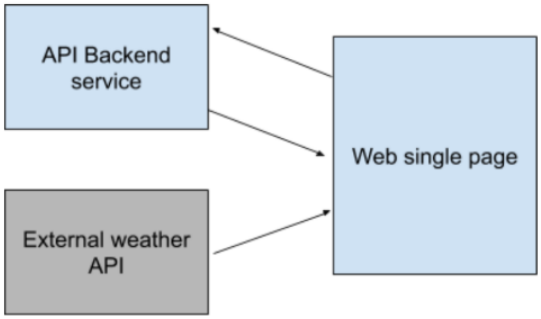
1 INDEX

2 INTRODUCTION

This document aims to expose the main aspects of the **Event Catalogue** backend service and its architecture and design basis.



The API backend service uses the ASP.NET framework to build a web service together with .NET C# in combination with HTML/JavaScript. The server-side logic uses C# and a Service Oriented Architecture (SOA). On the other hand, the web page uses HTML and JavaScript languages.

3 BACKEND DESCRIPTION

As already mentioned, the backend service uses C# and a SOA in order to build an Application Programming Interface (API) that can be used by clients and pages to enable bi-directional communication between server and client in real-time.

The server-side application is formed by means of several assemblies that allow máximum flexibility and scability of the software.

3.1 Application

This assembly contains the services and managers that carry out most of the server operations. In this case, the core **EventCatalogueManager** which handles events and their related operations:

- Add event.
- Filter events.
- Update event.
- Delete event.

3.2 Domain

This assembly contains the main interfaces or foundations used by the software in order to build appropiated services according on-demand. In this case, it contains the **IEventManager** which holds the basic structure that an **EventCatalogueManager** needs to keep in order to be functional.

Another reason for the **Domain** to exist is to facilitate injection of dependencies and keep application and domain isolated from each other.

3.3 Shared

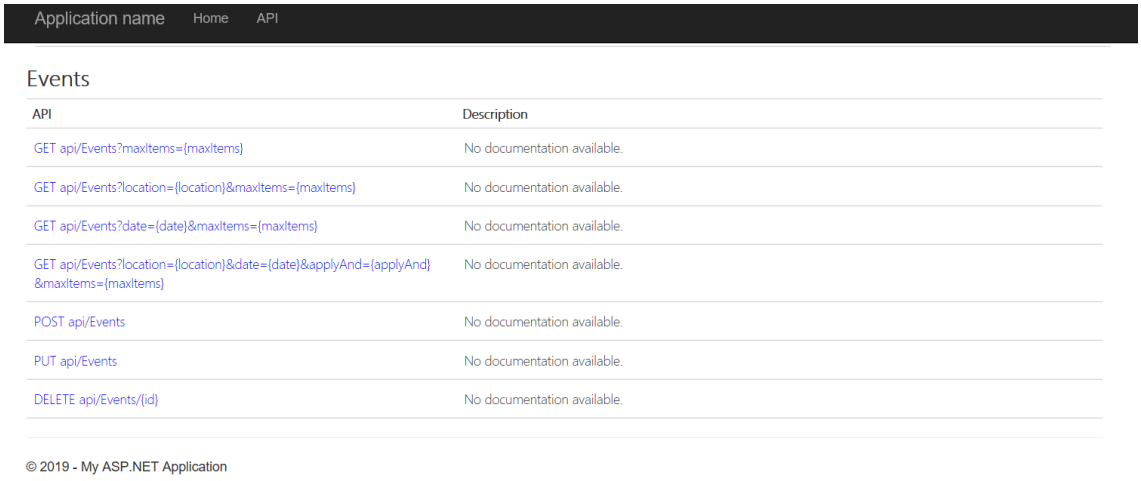
This assembly contains all those classes or objects that are used across different assemblies to communicate or share information. This is the reason why the **EventInfo** class lies in this assembly. This assembly ensures access to serializable classes that can be exchanged across the application.

3.4 Web

The orchestration assembly that allows to build up the web API and also coordinates all the other components. It is in charge of loading all necessary objects by means of dependency injection so the web api is available to be used.

4 FRONTEND DESCRIPTION

Once the server-side application is up and running. HTML/JavaScript pages can access the content of the **EventCatalogue** by means of the API methods and their corresponding addresses.



The webpage is available in http://{server_address}:{iis_port}/Home/EventsHome.
In the solution the home page for the events is available in:
EventCatalogue.Web\Views\Home\EventsHome.cshtml.

4.1 Functionality

The web page can add, update and display events.

4.1.1 Display / Search by location

By clicking in "Search" it will display all events (with a limitation of data to show).
If a location is entered then it will display events in certain location.

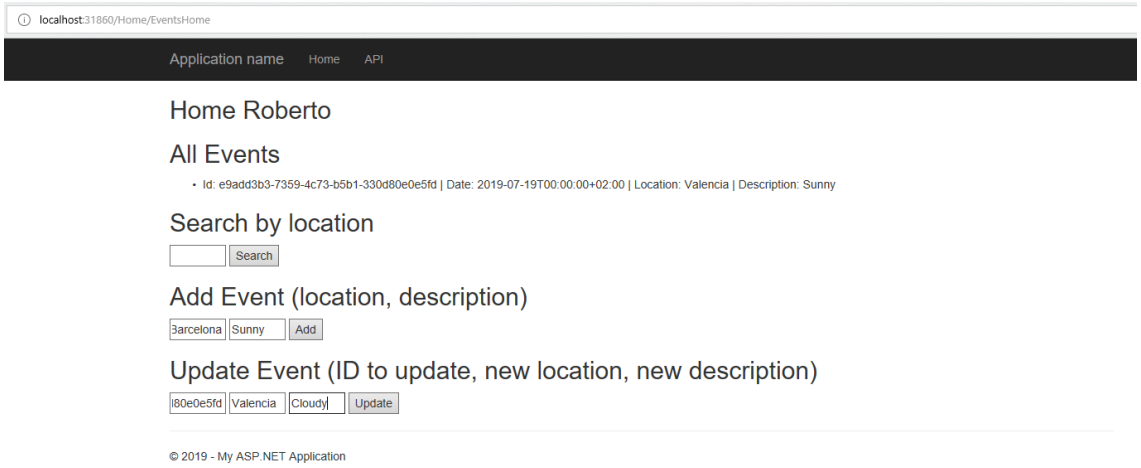
4.1.2 Add event

It allows user to ad dan event into the server.

4.1.3 Update event

Given the event ID it allows the user to update certain event.

4.2 Content



The content of EventsHome.cshtml:

```
@model EventCatalogue.Shared.Classes.EventInfo

@{
    ViewBag.Title = "EventsHome";
}

<h2>Home Roberto</h2>
<body>
    <div>
        <h2>All Events</h2>
        <ul id="events" />
    </div>
    <div>
        <h2>Search by location</h2>
        <input type="text" id="eventLocation" size="5" />
        <input type="button" value="Search" onclick="find();" />
        <p id="product" />
    </div>
    <div>
        <h2>Add Event (location, description)</h2>
        <input type="text" id="location" size="5" />
        <input type="text" id="description" size="5" />
        <input type="button" value="Add" onclick="add();" />
    </div>
    <div>
        <h2>Update Event (ID to update, new location, new description)</h2>
        <input type="text" id="update_id" size="5" />
        <input type="text" id="update_location" size="5" />
        <input type="text" id="update_description" size="5" />
        <input type="button" value="Update" onclick="update();" />
    </div>

    <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
    <script>
        var apiUrl = 'http://localhost:31860/api/Events?';

        $(document).ready(function () {
            // Send an AJAX request
            $.getJSON(apiUrl + 'maxItems=100')
                .done(function (data) {
                    // On success, 'data' contains a list of products.
```

```
                    $.each(data, function (key, item) {
                        // Add a list item for the product.
                        $('<li>', { text:
formatItem(item) }).appendTo($('#events'));
                    });
                });
        });

        function formatItem(item) {
            return 'Id: ' + item.Id + ' | Date: ' + item.DateCreated + ' |
Location: ' + item.Location + ' | Description: ' + item.Description;
        }



        function find() {
            var id = $('#eventLocation').val();

            $('#events').empty();

            $.getJSON(apiUrl + 'location=' + id + '&maxItems=100')
                .done(function (data) {
                    // On success, 'data' contains a list of products.
                    $.each(data, function (key, item) {
                        // Add a list item for the product.
                        $('<li>', { text:
formatItem(item) }).appendTo($('#events'));
                    });
                });

        }

        function add() {
            var location = $('#location').val();
            var description = $('#description').val();

            $(function () {
                var event = { Id: "00000000-0000-0000-0000-000000000000",
DateCreated: "2019-07-19T11:01:37.5924672+02:00", Location: location, Description:
description };
                $.ajax({
                    type: "POST",
                    data: JSON.stringify(event),
                    url: "http://localhost:31860/api/Events",
                    contentType: "application/json"
                });
            });
        }

        function update() {
            var updateId = $('#update_id').val();
            var updateLocation = $('#update_location').val();
            var updateDescription = $('#update_description').val();

            $(function () {
                var event = { Id: updateId, DateCreated: "2019-07-
19T11:01:37.5924672+02:00", Location: updateLocation, Description:
updateDescription };
                $.ajax({
                    type: "PUT",
                    data: JSON.stringify(event),
                    url: "http://localhost:31860/api/Events",
```

```
                    contentType: "application/json"
                });
            });
        }

    </script>

</body>
```