

## Web Service REST API Endpoints

Recipe Web Service comes with ReST API Ends points for authentication, creating a new recipe, retrieving an existing recipe, retrieving all existing recipes as list, updating an an existing recipe and deleting an existing recipe. Below table lists and describes on the implemented ReST APIs: Note: With all given below api end points request, make sure to include header **Content-Type as application/json**

| API End Point     | Method | Purpose                        | Request  | Response   |
|-------------------|--------|--------------------------------|--|--|
| /api/authenticate | POST   | Authenticate and get JWT Token | User Model with user name and password   | JWT Token on Success, 403 Forbidden on failure                       |
| /api/recipe       | POST   | Create a new recipe            | Recipe Model and valid JWT Token as bearer token as auth header                | Recipe Model with 201 Created on Success, 400 Bad request on failure |
| /api/recipe/{id}  | GET    | Get an existing recipe         | Recipe id as path parameter and valid JWT Token as bearer token as auth header | Recipe Model with 200 OK on Success, 401 Not Found on failure        |

|                  |        |                                  |  |   |
|------------------|--------|----------------------------------|--|---|
| /api/recipes     | GET    | Get all existing recipes as list | Valid JWT Token as bearer token as auth header                                 | Recipes as list with 200 OK on success, 401 Not Found on failure  |
| /api/recipe      | PUT    | Update an existing recipe        | Updated Recipe Model and valid JWT Token as bearer token as auth header        | Recipe Model with 200 OK on Success, 401 Not Found on failure     |
| /api/recipe/{id} | DELETE | Delete an existing recipe        | Recipe id as path parameter and valid JWT Token as bearer token as auth header | Deletion message with 200 OK on success, 401 Not Found on failure |

## Web Service ReST Endpoints Usage and Sample Response

In order to consume Recipe Webservice ReST API End points, one has to first authenticate and get JWT Token in order to place subsequent client requests. There are 2 object models needed to be aware of - one for Authentication and the other one for Recipe contents. Below given are details and examples on needed models:

- User Model

### JSON Schema

```
{
  "userName": "Username as configured for recipe web service",
  "password": "JWT Secret Key as configured for recipe web service"
```

```
}
```

## JSON Example

```
{
  "userName": "abnamro",
  "password": "recipeKey"
}
```

- Recipe Model

## JSON Schema

```
{
  "id": "recipeId as integer value",
  "name": "recipeName as string",
  "type": "recipeType - ng/vg/eg as string",
  "servingCapacity": "number of people the dish to be served as
integer value",
  "creationDateTime": "creation date time as Date or null",
  "ingredients": "list of ingredients objects with name and
quantity as fields or null",
  "instructions": "step by step procedure to prepare recipe as
text or null"
}
```

## JSON Example

```
{
  "id": 101,
  "name": "Sweet Bun",
  "type": "vg",
  "servingCapacity": 5,
  "creationDateTime": null,
  "ingredients": [{
    "name": "all purpose floor",
    "quantity": "500 gms"
  }, {
    "name": "yeast",
    "quantity": "5 gms"
  }, {
    "name": "sugar",
    "quantity": "10 gms"
  }, {
```

```

        "name": "milk",
        "quantity": "100 ml"
    }],
    "instructions": "1.Activate the yeast by with milk and sugar
\n 2.Take all purpose floor in mixing bowl and mix with sugar and
milk \n 3.Mix yeast with floor in bowl\n 4.Wait for few minutes to
raise and then put inti oven for about 20-25 minutes at 180 degree
temperatur"
}

```

- **ReST API Calls and responses**

**POST request to /api/authenticate end point with above given user model :**

```

{
    "userName": "abnamro",
    "password": "Bearer header.payload.signature"
}

```

- **POST request to /api/recipe end point with sample recipe model and JWT Token as auth header returned in call to /api/authenticate :**

```

{
    "id": 102,
    "name": "White Forest Pastry",
    "type": "eg",
    "servingCapacity": 10,
    "ingredientsList": [
        {
            "name": "eggs",
            "quantity": "10 nos"
        },
        {
            "name": "floor",
            "quantity": "500 gms"
        },
        {
            "name": "vippping",
            "quantity": "250 gms"
        }
    ],
    "creationDateTime": "2021-07-01T15:21:34.561+00:00",
}

```

```

        "instructions": "Step by Step procedure to prepare Butter  
Sponge Cake:\nStep-1:\nStep-2:\nStep-3:\nStep-4:\nStep-5:",
        "cdateTimeString": "01-07-2021 20:51:34"
    }

```

- GET request to `/api/recipe/{id}` end point with path parameter as recipe id:  
`/api/recipe/102` end point :

```

{
    "id": 102,
    "name": "White Forest Pastry",
    "type": "eg",
    "servingCapacity": 10,
    "ingredientsList": [
        {
            "name": "eggs",
            "quantity": "10 nos"
        },
        {
            "name": "floor",
            "quantity": "500 gms"
        },
        {
            "name": "vipping",
            "quantity": "250 gms"
        }
    ],
    "creationDateTime": "2021-07-01T15:21:34.561+00:00",
    "instructions": "Step by Step procedure to prepare Butter  
Sponge Cake:\nStep-1:\nStep-2:\nStep-3:\nStep-4:\nStep-5:",
    "cdateTimeString": "01-07-2021 20:51:34"
}

```

- GET request to `/api/recipes` end point :

```

[
    {
        "id": 101,
        "name": "Sweet Bun",
        "type": "vg",
        "servingCapacity": 5,
        "ingredientsList": [],
        "creationDateTime": "2021-07-01T15:19:27.809+00:00",
        "instructions": null,
    }
]

```

```

        "cdateTimeString": "01-07-2021 20:49:27"
    },
    {
        "id": 102,
        "name": "White Forest Pastry",
        "type": "eg",
        "servingCapacity": 10,
        "ingredientsList": [
            {
                "name": "eggs",
                "quantity": "10 nos"
            },
            {
                "name": "floor",
                "quantity": "500 gms"
            },
            {
                "name": "vipping",
                "quantity": "250 gms"
            }
        ],
        "creationDateTime": "2021-07-01T15:21:34.561+00:00",
        "instructions": "Step by Step procedure to prepare
Butter Sponge Cake:\nStep-1:\nStep-2:\nStep-3:\nStep-4:\nStep-5:",
        "cdateTimeString": "01-07-2021 20:51:34"
    }
]

```

- PUT request to `/api/recipe` end point with updated model with recide id 102 and name renamed to `Sponge Cake` and `servingCapacity` to 4 :

```

{
    "id": 102,
    "name": "Sponge Cake",
    "type": "eg",
    "servingCapacity": 4,
    "ingredientsList": [
        {
            "name": "eggs",
            "quantity": "10 nos"
        },
        {
            "name": "floor",
            "quantity": "500 gms"
        },
    ],
}

```

```

        {
            "name": "vipping",
            "quantity": "250 gms"
        },
        "creationDateTime": "2021-06-30T15:10:59.642+00:00",
        "instructions": "Step by Step procedure to prepare
Butter Sponge Cake:\nStep-1:\nStep-2:\nStep-3:\nStep-4:\nStep-5:",
        "cdateTimeString": "30-06-2021 20:40:59"
    }

```

- GET request to `/api/recipe/{id}` end point with path parameter as recipe id: `/api/recipe/102`, one should notice updated fields in previous request reflected, i.e. recipe name and serving capacity:

```

    {
        "id": 102,
        "name": "Sponge Cake",
        "type": "eg",
        "servingCapacity": 4,
        "ingredientsList": [
            {
                "name": "eggs",
                "quantity": "10 nos"
            },
            {
                "name": "floor",
                "quantity": "500 gms"
            },
            {
                "name": "vipping",
                "quantity": "250 gms"
            }
        ],
        "creationDateTime": "2021-06-30T15:10:59.642+00:00",
        "instructions": "Step by Step procedure to prepare
Butter Sponge Cake:\nStep-1:\nStep-2:\nStep-3:\nStep-4:\nStep-5:",
        "cdateTimeString": "30-06-2021 20:40:59"
    }

```

- DELETE request to `/api/recipe/{id}` end point with path parameter as recipe id: `/api/recipe/101`, one should notice response message and status code as 200 OK :Requested recipe deleted from DB

- GET request to `/api/recipe/{id}` end point with path parameter as recipe id: `/api/recipe/101`, one should notice response message and status code as 401 Not Found :

```

{
  "status": 404,
  "message": "404 NOT_FOUND \ "Requested recipe not found in
DB\\"",
  "dateTime": "2021-07-01T21:08:53.797"
}

```

- GET request to `/api/recipes` end point should return only one recipe with id 102:

```

[
  {
    "id": 102,
    "name": "Sponge Cake",
    "type": "eg",
    "servingCapacity": 4,
    "ingredientsList": [
      {
        "name": "eggs",
        "quantity": "10 nos"
      },
      {
        "name": "floor",
        "quantity": "500 gms"
      },
      {
        "name": "vipping",
        "quantity": "250 gms"
      }
    ],
    "creationDateTime": "2021-06-30T15:10:59.642+00:00",
    "instructions": "Step by Step procedure to prepare Butter Sponge
Cake:\nStep-1:\nStep-2:\nStep-3:\nStep-4:\nStep-5:",
    "cdateTimeString": "30-06-2021 20:40:59"
  }
]

```



## Future Enhancements

- Integrate Web Service with Authorisation server for authentication and authorisation
- Design and Implementation of Multi Factor Authentication feature
- Design and build simple,beautiful front end view
- Integration between Backend API with Front End view
- Data Normalization in Persistence layer if recipes data grows exponentially