PostalProbe - Pincode A.P.I.

1. Get All Pincodes

Method: GET

cURL: https://localhost:44337/api/Pincode/all

Description: List all the pincodes available in the database

Response:

```
[
{
  "id": {
    "officeName": " officeName value1",
    "pincode": pincode value1,
    "district": "district value1",
    "divisionName": " divisionName value1"
  },
  "circleName": " circleName value1",
  "officeType": " officeType value1",
  "delivery": delivery_value1,
  "stateName": " stateName value1"
},
{
  "id": {
    "officeName": " officeName value2",
    "pincode": pincode value2,
    "district": "district value2",
    "divisionName": " divisionName value2"
  },
  "circleName": " circleName value2",
  "officeType": " officeType value2",
  "delivery": delivery_value2,
  "stateName": " stateName value2"
},
```

```
{
  "id": {
    "officeName": " officeName value3",
    "pincode": pincode value3,
    "district": "district value3",
    "divisionName": " divisionName value3"
  },
  "circleName": " circleName value3",
  "officeType": " officeType value3",
  "delivery": delivery_value3,
  "stateName": " stateName value3"
}
.
.
.
{
  "id": {
    "officeName": " officeName value$^n$",
    "pincode": pincode value$^n$,
    "district": "district value$^n$",
    "divisionName": " divisionName value$^n$"
  },
  "circleName": " circleName value$^n$",
  "officeType": " officeType value$^n$",
  "delivery": delivery_value$^n$,
  "stateName": " stateName value$^n$"
}
]
```

2. Search by Pincode

Method: GET

cURL: https://localhost:44337/api/Pincode/**{{pincode}}**

Description:

 - Returns an error message if no rows contain the pincode.

 - Returns the entire row as a JSON object if a single row matches the pincode.

 - Returns an array of office names if multiple rows match the pincode.

Response:

Single Match:

```
{
  "id": {
    "officeName": " officeName value",
    "pincode": pincode value,
    "district": "district value",
    "divisionName": " divisionName value"
  },
  "circleName": " circleName value",
  "officeType": " officeType value",
  "delivery": delivery_value,
  "stateName": " stateName value"
}
```

Multiple Matches:

```
[
"officeName1",
"officeName2",
   .
   .
   .
"officeName$^{n}$"
]
```

3. Search by District

Method: GET

cURL: https://localhost:44337/api/Pincode/search/district/**{{district}}**

Description:

 - Returns an error message if no rows match the district.

 - Returns an array of objects with "Office Name", "Pincode", and "Delivery" values if rows match the district.

Response:

```
[
  {
    "officeName": " officeName value1",
    "pincode": pincode_value1,
    "delivery": delivery_value1
  },
  {
    "officeName": " officeName value2",
    "pincode": pincode_value2,
    "delivery": delivery_value2
  },
  {
    "officeName": " officeName value3",
    "pincode": pincode_value3,
    "delivery": delivery_value1
  },
  .
  .
  .
  {
    "officeName": " officeName value$^n$",
    "pincode": pincode_value$^n$,
    "delivery": delivery_value$^n$
  },
]
```

4. Search by Office Name

Method: GET

cURL: https://localhost:44337/api/Pincode/search/officename/**{{officeName}}**

Description:

- Returns an error message if no rows match the office name.

- Returns the "Pincode" and "Delivery" values if a single row matches the office name.

- Returns an array of objects with "Circle Name", "Division Name", "Pincode", "District", and "StateName" values if multiple rows match the office name.

Response:

Single Match:

```
{
 "pincode": pincode_value,
 "delivery": delivery_value
}
```

Multiple Matches:

```
[
    {
        "CircleName": "CircleName value1",
        "DivisionName": "DivisionName value1",
        "Pincode": Pincode_value1,
        "District": "District value1",
        "StateName": "StateName value1"
    },
    {
        "CircleName": "CircleName value2",
        "DivisionName": "DivisionName value2",
        "Pincode": Pincode_value2,
        "District": "District value2",
        "StateName": "StateName value2"
    },
```

```
            {
                "CircleName": "CircleName value3",
                "DivisionName": "DivisionName value3",
                "Pincode": Pincode_value3,
                "District": "District value3",
                "StateName": "StateName value3"
            },
            .
            .
            .
            {
                "CircleName": "CircleName value$^n$",
                "DivisionName": "DivisionName value$^n$",
                "Pincode": Pincode_value$^n$,
                "District": "District value$^n$",
                "StateName": "StateName value$^n$"
            }
        ]
```

5. Delivery Status for Pincode

Method: GET

cURL: https://localhost:44337/api/Pincode/deliverystatus/pincode/**{{pincode}}**

Description:

 - Returns the "Delivery" value as a message (e.g. "*Delivery is available*" for 'true' and "*Delivery is not available*" for'false') if a single record matches the pincode.

 - Returns an array of objects with "Office Name" and "Delivery" values if multiple records match the pincode.

Response:

Single Match:

```
{
  "message": "Delivery is available"
}
```

Multiple Matches:

```
[
  {
    "officeName": "officeName value1",
    "message": "message1"
  },
  {
    "officeName": " officeName value2",
    "message": "message2"
  },
  .
  .
  .
]
```

6. Delivery Status for Office Name

Method: GET

cURL: https://localhost:44337/api/Pincode/deliverystatus/officename/**{{officeName}}**

Response:

```
{
  "message": "Delivery is available"
}
```

7. Delivery Status for Primary Key

Method: GET

cURL:
https://localhost:44337/api/Pincode/deliverystatus/**{{officeName}}**/**{{pincode}}**/**{{district}}**/**{{divisionName}}**

Description: User provides the complete composite primary key in the request and receives the delivery status in the response.

Response:

```
{
  "message": "Delivery is available"
}
```

8. Add Pincode Record

Method: POST

cURL: https://localhost:44337/api/Pincode/add

Body:

```
{
  "id": {
    "officeName": "string",
    "pincode": 0,
    "district": "string",
    "divisionName": "string"
  },
  "circleName": "string",
  "officeType": "string",
  "delivery": true,
  "stateName": "string"
}
```

Description: Add a new pincode record (Basic Authentication Required)

Response:

Authorised User: "Pincode added successfully"

Unauthorized User: "You need to provide valid credentials to access this resource."

9. Update Pincode Details

Method: PUT

cURL:
https://localhost:44337/api/Pincode/**{{officeName}}**/**{{pincode}}**/**{{district}}**/**{{divisionName}}**

Body:

```
{
  "id": {
    "officeName": "string",
    "pincode": 0,
    "district": "string",
    "divisionName": "string"
  },
  "circleName": "string",
  "officeType": "string",
  "delivery": true,
  "stateName": "string"
}
```

Description: All details of the pincode record can be updated except for the pincode. (Basic Auth Required)

Response:

Authorised User: `"Pincode details updated successfully"`

Unauthorized User: "`You need to provide valid credentials to access this resource.`"

10. Delete Pincode Record

Method: DELETE

cURL:
https://localhost:44337/api/Pincode/**{{officeName}}**/**{{pincode}}**/**{{district}}**/**{{divisionName}}**

Description: Delete an existing Pincode record.

Response:

 Authorised User: `"Pincode deleted successfully"`

 Unauthorized User: "You need to provide valid credentials to access this resource."

11. Offices In a District

Method: GET

cURL: https://localhost:44337/api/Pincode/officetype/district/**{{district}}**/**{{officeType}}**

Description: Pass the District and Office Type values in the request and get a list of office names with the specific office type in the particular district.

Response:
```
[
    "officeName1",
    "officeName2",
     "officeName3"
  .
  .
  .
  "officeNameⁿ"
]
```

12. Offices In a Division

Method: GET

cURL:
https://localhost:44337/api/Pincode/officetype/division/**{{divisionName}}**/**{{officeType}}**

Description: Pass the Division and Office Type values in the request and get a list of office names with the specific office type in the particular division.

Response:

```
[

    "officeName1",

    "officeName2",

     "officeName3"

  .

  .

  .

  "officeNameⁿ"

]
```

13. Get Office Type for an Office Name

Method: GET

cURL: https://localhost:44337/api/Pincode/officetype/officename/**{{officeName}}**

Description: Pass the Office Name value in the request and get the office type for the corresponding office name.

Response:

```
"officeType"
```

Variable	Description
pincode	6 digit number
district	Name of the district
officeName	Name of the OfficeName
divisionName	Name of the DivisionName
officeType	Name of the OfficeType (e.g. B.O, S.O, H.O)
username	Username used for Authentication(Basic Auth)
password	Password used for Authentication(Basic Auth)