Calling Web Services from Unity

Developing a web service

- This demo will use Node.js (https://nodejs.org/)
  - Runs on command line -- you don’t have to install Apache, IIS, etc. to work locally
  - Makes sending and receiving JSON easy
- DotNetCore has a similar approach
- Really, any platform will work

Connecting to HTTP/HTTPS in Unity

- Unity supports this with UnityWebRequest
  - Asynchronous; check isDone each Update/coroutine
- This demo will use RestClient (https://github.com/proyecto26/RestClient)
  - Wrapper around UnityWebRequest for compatibility
  - Asynchronous via Promises

Promises in C#

- Alternative to callbacks
- RestClient implements via https://github.com/Real-Serious-Games/C-Sharp-Promise
- Creating a Promise object begins an asynchronous process (defined as a function)
- Methods on Promise handle the result of that process:
  - .Then((result) => Debug.Log("Got response: " + result))
  - .Catch((error) => Debug.Log(error))
- Can chain multiple Promises and Then methods together

Hosting Node.js or DotNetCore

- ❖ Applications run their own servers, rather than being run through a standard web server
- ❖ The application is proxied through a web server like Apache or IIS, where other options (SSL, etc.) can be configured
- ❖ Many cloud services (Azure App Services, etc.) can handle these applications directly
- ❖ IIS can run Node.js via iisnode (https://github.com/Azure/iisnode)
- ❖ Apache and other servers can run through a proxy

Questions? Comments?

https://www.dylanwolf.com/

@dylanwolf