November 9-11, 2021

www.dotnetconf.net

.NET Conf

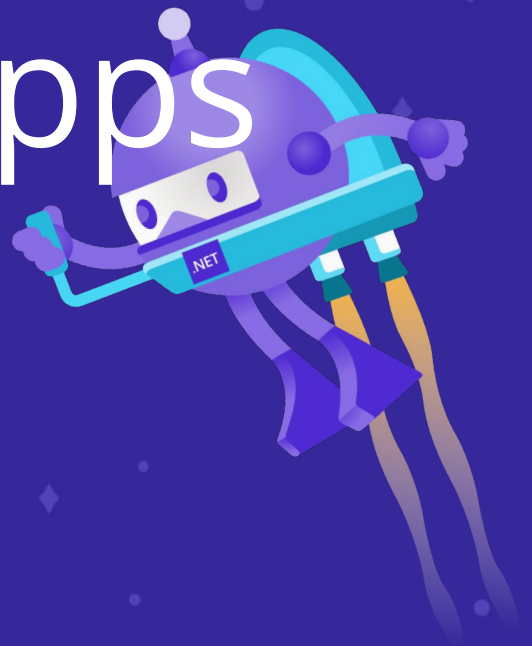
Discover the world of .NET

Raspberry-Pi hand sanitizer controlled by Mobile Apps

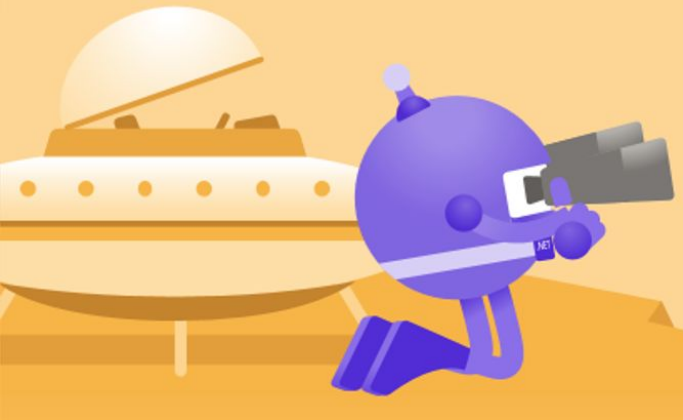
Saamer Mansoor

Founder, The First Prototype

11/8/2021



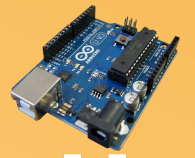
Introduction



Me

Omar A.

Arduino



EpicU

C#

Exigo

Xamarin

Hackathon

James Q

Windsor

Inspiration

Xamarin Basics

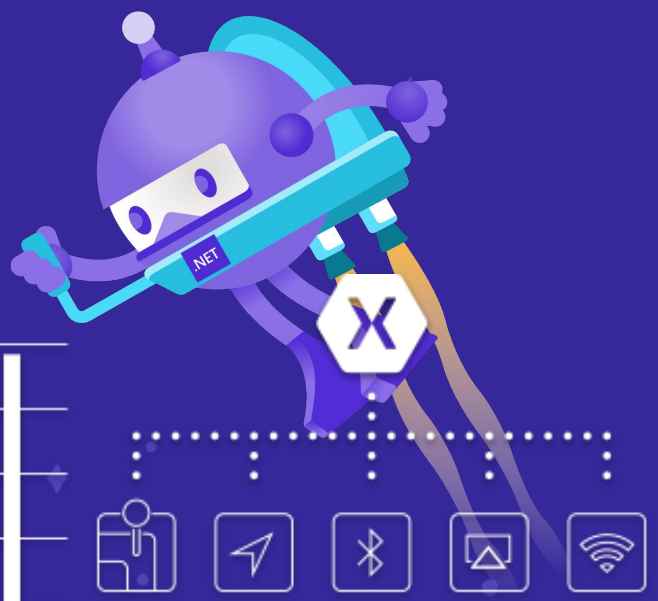
Part 1

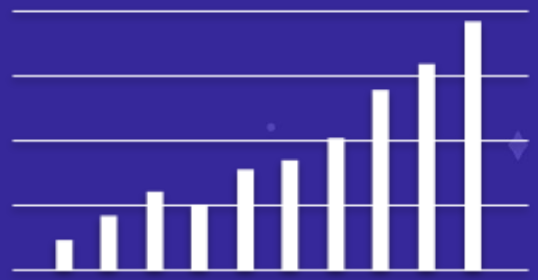
No Mobile Development experience? No problem!

Hybrid vs Cross Platform

- Many definitions, not much consensus
- Hybrid apps are developed using more than one language/technology.
- Does not always mean it will be cross-platform.
- One can use both Objective-c and Swift in single iOS app, both languages talk with each other by using a bridge.
- Commonly accepted as Hybrid:
  Apache Cordova or PhoneGap

Native User Interfaces





Native Performance

Native API Access

Basic electronic components

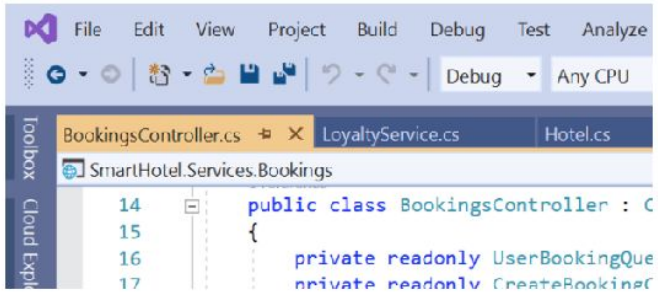
- **Xamarin Native** - “Traditional Xamarin”
- Anything you can do in Objective-C/Swift for iOS, or Java/Kotlin in Android possible in C#
- Higher learning curve

- **Xamarin Forms** - “Instant App Maker”
- Even the UI can be edited just once, for the most part.
- Basic C# skills suffice for maintenance.

Getting Started

Visual Studio

Full-featured IDE to code, debug, test, and deploy to any platform. **Learn more**

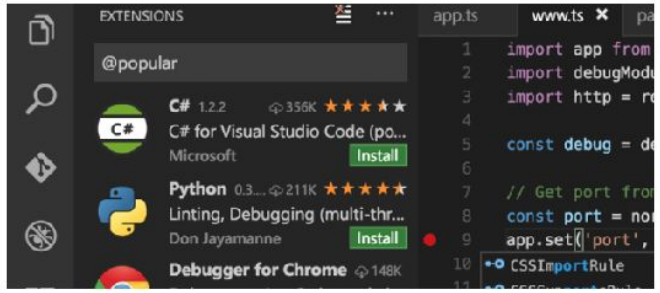


Download Visual Studio

macOS

Visual Studio Code

Editing and debugging on any OS. **Learn more**



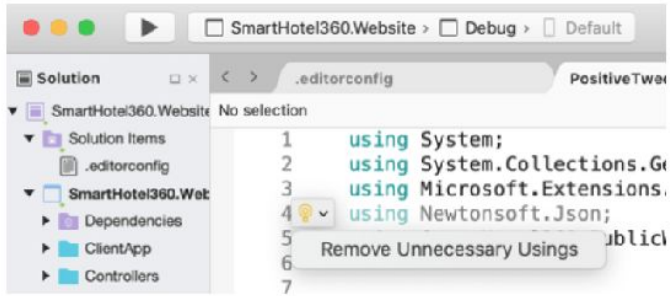
By using Visual Studio Code you agree to its **license and privacy statement**

Download Visual Studio Code

macOS

Visual Studio for Mac

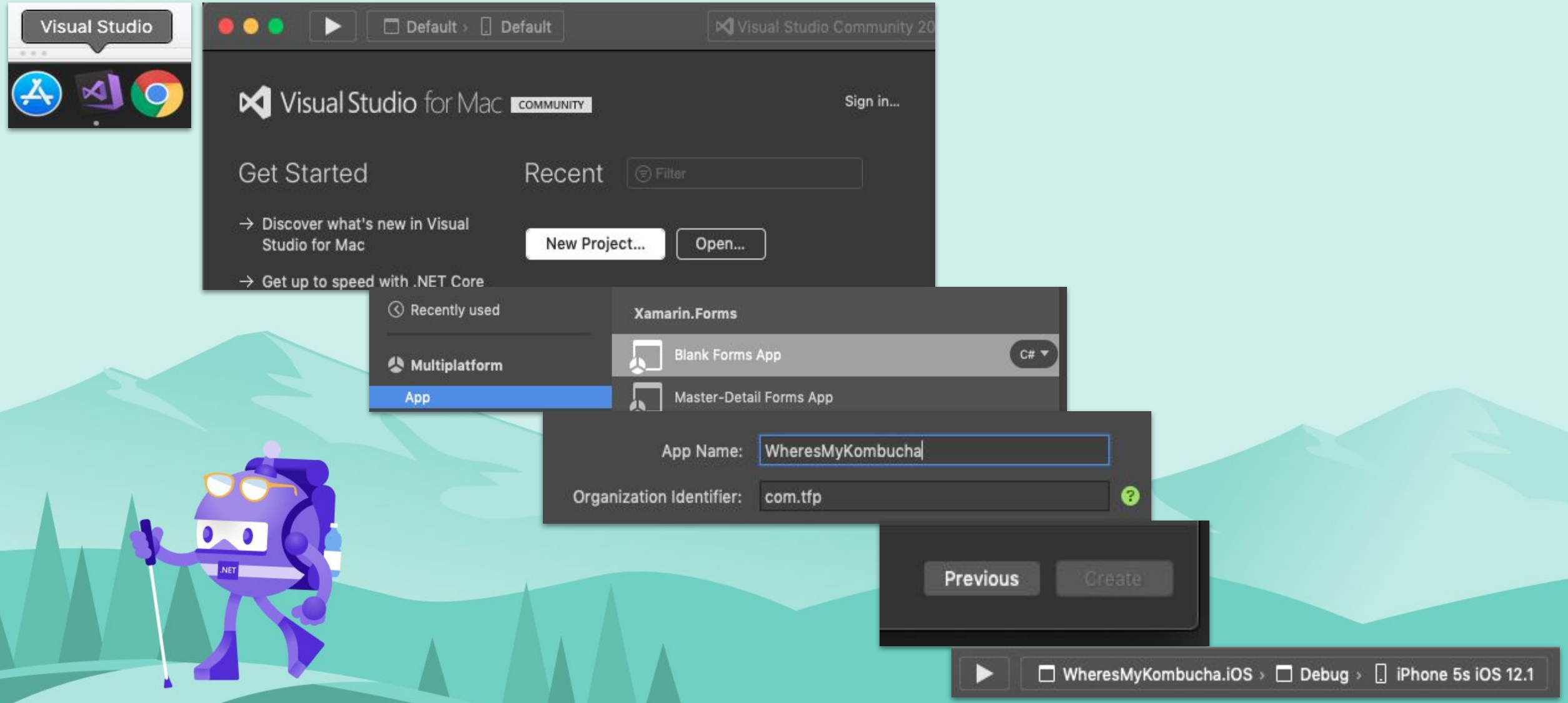
Develop apps and games for iOS, Android, and web using .NET. **Learn more**



Read more about activating your license

Download Visual Studio for Mac

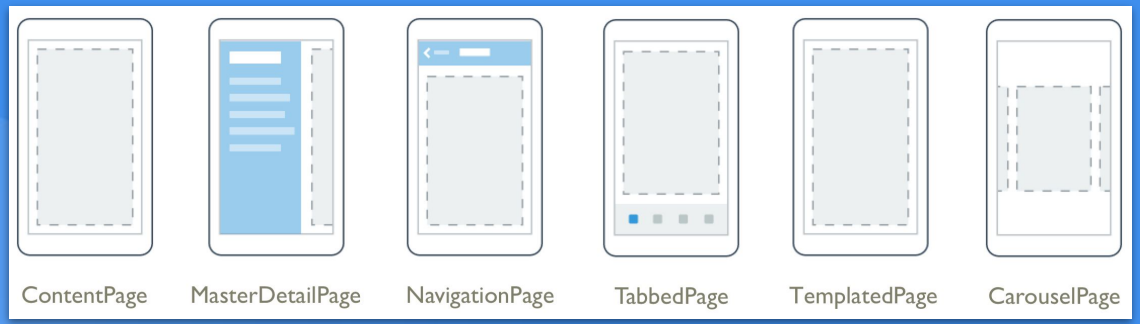
After Installation



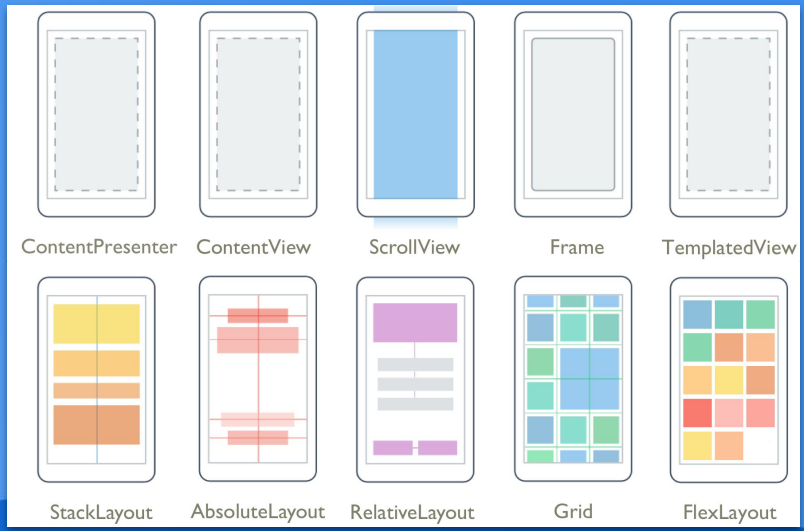
Xamarin Forms Controls

```
<ContentPage Title="Profile" Icon="Profile.png">
  <StackLayout Spacing="20" Padding="20"
        VerticalOptions="Center">
    <Entry Placeholder="Username"
        Text="{Binding Username}"/>
    <Entry Placeholder="Password"
        Text="{Binding Password}"
        IsPassword="true"/>
    <Button Text="Login" TextColor="White"
        BackgroundColor="#77D065"
        Command="{Binding LoginCommand}"/>
  </StackLayout>
</ContentPage>
<ContentPage Title="Settings" Icon="Settings.png">
  <!-- Settings -->
</ContentPage>
```

Pages



Layouts



Quick Changes

- Choose to use XAML and designer files or just C#.
- Add views inside a layout, and set that inside a page.
- Notice the ease of making updates:

```
public class SimpleContentPage : ContentPage
{
    public SimpleContentPage ()
    {
        var label1 = new Label {
            Text = "This is my label",
            Font = Font.BoldSystemFontOfSize(NamedSize.Large)
        };

        var entry1 = new Entry {
            Placeholder = "Type something here"
        };

        var button1 = new Button {
            Text = "Click Me!"
        };

        button1.Clicked += (sender, e) => {
            label1.Text = "User typed: " + entry1.Text;
        };

        Content = new StackLayout {
            Padding = 30,
            Spacing = 10,
            Children = {label1, entry1, button1}
        };
    }
}
```

Connected Electronics

Part 2

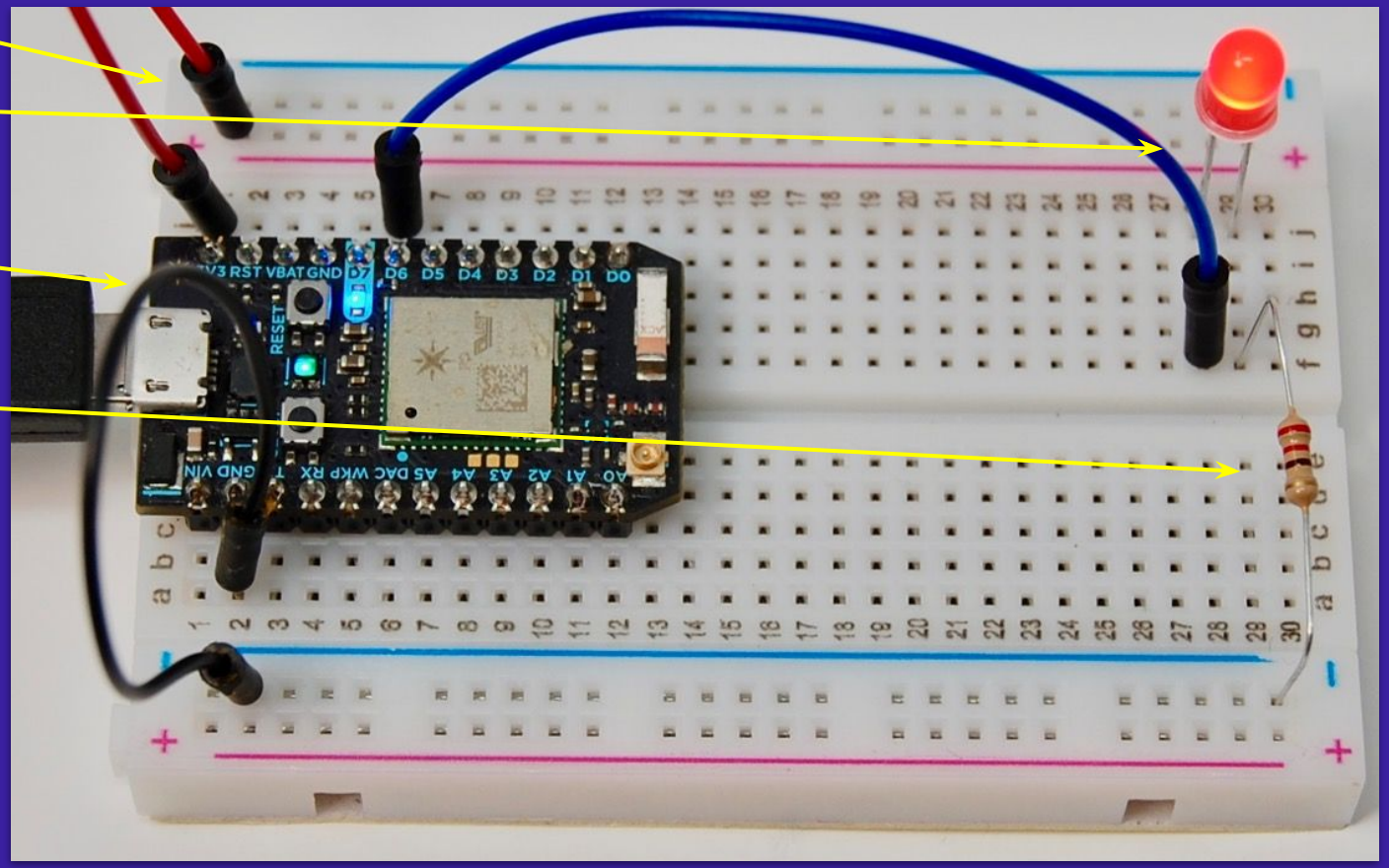
Connecting apps with the physical world

Brief Intro to Electronics

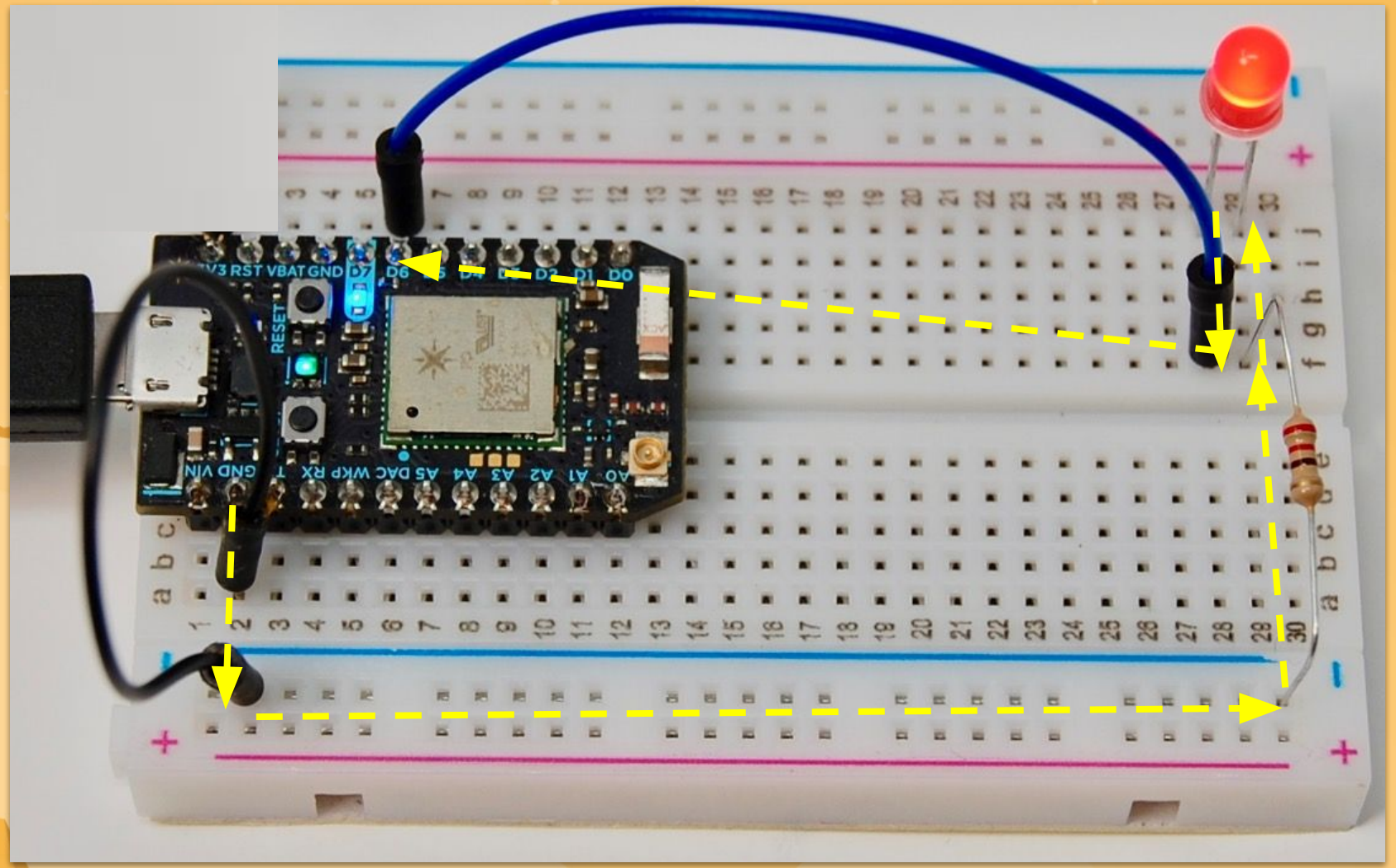
- Microcontrollers vs Microprocessors
- Direct Current - Ground = Black = Negative Photon Particle vs Arduino vs Node ESP 32
- Led - Shorter End is negative, It won't work if you flip the leads
- Code through the browser
- C++, setup() & loop()
- Particle Function, Variable & Events
- Download Particle & connect it to wifi
- DeviceId & `Particle Token Create`

Basic electronic components

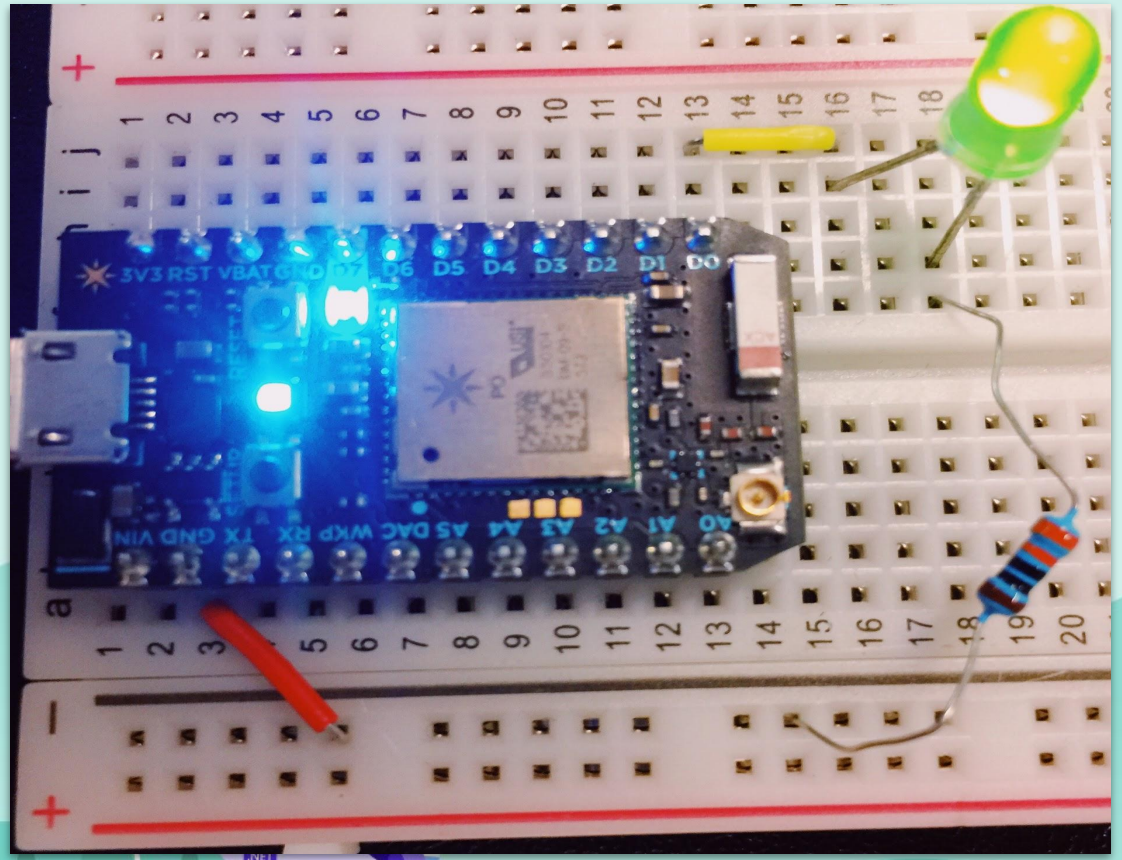
- Breadboard
- LED
- Microcontroller
- Resistor
- Voltage

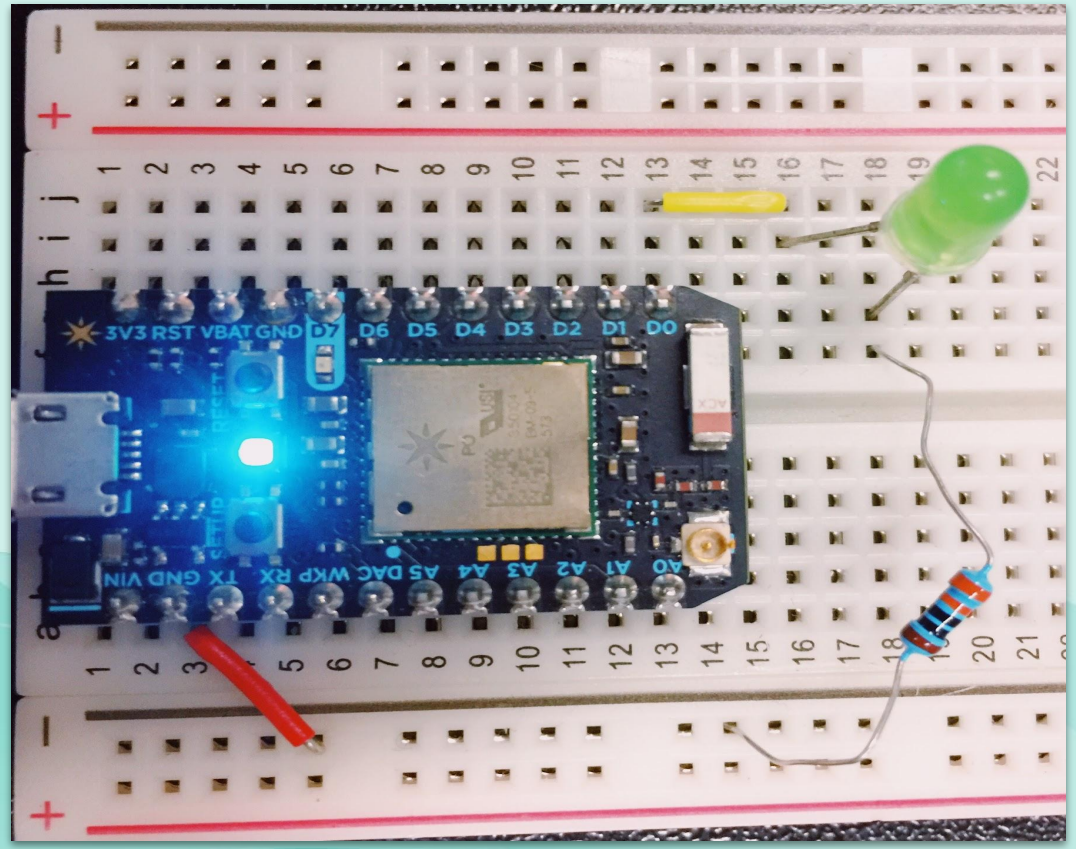


Current Flow Direction



On-Off States





Check Real-time status from Web

https://codepen.io/saamerm/pen/wOxpLm?editors=1100

Where's My Kombucha?

A PEN BY Saamer Mansoor

Save

HTML

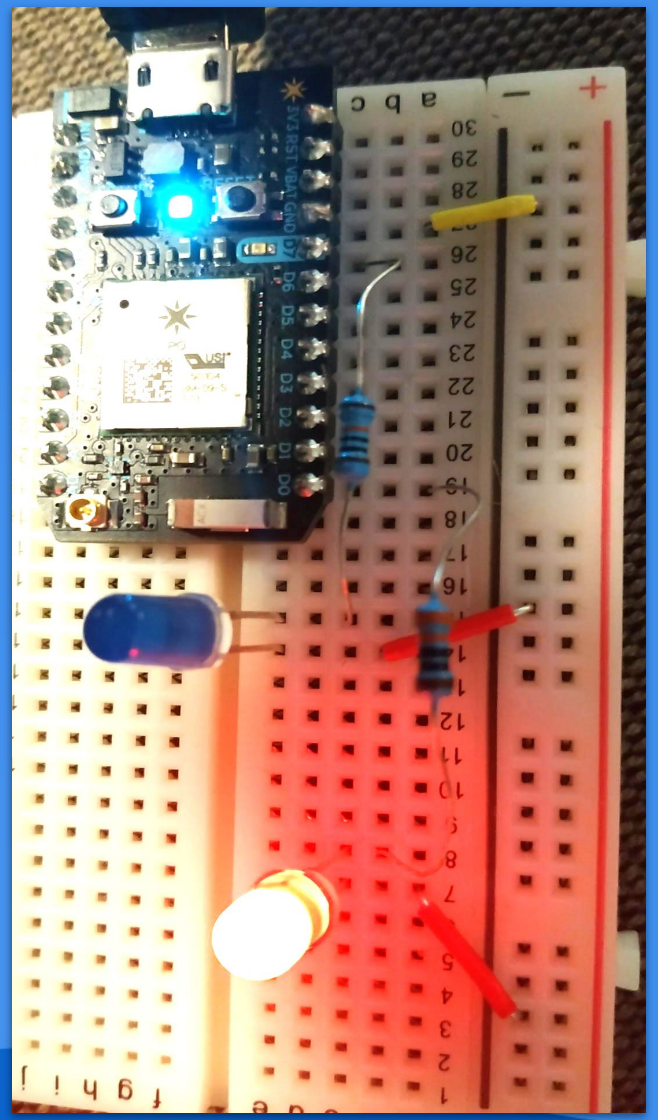
```
<!DOCTYPE>
<html> <body> <center> <br> <br> <br>
  <form
action="https://api.particle.io/v1/devices/28003d001847343338333633/led?
access_token={ACESSTOKEN}" method="POST">
    What's the status of the Kombucha comrade?<br>    <br>
    <input type="radio" name="arg" value="on">Kombucha has arrived!    <br>
    <input type="radio" name="arg" value="off">The buggers finished it all.
    <br>    <br>
    <input type="submit" value="Dispatch Message!">
  </form>  </center>  </body>
</html>
```

What's the status of the Kombucha comrade?

Kombucha has arrived!

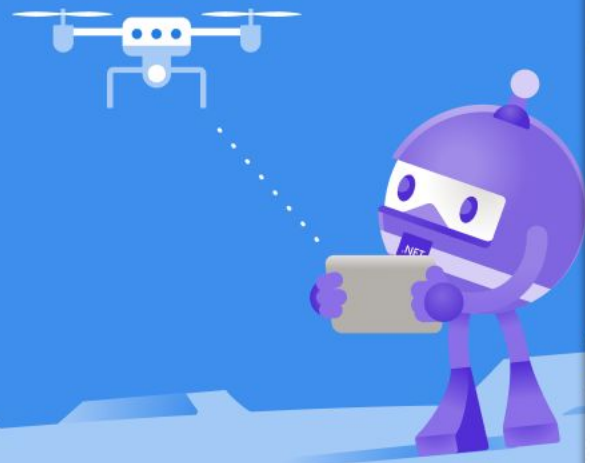
The buggers finished it all.

Dispatch Message!

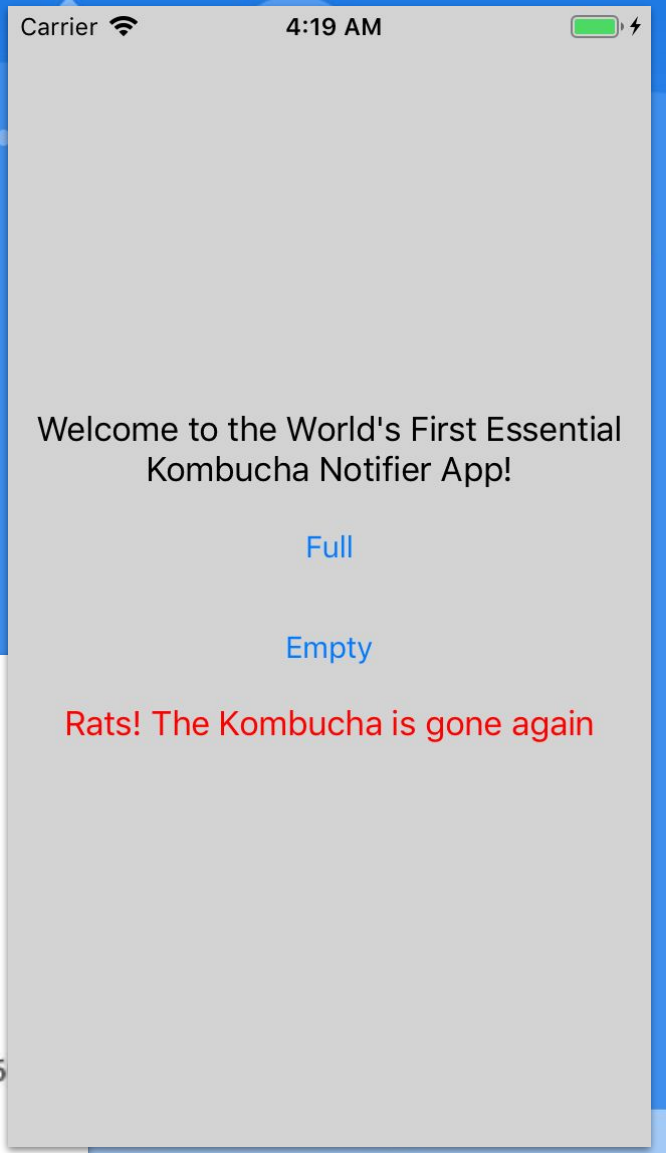


Particle Variable

Get Current Status of the Cloud Variable by calling a GET call to this URL: https://api.particle.io/v1/devices/{DeviceID}/digitalValue?access_token={AccessToken}




```
{
   "cmd":"VarReturn",
   "name":"digitalValue",
   "result":0,
   "coreInfo":{
      "last_app":"",
      "last_heard":"2019-03-19T07:48:13.640Z",
      "connected":true,
      "last_handshake_at":"2019-03-19T07:37:12.36
      "deviceID":"28003d001847343338333633",
      "product_id":6
   }
}
```

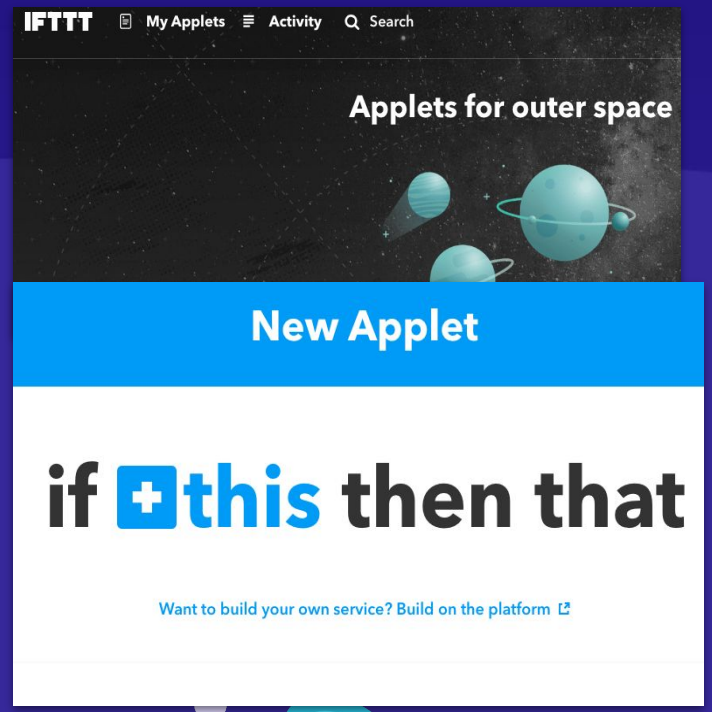
IOT - Internet of Things

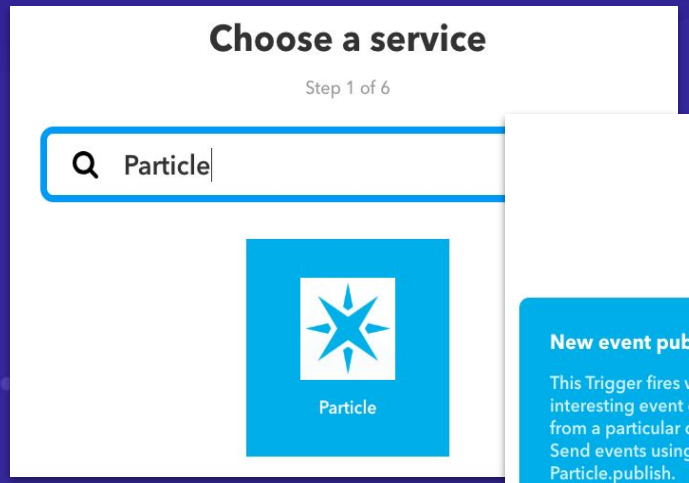
Part 3

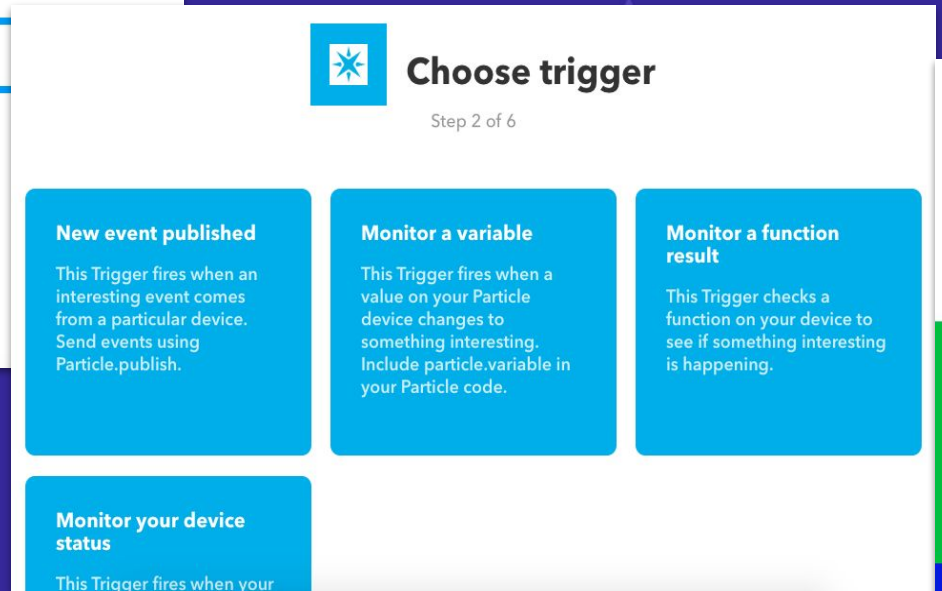
Leveraging the power of other connected services

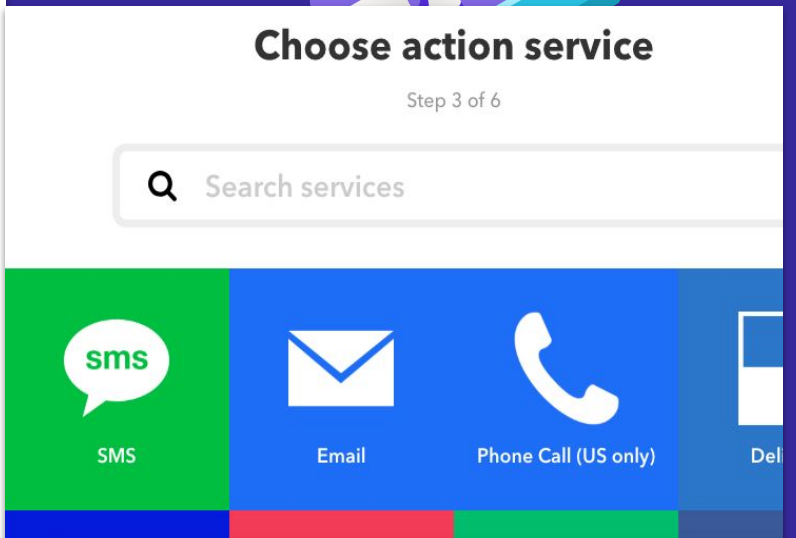
IFTTT - If This Then That

- Go to https://ifttt.com
- Select New Applet.

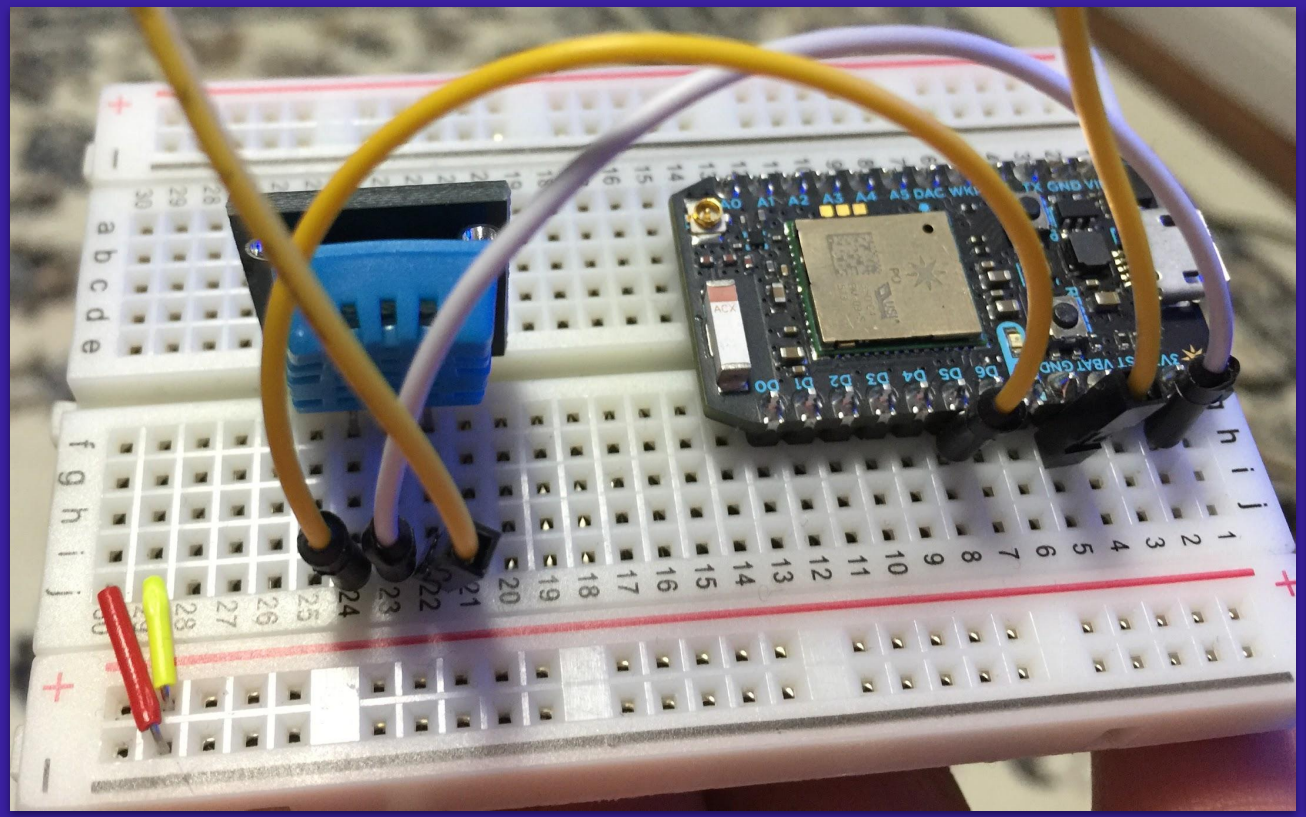




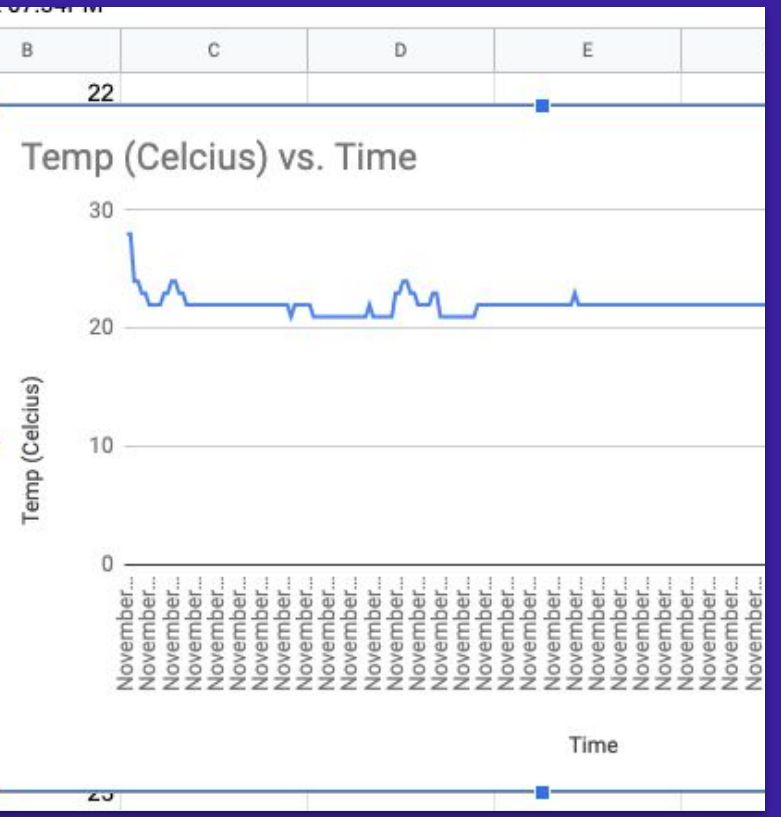




Experimenting with Temperature



	A	B	C	D	E
233	November 19, 20				
234	November 19, 20				
235	November 19, 20				
236	November 19, 20				
237	November 19, 20				
238	November 19, 20				
239	November 19, 20				
240	November 19, 20				
241	November 19, 20				
242	November 19, 20				
243	November 19, 20				
244	November 19, 20				
245	November 19, 20				
246	November 19, 20				
247	November 19, 20				
248	November 19, 20				
249	November 19, 20				
250	November 19, 20				
251	November 19, 20				



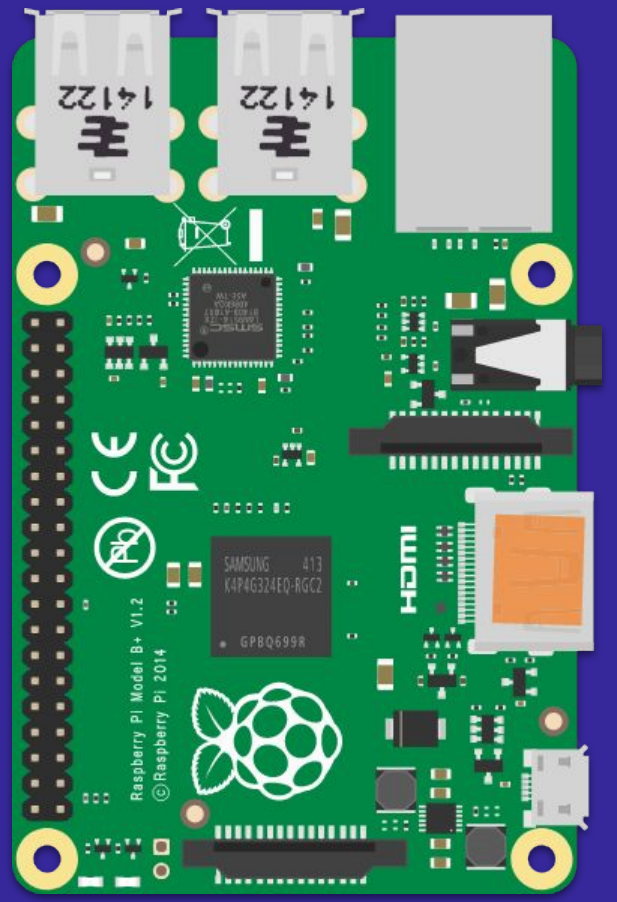
Using Raspberry Pi’s instead

Part 4

Swapping electronics to create the sanitizer

What do we need to change?

- Microprocessor-Can run multiple applications
- Can be coded with python
- 4 USBs, Audio Out, HDMI, WiFi



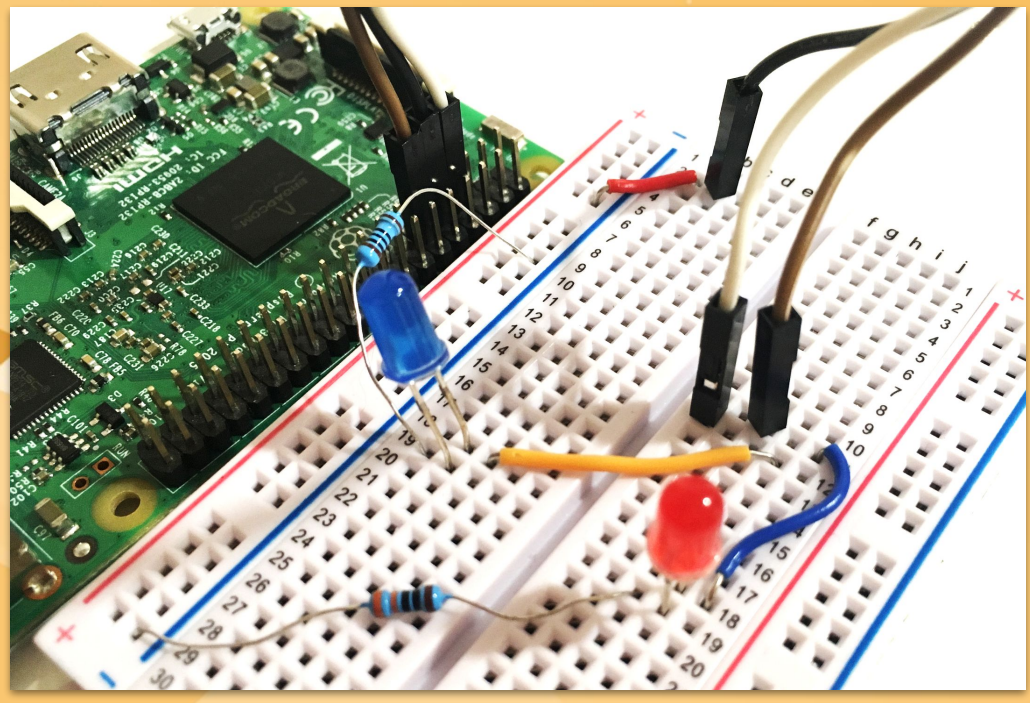
Raspberry Pi Setup - From scratch

- Install Raspbian OS on your Raspberry Pi's micro SD Card:
  https://www.raspberrypi.org/software/
- Open the terminal, install particle library:
  bash <(curl -sL https://particle.io/install-pi)
- Create account here: build.particle.io
- Add your raspberry pi to the Particle Cloud:
  "particle-agent setup" and login.
- Change the Device type in the Particle Web IDE, so code flashes to the Raspberry Pi
- Don't forget to update the DeviceID in your Xamarin code
- Create a token : $ particle token create

What do we need to change?

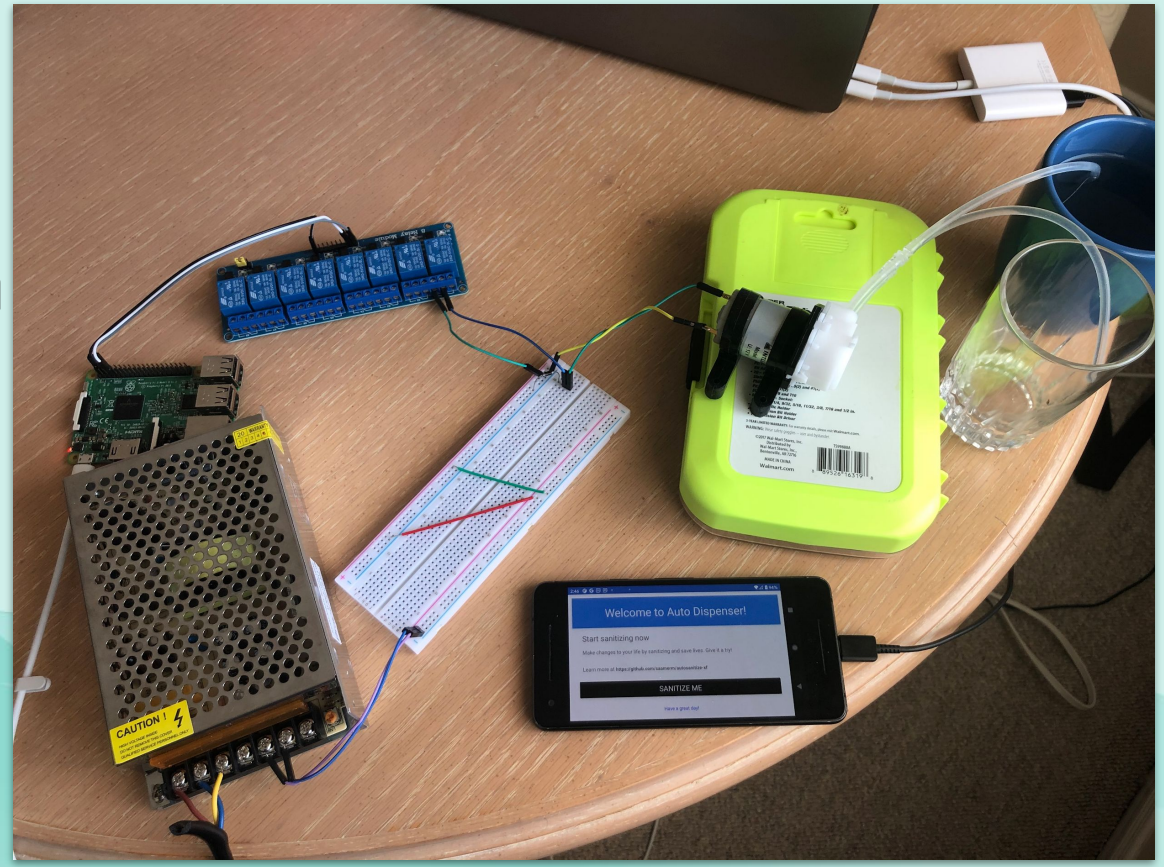
- Search for Pin-Out Diagram
- Connect D0, GND & D1 of the Raspberry Pi
- Deploy same code from Build IDE

Peripherals	GPIO	Particle	Pin #			Pin #	Particle	GPIO	Peripherals
3.3V			1	X	X	2	5V		
I2C	GPIO2	SDA	3	X	X	4	5V		
	GPIO3	SCL	5	X	X	6	GND		
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND		
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND		
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI (chip enable)
GND			25	X	X	26	CE1	GPIO7	
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND		
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND		
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O
GND			39	X	X	40	D16/A7	GPIO21	Digital I/O



What do we need to change?

- Power supply, Relay, Pump,
- Diode & Sanitizer for protection
- Code changes
- RaspberryPi

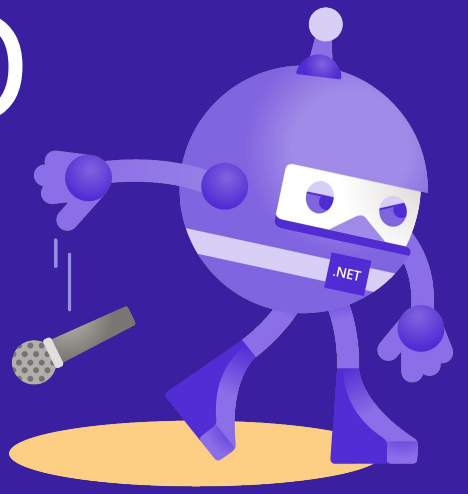


Thanks for joining!

Questions?

GitHub: github.com/saamerm/WheresMyKombucha
Email: saamer@thefirstprototype.com
LinkedIn: linkedin.com/in/saamerm
Twitter: @saamerm

SLIDES BEYOND THIS TO BE DELTED ONCE FINISHED



Single column of content

Two columns of content

Demo

Presenter

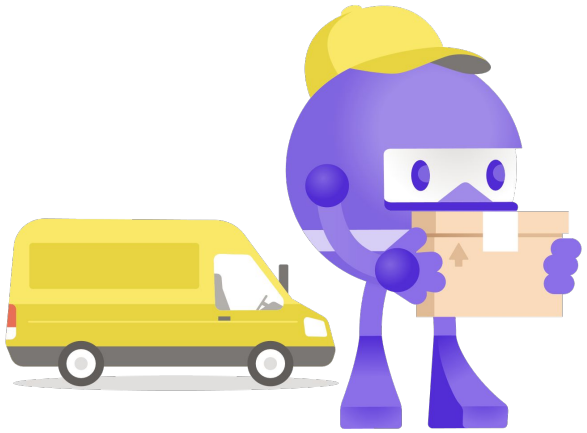
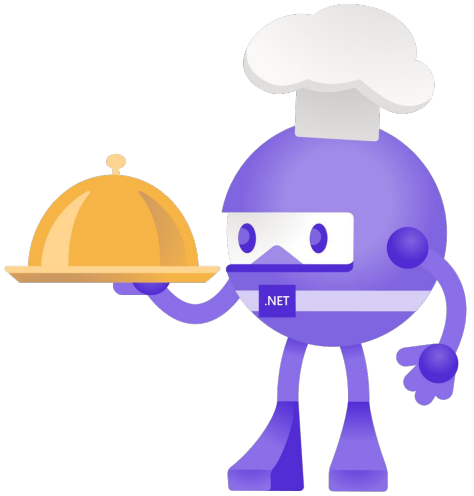
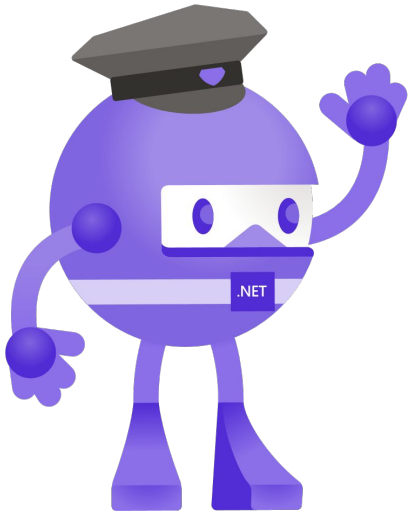
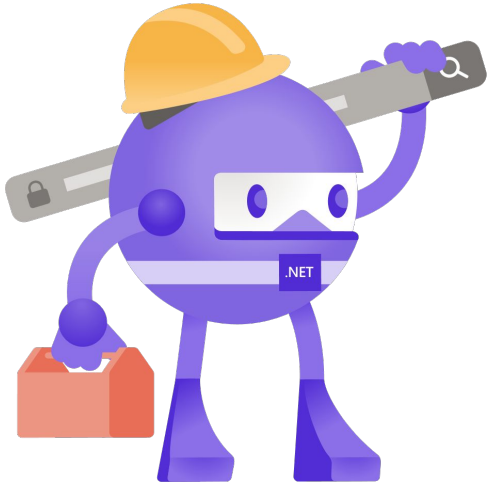
Code Sample

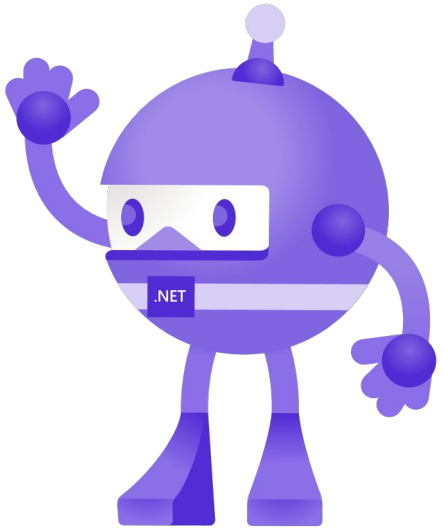
```
Code sample
```

RELEASED

Announcemnet

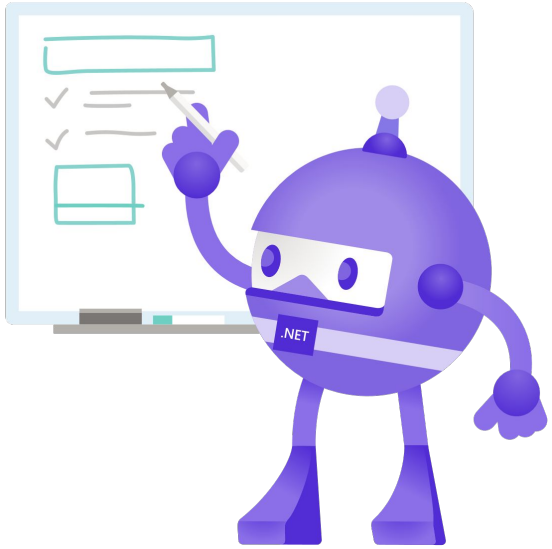
- Value prop 1
- Value prop 2
- Value prop 3

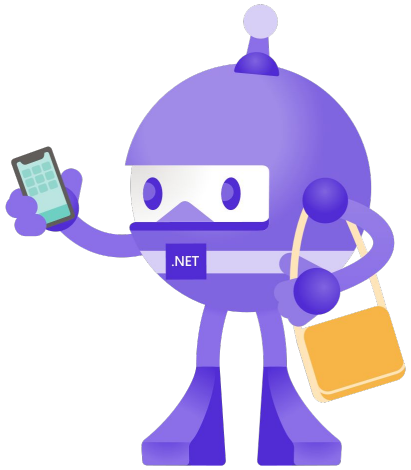




.NET

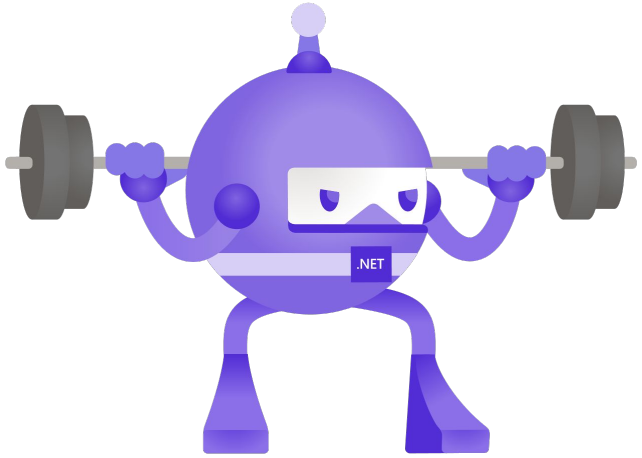

.NET

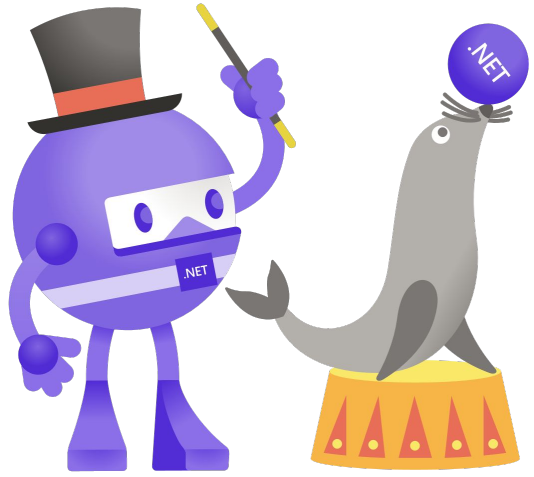

.NET

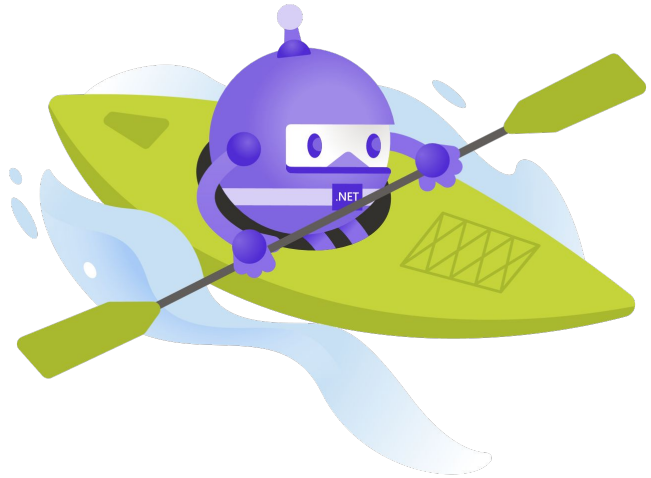

.NET

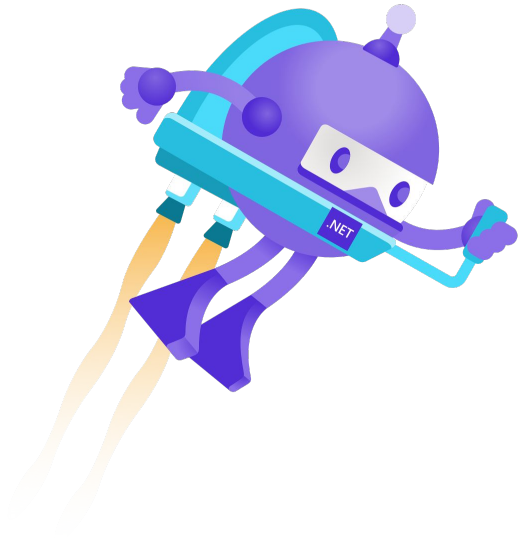
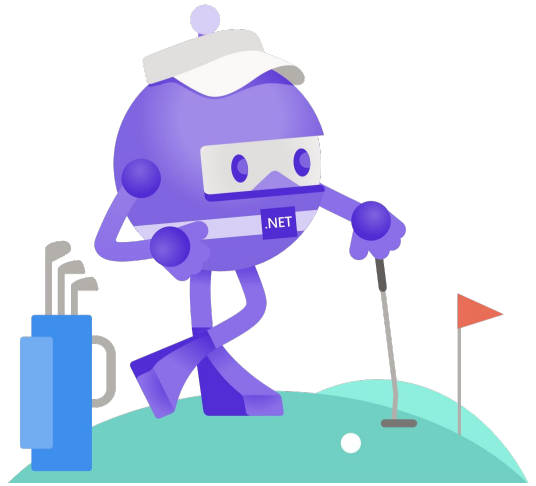

.NET

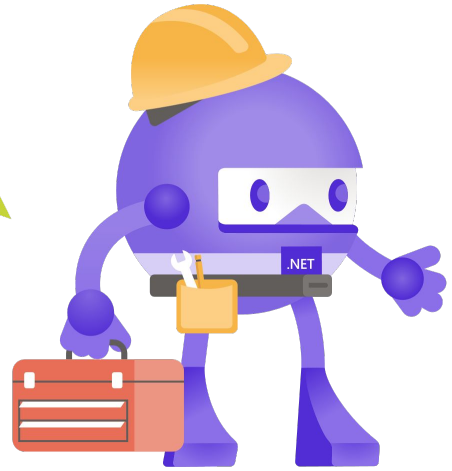

.NET

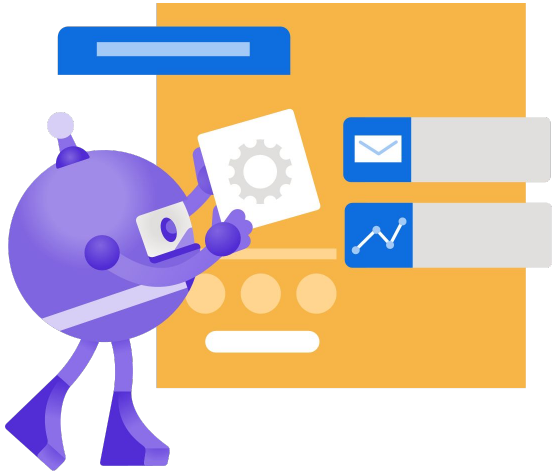
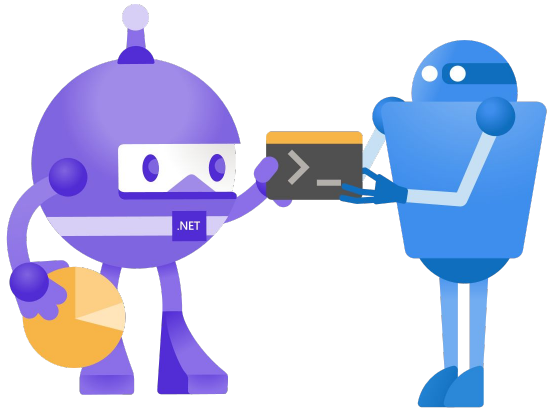

.NET

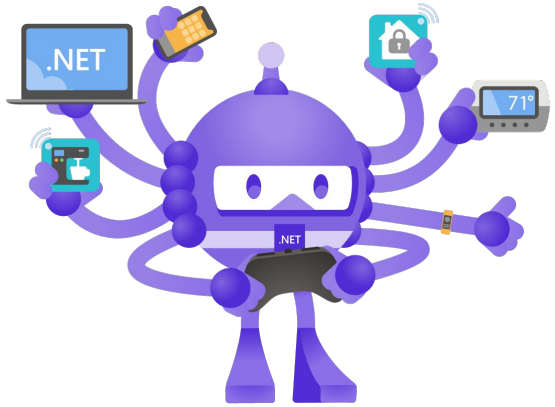

.NET
.NET

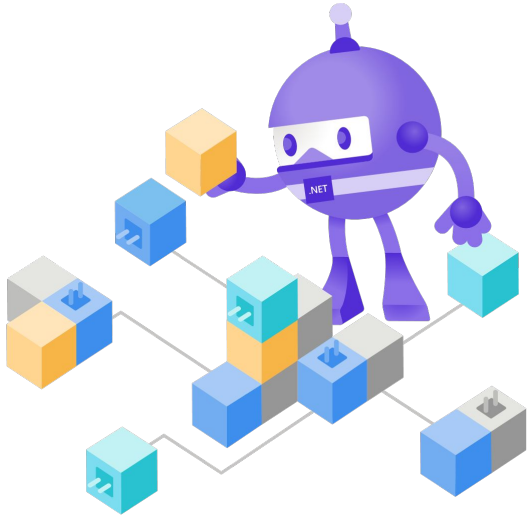
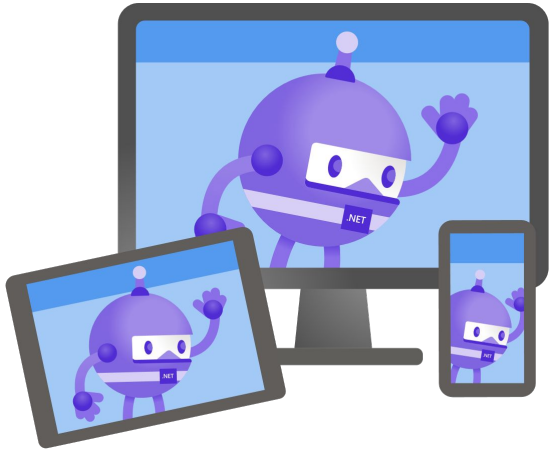


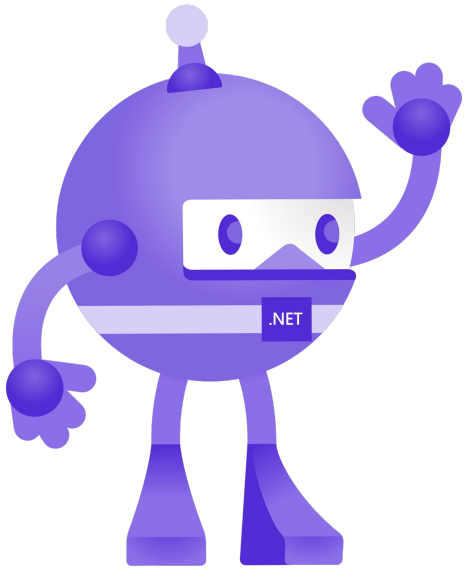



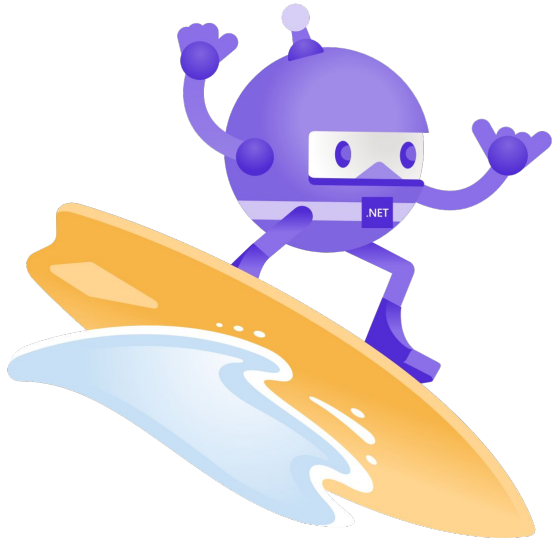


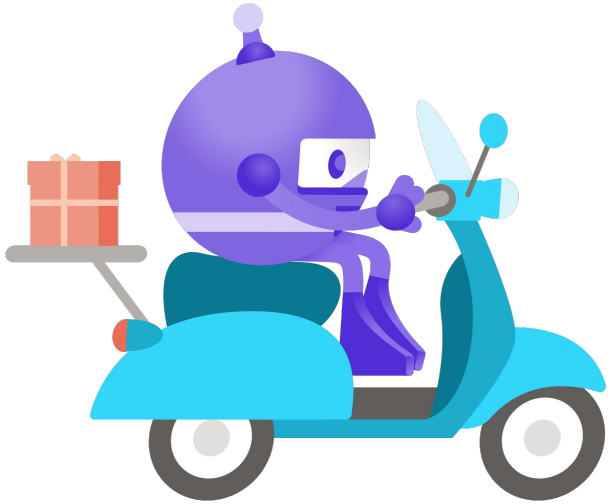

.NET
71°


.NET


.NET


.NET




.NET


.NET