - *ID* (int, PK), CodiceCliente (*string*), Nome (*string*), Cognome (*string*)
  + ***Ordine***
    - *ID* (int, PK), *DataOrdine* (date), *CodiceOrdine* (string), *CodiceProdotto* (string), *Importo* (decimal)
  + La realizzazione dello strato di accesso al dato deve essere realizzata con EF Code-first.
* Realizzare un servizio WCF per la gestione di una Anagrafica Clienti (CRUD)
* Realizzare un servizio REST per la gestione di una Anagrafica Ordini (CRUD)
* Realizzare un client (Console app) per:
  + CRUD Clienti
  + CRUD Ordini
  + Stampa Elenco Ordini per uno specifico Cliente
  + Stampa Dettagli Ordine per uno specifico Ordine
  + Report Ordini aperti per Anno con numero di ordini e importo totale ordini