# Open-Australia API Integration for ASP.NET

## Base Configuration

public class OpenAustraliaConfig

{

public const string BaseUrl = "http://www.openaustralia.org/api";

public const string DefaultOutput = "php" *or "js" for JSON*

}

# API Endpoints and Implementation

# 1. Representatives API

## Endpoints:

* /getRepresentative
* /getRepresentatives

public async Task<IActionResult> GetRepresentative(string id = null, string division = null, bool? alwaysReturn = null)

{

using (var client = new HttpClient())

{

var query = new Dictionary<string, string>

{

["output"] = OpenAustraliaConfig.DefaultOutput

};

if (!string.IsNullOrEmpty(id)) query.Add("id", id);

else if (!string.IsNullOrEmpty(division))

{

query.Add("division", division);

if (alwaysReturn.HasValue && alwaysReturn.Value)

query.Add("always\_return", "1");

}

var response = await client.GetAsync(QueryHelpers.AddQueryString(

$"{OpenAustraliaConfig.BaseUrl}/getRepresentative",

query));

return ProcessResponse(response);

}

}

public async Task<IActionResult> GetRepresentatives(

string postcode = null,

string date = null,

string party = null,

string search = null)

{

using (var client = new HttpClient())

{

var query = new Dictionary<string, string>

{

["output"] = OpenAustraliaConfig.DefaultOutput

};

if (!string.IsNullOrEmpty(postcode)) query.Add("postcode", postcode);

if (!string.IsNullOrEmpty(date)) query.Add("date", date);

if (!string.IsNullOrEmpty(party)) query.Add("party", party);

if (!string.IsNullOrEmpty(search)) query.Add("search", search);

var response = await client.GetAsync(QueryHelpers.AddQueryString(

$"{OpenAustraliaConfig.BaseUrl}/getRepresentatives",

query));

return ProcessResponse(response);

}

}

This is a C# implementation of a client for accessing the OpenAustralia API, which provides data about Australian political representatives.

The code has two main parts:

1. A configuration class (OpenAustraliaConfig) that sets up the base URL for the API and specifies the default output format (PHP or JSON).
2. Two main API endpoint methods:
   * GetRepresentative: Fetches data about a single representative, either by their ID or by their division (electoral area)
   * GetRepresentatives: Retrieves multiple representatives, which can be filtered by postcode, date, party, or a search term

Both methods work by constructing HTTP GET requests with the appropriate query parameters and sending them to the OpenAustralia API. The code uses HttpClient to make these web requests and returns the API responses.

In essence, this is a wrapper that makes it easier for developers to access Australian parliamentary representative data through the OpenAustralia service.