# Definition

The DrinkApi has been implemented so to manage the drink shopping list of a third party.

The API allows to do CRUD operations on the shopping list using the methods it displays. The methods will be discussed further down below.

The requirements of the API are as follows:

The endpoint will work based on the following web requests:

* Http Post request should adding a drink in to the list with quantity e.g. name of drink (Pepsi) and quantity (1)
* Http Put request for updating a drinks quantity
* Http Delete request for removing a drink from the shopping list
* Http Get request for retrieving a drink by its name and its quantity so we can see how many ordered
* Http Get request for retrieving full list of what we have in the shopping list

Notes:

· This doesn’t have to use database you can use in memory solution to hold the shopping list. Anything simple that works

· Ideally it should contain one unique drink name for each entry to avoid repetitions

· Please feel free to implement/explain any advanced features to demonstrate your skills and experience such as paginated lists api authorization and validation etc.

# Technical Implementation:

## Api Authorization

Every request that is made to this webapi has to contain an ApiKey which has been defined and set in the web.config.

Users must add an Authorization key in the header of the request, having value “pk\_test\_2997d616-471e-48a5-ba86-c775ed3ac38a”.

A message handlers has been implemented that intercepts any incoming request, by adding it in the Global.asax, to check if the header contains the Authorization key, if the request does not contain the authorization key, it is blocked and an httpresponse is sent to the requester having HttpStatusCode 403.

Directory of handler: \MessageHandlers\ApiKeyHandler.cs

Example of an HttpResponse for an HttpRequest not containing the Authorization key

## Model Binding and validation

In order to ensure that all the body of request complies with conditions and property of the model describe, the class /Filter/ValidateModelAttribute has been implemented which extends System.Web.Http.Filters.ActionFilterAttribute.

The filter has then been configured in the WebApiConfig.cs, so that a check is made to validate the object being passed to method. If ever the model is not valid, a bad request is sent as HttpResponse to the requester.

Data annotations has also been used to define validations of the models.

Example

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace DrinkApi.Models

{

public class Drink

{

[Required]

public string DrinkName { get; set; }

[Required]

[Range(-50,100)]

public int DrinkCount { get; set; }

}

}

Example of HttpResponse for a request which does not match validation of model:

|  |
| --- |
| Sample Output- JSON |
| {  "Message": "The request is invalid.",  "ModelState": {  "drink": [  "Required property 'DrinkCount' not found in JSON. Path '', line 1, position 24."  ]  }  } |
|  |

# Methods

Most of the methods returns an object of type **Result** defined as follows:

using Newtonsoft.Json;

using Newtonsoft.Json.Converters;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace DrinkApi.Models

{

public class Result

{

[JsonProperty(PropertyName = "Result")]

[JsonConverter(typeof(StringEnumConverter))]

public ResultCode ResultCode { get; set; }

[JsonProperty(NullValueHandling = NullValueHandling.Ignore)]

public string Exception { get; set; }

[JsonProperty(PropertyName = "Drink",NullValueHandling =NullValueHandling.Ignore)]

public Drink drink { get; set; }

[JsonProperty(NullValueHandling = NullValueHandling.Ignore)]

public List<Drink> DrinkList { get; set; }

}

public enum ResultCode

{

OK,

INVALID\_REQUEST,

UNEXPECTED\_ERROR,

FAILED,

NOT\_FOUND,

}

}

## Add Drink Method

The Url for adding a drink is: /drinkApi/addDrink

An object of type Drink should be the body of the Http Post request.

Drink:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace DrinkApi.Models

{

public class Drink

{

[Required]

public string DrinkName { get; set; }

[Required]

[Range(-50,100)]

public int DrinkCount { get; set; }

}

}

### Logic implemented in method:

The number of drink being added should at least be one else a bad request response is obtained.

|  |
| --- |
| Sample Input - JSON |
| {"DrinkName":"d","DrinkCount": 0} |
| Sample Output - JSON |
| {  "Result": "INVALID\_REQUEST",  "Exception": "Drink count should be at least 1"  } |

Also name of drinks are stored in lowercase to avoid duplicate addition of a drink just because of alphabet is uppercase.

Also the add drink method caters if a user is trying to add a drink that is already present in the shopping list. In case a drink is already present, the stock of the drink is updated by adding the quantity passed into the request. That is, if we already have 5 coca-cola drink in the shopping list and a user try to add 5 coca-cola again, then the quantity of coca-cola is updated to 10.

Example of request and response:

|  |
| --- |
| Sample Input - JSON |
| {"DrinkName":"d","DrinkCount": 15} |
| Sample Output - JSON |
| {  "Result": "OK"  } |

## Update Drink method

Url for updating a drink quantity: /drinkApi/UpdateDrinkStock

An object of type **Drink** should be the body of the request.

### Logic implemented in method:

If an Http Put request is made to update the quantity of a drink that is not present in the shopping list, then an appropriate response is sent.

Example:

|  |
| --- |
| Sample Input - JSON |
| {"DrinkName":"-Cola","DrinkCount": -15} |
| Sample Output - JSON |
| {  "Result": "NOT\_FOUND",  "Exception": "Drink not present in shopping list"  } |

Also, this method allows a user to remove X quantity of drink from the shopping list, by passing a negative value for DrinkCount , however we have restricted the quantity of drinks that can be removed per request to 50, and the number of drink that can be added to 100.

While updating the quantity of a drink, if the quantity is below 1, then the drink is removed from the shopping list.

Sample request and response of the method.

|  |
| --- |
| Sample Input - JSON |
| {"DrinkName":"a","DrinkCount": 15} |
| Sample Output - JSON |
| {  "Result": "OK"  } |

## Delete Drink method

Http Delete request Url to delete a drink from the shopping list: /drinkApi/deleteDrink/ {drinkName}

### Logic implemented in method:

The drink name to be deleted is passed in the drink name variable, to avoid any miss-spelling when searching/deleting/updating/creating/deleting, because of upper or lower case, when adding a drink the name has been changed to lower and then saved.

Same logic is applied when deleting a drink from the shopping list, the name is changed to lowercase and then the appropriate drink is deleted.

Example of url to delete drink “Coca-coLa”, which does not exist in the shopping list: /drinkApi/deleteDrink/Coca-coLa

|  |
| --- |
| Sample Output - JSON |
| {  "Result": "NOT\_FOUND",  "Exception": "Drink Not Found !"  } |

Example of url to delete drink “a”, which exist in the shopping list: /drinkApi/deleteDrink/a

|  |
| --- |
| Sample Output - JSON |
| {  "Result": "OK"  } |

## Retrieve drink methods

### Get all drinks in the shopping list

Method URL: /drinkApi/getAllDrink

Request type: Http Get request

|  |
| --- |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": [  {  "DrinkName": "d",  "DrinkCount": 15  },  {  "DrinkName": "a",  "DrinkCount": 91  },  {  "DrinkName": "b",  "DrinkCount": 82  },  {  "DrinkName": "c",  "DrinkCount": 73  }  ]  } |

### Get a specific drink

Method URL: /drinkApi/getDrink/{name}, where {name} is the drink name.

Request type: Http Get request

Example

|  |
| --- |
| Url : /drinkApi/getDrink/a |
| Sample Output - JSON |
| {  "Result": "OK",  "Drink": {  "DrinkName": "a",  "DrinkCount": 91  }  } |

### Custom Search

Method URL: dev.drinkapi.com/drinkApi/CustomSearch

Request type: Http Get request

Mandatory URL parameters: PageSize and PageNumber

Optional URL parameters: Orderby and SortDirection

For the custom search, the search object has been created:

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace DrinkApi.Models

{

public class Search

{

[Required]

[Range(1,500)]

public int PageSize { get; set; }

[Required]

[Range(1,999)]

public int PageNumber { get; set; }

public string OrderBy { get; set; }

public string SortDirection { get; set; }

}

}

Examples:

|  |
| --- |
| URL : /drinkApi/CustomSearch?PageSize=100&PageNumber=1 |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": [  {  "DrinkName": "a",  "DrinkCount": 91  },  {  "DrinkName": "b",  "DrinkCount": 82  },  {  "DrinkName": "c",  "DrinkCount": 73  },  {  "DrinkName": "d",  "DrinkCount": 64  },  {  "DrinkName": "e",  "DrinkCount": 55  },  {  "DrinkName": "f",  "DrinkCount": 46  },  {  "DrinkName": "g",  "DrinkCount": 37  },  {  "DrinkName": "h",  "DrinkCount": 28  },  {  "DrinkName": "i",  "DrinkCount": 19  }  ]  } |

|  |
| --- |
| URL : /CustomSearch?PageSize=5&PageNumber=1&Orderby=drinkname&sortdirection=desc |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": [  {  "DrinkName": "i",  "DrinkCount": 19  },  {  "DrinkName": "h",  "DrinkCount": 28  },  {  "DrinkName": "g",  "DrinkCount": 37  },  {  "DrinkName": "f",  "DrinkCount": 46  },  {  "DrinkName": "e",  "DrinkCount": 55  }  ]  } |

|  |
| --- |
| URL : /CustomSearch?PageSize=5&PageNumber=2&Orderby=drinkname&sortdirection=desc |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": [  {  "DrinkName": "d",  "DrinkCount": 64  },  {  "DrinkName": "c",  "DrinkCount": 73  },  {  "DrinkName": "b",  "DrinkCount": 82  },  {  "DrinkName": "a",  "DrinkCount": 91  }  ]  } |

|  |
| --- |
| URL : /CustomSearch?PageSize=5&PageNumber=2&Orderby=drinkname |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": [  {  "DrinkName": "f",  "DrinkCount": 46  },  {  "DrinkName": "g",  "DrinkCount": 37  },  {  "DrinkName": "h",  "DrinkCount": 28  },  {  "DrinkName": "i",  "DrinkCount": 19  }  ]  } |

|  |
| --- |
| URL : /CustomSearch?PageSize=10&PageNumber=1&Orderby=drinkcount |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": [  {  "DrinkName": "i",  "DrinkCount": 19  },  {  "DrinkName": "h",  "DrinkCount": 28  },  {  "DrinkName": "g",  "DrinkCount": 37  },  {  "DrinkName": "f",  "DrinkCount": 46  },  {  "DrinkName": "e",  "DrinkCount": 55  },  {  "DrinkName": "d",  "DrinkCount": 64  },  {  "DrinkName": "c",  "DrinkCount": 73  },  {  "DrinkName": "b",  "DrinkCount": 82  },  {  "DrinkName": "a",  "DrinkCount": 91  }  ]  } |

|  |
| --- |
| URL : /drinkApi/CustomSearch?PageSize=10&PageNumber=100&Orderby=drinkcount |
| Sample Output - JSON |
| {  "Result": "OK",  "DrinkList": []  } |