**High Level**

RemoteControl -> Cryptohome Api -> Cryptohome

**Relationships**

CryptohomeManager -> ControlCentreManage -> IDeviceManager

Devices have a Type property that instantiates the IDeviceManager to the correct instance.

* OpenClose => DeviceOpenCloseManager
* OnOff => DeviceOnOffManager

**Behaviours**

* Cryptohome
  + Receives commands from RemoteControl
  + Simulates device actions
* RemoteControl
  + Communicates with Cryptohome using standard HTTP protocol
  + Controls one to many home devices

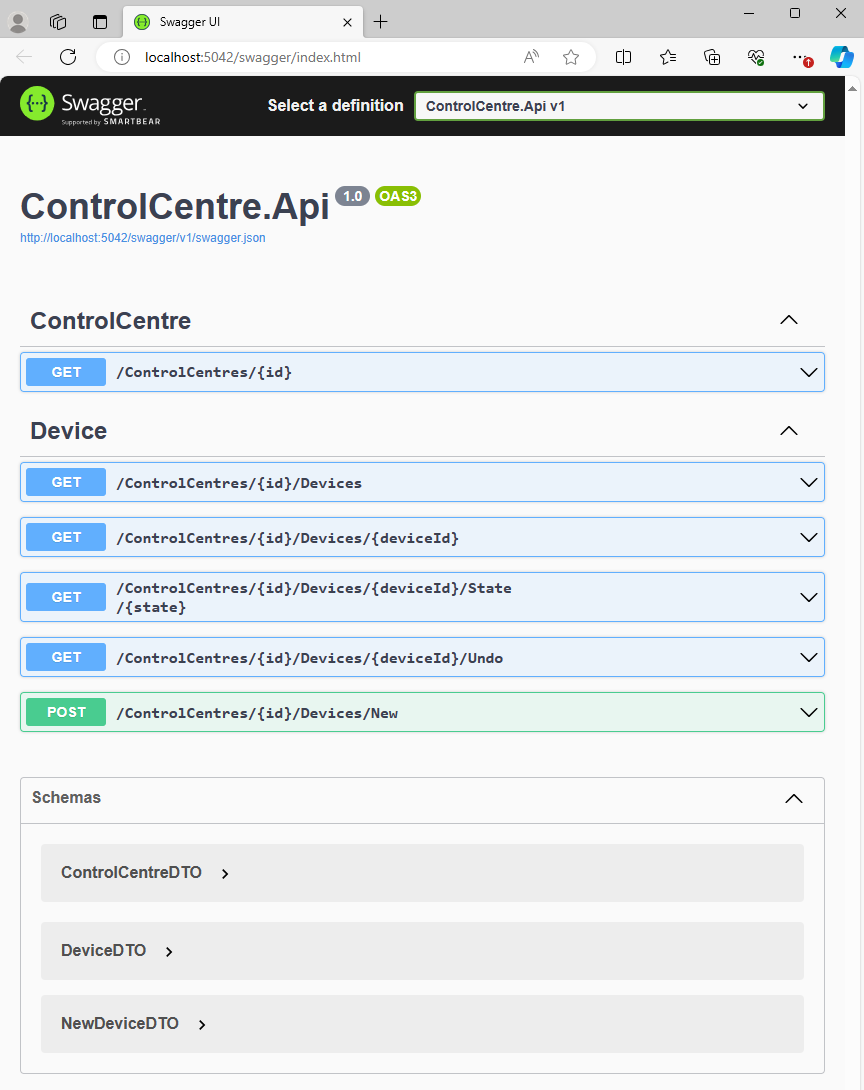
API has 2 controllers

* ControlCentre
* Device

All communication is transferred using 3 DTO’s

* ControlCentreDTO
* DeviceDTO
* NewDeviceDTO (single responsibility)

The API supports multiple ControlCentres



This is the swagger file for the api. For the mock data use ControlCentres id = 1

**Example Api Usage**

Display Control Centre 1

/ControlCentres/1

ControlCentreId: 1. ControlCentreName = Test Control Centre

List Devices in Control Centre 1

/ControlCentres/1/Devices

DeviceId = 1, Device Name = Garage Door, Device State = close, Previous State = close

DeviceId = 2, Device Name = Dishwasher, Device State = off, Previous State = off

DeviceId = 3, Device Name = Lights, Device State = off, Previous State = off

Open Garage Door

/ControlCentres/1/Devices/1/State/Open

DeviceId = 1, Device Name = Garage Door, Device State = open, Previous State = close

Undo Garage Door Change

/ControlCentres/1/Devices/1/Undo

DeviceId = 1, Device Name = Garage Door, Device State = close, Previous State = open

Add New Device – this html post is for effect not tested but most likely will work

/ControlCentres/1/Devices/New Body -> NewDeviceDTO

**Design Patterns**

* Lean Methodology
* Dependency Injection Pattern
* Dependency Inversion Principles (SOLID)
* Interface Segregation Principles (SOLID)
  + DTO – Data Transfer Objects
  + IDeviceManager – allows extensibility of device types
* MVC – Model View Controller Pattern
* Command Pattern
* BFF – Backend for Front End Pattern – single use API only to support the remote control
* Singleton Design Pattern
* Single Responsibility Principles (SOLID)
* Open-Closed Principles (SOLID)