EVENTS OCCUR IN REAL TIME

FLUTTER BELGIUM

TIM GEYSSENS
SENIOR DEV/TECH LEAD
AALTRA.EU

NEWBIE FLUTTER DEV

WEB BACKGROUND UMBRACO CMS

REAL-TIME FUNCTIONALITY IS/WAS HARD

ADDING REAL-TIME FUNCTIONALITY IS AS EASY AS BREAKING A STYROFOAM BOARD.

Tim Geyssens

MEET SIGNALR

SIGNALR

- Created by David Fowler and Damian Edwards in **July 2011**. It started as an open-source project and attracted other developers to contribute to it. It was later adopted and released as **part of the ASP.NET project in 2013**.
- Allows server code to send **asynchronous notifications to client-side** web applications. The library includes server-side and client-side JavaScript components.
- Takes advantage of **WebSocket**, an HTML5 API that enables bi-directional communication between the browser and server. SignalR will use WebSockets under the covers when it's available, and **gracefully fall back** to other techniques and technologies when it isn't, while the application code remains the same.

SIGNALR

- .NET
- Open source
- Cross platform
- Abstraction layer on top of WebSockets, making it easier
- Introduces the concept of hubs

PERSISTANT, BIDIRECTIONAL COMMUNICATION

USE CASES

- Live chat. 1-to-1 chat, chat rooms, chat bots, support chat, in-game chat.
- Data broadcast. High-frequency updates sent in a 1-to-many fashion, such as live scores and traffic updates, live captions, voting, polling, and auction apps.
- Multiplayer collaboration. Whiteboards and co-authoring apps (such as Google Docs).
- Notifications across social media and chat apps, online marketplaces, gaming, travel and transportation, etc.
- GPS location tracking for logistics, fleet management, urban mobility, and food delivery apps.

TURBO POLYP RESERVATION APP

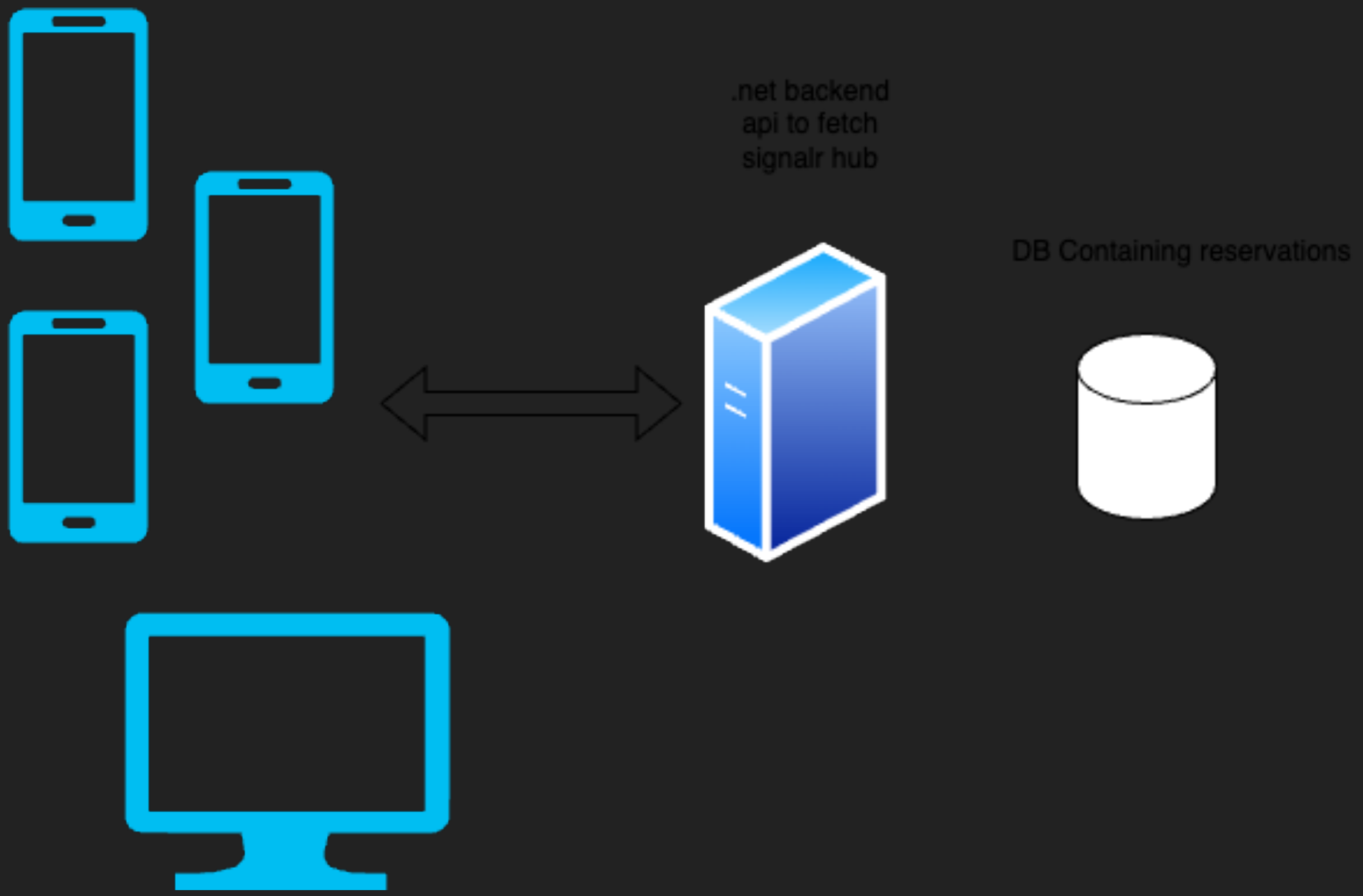
TURBO POLYP?

TURBO POLYP

- Maikel : https://www.youtube.com/watch?v=PQzOrjpXyNs
- Joury: https://www.youtube.com/watch?v=xb0-7XOYDxg

TURBO POLYP RESERVATION APP

- Overview of all seats
- Limited amount available per day
- Once a seat is taken it must be unavailable for others immediately


.net backend
api to fetch
signalr hub
DB Containing reservations

BACKEND

BACKEND

DB TABLE

Results | Messages

	Id	Number	TimeSlot	Taken
1	A09A1B12-7B7A-45B4-BFAB-00AF1CBFCFE2	13	2024-03-12 20:00:00.000	1
2	C766D068-7C97-4541-91E3-00C78E44C9E1	16	2024-03-11 20:00:00.000	1
3	A9F43A83-8D11-4F92-A067-03DEF21C707B	15	2024-03-12 20:00:00.000	1
4	91BB9A63-A5C5-48A8-9A99-06C33CE028B3	7	2024-03-11 22:00:00.000	0
5	DA7060B8-B67D-42C1-9F80-0C1C56346AC0	3	2024-03-11 20:00:00.000	0
6	D3BD544E-777D-469C-84BB-0CD0E3A03595	10	2024-03-12 21:00:00.000	0
7	40328837-61D3-4324-92AB-0F46BB4C9F3B	10	2024-03-11 22:00:00.000	0
8	D5A86A1B-C82F-432C-8BCC-0FA2999232DD	7	2024-03-11 19:00:00.000	1
9	DA778001-F277-40FA-B9D8-107092D31FE3	14	2024-03-11 18:00:00.000	1
10	9DB12D76-08B1-4A85-A780-1100FC6A08E7	1	2024-03-12 22:00:00.000	0
11	791FC4E0-45A1-4209-BE7F-11E51132C48B	9	2024-03-11 22:00:00.000	0
12	CADB55CC-D22D-47EB-A121-139C1469D616	10	2024-03-11 21:00:00.000	0
13	34962B75-32C7-47CC-A3D7-19B08CAB8A1C	14	2024-03-11 21:00:00.000	0
14	88B84191-54A6-4ED0-8978-1C60052604B3	13	2024-03-12 22:00:00.000	0
15	FC16C318-12C3-4868-A0F0-22EBCFBEDDAF	8	2024-03-12 20:00:00.000	0
16	[illegible]	5	[illegible]	0

BACKEND

MODEL

```
namespace TurboPolyp.Models
{
    9 references
    public class Seat
    {
        3 references
        public Guid Id { get; set; }
        0 references
        public int Number { get; set; }
        3 references
        public DateTime TimeSlot { get; set; }
        3 references
        public bool Taken { get; set; }
    }
}
```

BACKEND

REPO

```
public class SeatRepository
{
    private readonly QueryFactory _queryFactory;

    0 references
    public SeatRepository(QueryFactory queryFactory)
    {
        _queryFactory = queryFactory;
    }

    2 references
    public IEnumerable<Seat> GetTodaysSeats()
    {
        var todaysSeats = _queryFactory.Query("seat")
            .WhereBetween("timeslot", DateTime.Today, DateTime.Today.AddHours(23))
            .Get<Seat>();
        if (todaysSeats.Any())
            return todaysSeats;
        else return InsertTodaysSeats();
    }

    2 references
    public void ReserveSeat(Guid id)
    {
        _queryFactory.Query("seat").Where("id", id).Update(new
```

BACKEND

API

```
using Microsoft.AspNetCore.Mvc;
using TurboPolyp.Models;
using TurboPolyp.Repositories;

namespace TurboPolyp.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class SeatController : ControllerBase
    {
        private readonly SeatRepository _seatRepository;
        0 references
        public SeatController(SeatRepository repo)
        {
            _seatRepository = repo;

        }

        [HttpGet]
        0 references
        public IEnumerable<Seat> Get()
        {
            return _seatRepository.GetTodaysSeats();
        }

        [HttpPost]
        0 references
        public bool ReserveSeat(string id)
        {
            _seatRepository.ReserveSeat(Guid.Parse(id));
            return true;
        }
    }
}
```

BACKEND

SIGNALR HUB

```
using Microsoft.AspNetCore.SignalR;
using TurboPolyp.Repositories;

namespace TurboPolyp
{
    2 references
    public class SeatHub: Hub
    {
        private readonly SeatRepository _seatRepository;
        0 references
        public SeatHub(SeatRepository seatRepository)
        {
            _seatRepository = seatRepository;
        }
        0 references
        public async Task ReserveSeat(string seatId)
        {
            _seatRepository.ReserveSeat(Guid.Parse(seatId));

            await Clients.All.SendAsync("BlockSeat", seatId);
        }
    }
}
```

BACKEND

MAP HUB

```
app.MapHub<SeatHub>("/seatHub");

app.Run();
```

WEB APP

WEB APP

CONNECT

```
var connection = new signalR.HubConnectionBuilder().withUrl("/seatHub").build();

connection.start().then(function () {

}).catch(function (err) {
    return console.error(err.toString());
});
```

WEB APP

INVOKE

```
document.getElementById("seats").addEventListener("click", fun
    if (e.target && e.target.parentNode.parentNode.matches("li
        var seatId = e.target.parentNode.parentNode.id;
        connection.invoke("ReserveSeat", seatId).catch(functio
            return console.error(err.toString());
        });
        alert("Seat reserved, ok let's go!");
        event.preventDefault();
    }
});
```

WEB APP

LISTEN

```
connection.on("BlockSeat", function (seatId) {

    document.getElementById(seatId).classList.add("taken");
});
```

DEMO

FLUTTER APP

FLUTTER APP

PACKAGE

signalr_netcore 1.3.7

Published 36 days ago Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

150

Readme Changelog Example Installing Versions Scores

signalr_client

pub v1.3.7

A Flutter SignalR Client for ASP.NET Core.
ASP.NET Core SignalR is an open-source library that simplifies adding real-time web functionality to apps. Real-time web functionality enables server-side code to push content to clients instantly.

Tested with ASP.NET Core 3.1 & ASP .NET Core 6

The client is able to invoke server side hub functions (including streaming functions) and to receive method invocations issued by the server. It also supports the auto-reconnect feature.

The client supports the following transport protocols:

- WebSocket
- Service Side Events
- Long Polling

```yaml
dependencies:
  flutter:
    sdk: flutter

  signalr_netcore: ^1.3.7
```

FLUTTER APP

CONNECT

```dart
import 'package:signalr_netcore/signalr_client.dart';
// The location of the SignalR Server.
const  serverUrl = 'https://60c6-20-16-76-181.ngrok-free.app';

final hubConnection = HubConnectionBuilder().withUrl('$serverUrl/seatHub').build();
```

```
  @override
  Codiumate: Options | Test this method
  void initState() {
    super.initState();
    _startHubConnection();

  }

Codiumate: Options | Test this method
void _startHubConnection()  {
  try {
    hubConnection.start();
    print('SignalR connection started.');
  } catch (e) {
    print('Error starting SignalR connection: $e');
  }

}
```

FLUTTER APP

INVOKE

```dart
void _reserveSeat(String seatId) async{
  try {
   await hubConnection.invoke('ReserveSeat', args: [seatId]);
    print('ReserveSeat invoked');
  } catch (e) {
    print('Error invoking reserving seat: $e');
  }
}
```

FLUTTER APP

LISTEN

```dart
@override
Codiumate: Options | Test this method
void initState() {
  super.initState();
  _startHubConnection();
  _fetchSeats();
  hubConnection.on("BlockSeat", _handleSeatReservation);

}

Codiumate: Options | Test this method
void _handleSeatReservation(List<Object?>? parameters) {
  print('Handling seat reservation broadcast');
  String seatId = parameters![0] as String;

  seats.where((element) => element.id == seatId).first;
```

DEMO

ADDING REAL-TIME FUNCTIONALITY IS AS EASY AS BREAKING A STYROFOAM BOARD.

Tim Geyssens

GITHUB.COM/AALTRA/
EVENTSOCCURINREALTIME