# Taco Fast Food API

## API Creation, Database

**Task:** Create an API for a Fast Food Taco Restaurant. All of the SQL commands needed are at the bottom. Use entity Framework to connect it to your api and use the specifications below to create the API.

**Build Specifications:**

- Create a TacosController with the following endpoints
    - GET /Tacos
        - Response: List of all Tacos
        - Response Code: 200 Ok
        - Take in a bool as an optional query parameter for SoftShell. Filter the response if SoftShell is provided.
    - GET /Tacos/{id}
        - Response: Taco that matches the provided id
        - Response Code:
            - 200 Ok (Taco found)
            - 404 Not Found (Taco not found)
        - Take in int id as a path parameter
    - POST /Tacos
        - Response: The created taco
        - Response Code: 201 Created
        - Take in Taco from body
    - DELETE /Tacos/{id}
        - Response: Nothing
        - Response Code: 204 No Content (Taco Deleted) or 404 Not found
- Create a DrinksController with the following endpoints
    - GET /Drinks
        - Response: List of all Drinks
        - Response Code: 200 Ok
        - Add an optional string SortByCost parameter. If it equals "ascending" then return the drinks in ascending cost order. If it equals "descending", then return the drinks in descending cost order.
    - GET /Drinks/{id}
        - Response: Drink that matches the provided id
        - Response Code:
            - 200 Ok
            - 404 Not Found
        - Take in int id as a path parameter
    - POST /Drinks
        - Response: The created Drink
        - Response Code: 201 Created
        - Take in Drink from body
    - PUT /Drinks/{id}
        - Response: the updated Drink
        - Response Code: 200 OK
        - Take in int id as a path parameter
        - Take in a Drink though the body

**Extended Challenges:**

- Use the Extended Exercise (API KEYS) SQL below. This will create a User table with ApiKeys. Re-run your EF commands to Scaffold the User table to your project.
    - Create a new Model Called UserDAL
    - In UserDAL, Create a static method called ValidateKey
        - This will take in a string ApiKey