Open Web Interface for .Net – подход, который используется для разработки веб-сервисов у майкрософта. ОВИН интерфейс – который описывает взаимодействие веб-сервера, серверного приложения и хоста (приложение, в которое встраиваем, которое обеспечивает процесс операционной системы). Взаимодействие между 3 этими компонентами описывается при помощи интерфейса owin. Именно этот подход позволил легко перескочить на asp .net core. Архитектура выглядит следующим образом: есть хост прилаги, есть owin сервер (веб-сервер), есть наше приложение. Owin описывает то, как они между собой взаимодействуют – хост, сервер и прилага. User-agent – хттп клиент, который взаимодействуте по протоколу хттп с нашим клиентом.

Хост нужен для того, чтобы создать процесс операционной системы. Для того, чтобы запустить какое-то приложение в рамках операционной системы, нужно написать exe, dll, которая будет загружена и активизированна. Вот такое приложение, в которое можно впихнуть объект сервера и прилагу – и есть хост. А сервер должен принимать запросы, обрабатывать битовые последовательности и превращать их в реквест. Формировать ответ и отправлять на сторону клиента.

Middleware – хттп модули, которые обеспечивают конвееер обработки между клиентом и сервером.

OWIN – это набор интерфейсов, которые должны реализовывать объекты сервера, хоста и прилаги.

В соответствии с этой спецификацией, которая описывает способ взаимодействия между этими компонентами, есть проект Катана, который реализовал такой сервер. В качестве хоста можно использовать iis или другое приложение (self-hosting), а еще интерфейс позволяет подключать middleware блоки. Катана предлагает 2 варианта реализации сервера:

- основан на иисе и предлагается на выбор 2 варианта сервера: systemweb или helios. Поддерживается middleware

- без использования iis: owinhost или self-host, предлагается использовать 1 сервер httplistener, также есть middleware.

Само приложение тоже поддерживает owin интерфейс, в соответствии с которым существует несколько готовых frameworkов, с помощью которых можно реализовать архитектуру owin: WebApi, SignalR. Nancy и др.

Таким образом, для чего это все делается? Для того, чтобы можно было самостоятельно разрабатывать любые компоненты, которые здесь есть. Можем создавать свои собственные конфигурации и фраемворки для нашего приложения.

Класс Startup – класс, который стартует приложение. Содержит конфигурации owin приложения. Теперь можно через nuget создать приложение различной конфигурации с необходимым подходом.

Позволяет менять хостинг – не обязательно юзать iis или не юзать iis. Для другой ОС можно выбрать свою конфигурацию