

# Implementing an API in ASP.NET Web API

## Module 2: API Basics

Shawn Wildermuth  
Wilder Minds LLC  
<http://wilder minds.com>



**pluralsight**  
hardcore developer training

# Agenda

- **API Basics**

- What Are We Building?
- Identifiers
- Implementing an Association
- Preparing for Security
- Implementing POST
- Implementing DELETE
- Implementing PUT/PATCH
- Implementing Paging
- Implementing RPC-Style Calls

# API Design: Nutrition

- **GET /api/nutrition/foods**
  - Gets a list of foods
    - Params: includeMeasures=false, page=0
- **GET /api/nutrition/foods/1**
  - Get all the information on a specific food
- **GET /api/nutrition/foods/1/measures**
  - Get a list of measures for a specific food
- **GET /api/nutrition/foods/1/measures/1**
  - Get a specific measure for a food

# API Design: User Diary

- **GET /api/user/diaries**
  - Get a list of diaries for the current user
- **POST /api/user/diaries**
  - Create a new diary for a specific date
- **GET /api/user/diaries/2001-01-01**
  - Get a specific daily diary
- **DELETE /api/user/diaries/2001-01-01**
  - Delete an entire diary
- **GET /api/user/diaries/2001-01-01/summary**
  - Get a summary of the nutritional content for a single diary

# API Design: Diary Entries

- **GET /api/user/diaries/2001-01-01/entries**
  - Get the entries for a diary
- **POST /api/user/diaries/2001-01-01/entries**
  - Add a new entry for a diary
- **GET /api/user/diaries/2001-01-01/entries/1**
  - Gets an individual diary entry
- **PATCH /api/user/diaries/2001-01-01/entries/1**
  - Update a diary entry
- **DELETE /api/user/diaries/2001-01-01/entries/1**
  - Delete a diary entry

## **Implementing an Association**

**Identifiers**

## **Preparing for Security**



# **Implementing POST**

## **Implementing DELETE**

## **Implementing PUT/PATCH**

# **Implementing Paging**

## **Implementing RPC-Style Calls**

# Summary

## ■ API Basics

- Web API allows you to build a variety of REST-ful scenarios
- Associations are solved with routes for specific related entities
- Implementing IDs and URLs can simplify the usage of your API
- Completing the API using HTTP Verbs on ApiController is straightforward
- Dealing with Paging for large data scenarios is as simple as your back-end