Intro:

- love me some .NET

- in .NET for ~2 years, focus in web, specifically new stuff

- work at A1, education space, .NET-stack systems {EF, MVC} and {Bootstrap, AngularJs}

Introduce talk/tech

- first talk

- indie .NET (may have roots, not mainstream)

- particularly Nancy, SignalR, Raven.

- main bits .NET, client-agnostic

Introduce organization of talk

- first, overview {use case, licensing, etc}

- speed-demo

- break for questions about tech

- guesstimate

What is NancyFx?

- "lightweight web framework for .NET and Mono"

- run-anywhere (agnostic of host)

- view-engine agnostic

- highly extensible

- highly testable

- something that can replace MVC/WebForms, given love for syntax

Get["/"]

hello world

[talk about solution as is]

- create module, like controller in MVC.

front-end of server that deals with HTTP requests

- HomeModule : NancyModule

Get["/"] = \_ => "Hello, I'm Nancy!";

Get["/{something}"] = ctx => "Hello, " + ctx.something + "!";

Get["view"] = \_ => View["index"];

- index

- explain dynamic dict, Func<dynamic, Response>

- run

Test hello/{something}

[talk about project as-is]

public class HomeModuleTests

{

[Fact]

public void hello\_something\_should\_respond\_appropriately()

{

var word = "world";

var sut = new Browser(new DefaultNancyBootstrapper());

var actual = sut.Get("/hello/" + word);

Assert.Equal(HttpStatusCode.OK, actual.StatusCode);

Assert.True(actual.Body.AsString().Contains(word));

}

}

Person Model

Person {Name, Id}

Test ApiModule

ApiModuleTests

SNIPPET: [sublime tab]

All should fail

explain stuff

ApiModule

SNIPPET: [sublime tab]

4 things to note:

-1- :base("api")

-2- Response & AsJson

-3- Post route; just another dynamic dict

-4- Bind<>, deserializes request body

NancyFx Questions?

Why should I care about real-time communication?

- before signalr, introduce problem

- virtualcubing1.azurewebsites.net

- I cube (yes, thing)

- 2 chromes, separate clients

- point: collaboratively solve virtual cubes

- updated psuedo-instantly

this is real-time and shows up in games, chat, even business apps

What is SignalR?

- "incredibly simple real-time web for .NET"

- standard .NET solution

- kinda like NodeJs' socket.io

- client-agnostic.

Official: .NET, JS, Java

Unofficial: Android, ObjC, even Go

Demo - server

[open sln]

hubs would be ready, but none available. set up endpoints

endpoints - hubs

[pull in scripts]

[pull in html]

[pull in main script]

More SignalR things

- have basic understanding

bulk of what you need real-world

- connectionids - guids, given by context

- groups, client-side regulated

- jquery dependency

SignalR questions

What is NoSQL, why care?

- sql been around for a while

Why care?

- data

- not relational

- load from 10 tables

- in nosql, could pull from one JSON document

- flexible data model

- easy to change your structure

- easy to reindex

- scaling

- scaling out (dist) vs scaling up ($$, big)

What is RavenDb?

- have rough idea of nosql

- "...2nd generation document DB."

- Essentially, it's a NoSQL document database that stores all of it's data as JSON.

- transactional

Running Raven, Mgmt Studio

- easy to set up, just download .zip

- [run]

- silverlight

.NET Raven Client

- note .NET & HTTP clients

- yet another nancyfx

- will define two routes: POST, GET

- [Install-Package RavenDb.Client]

- [make person model]

-

docStore = new DocumentStore { Url = "http://localhost:8081//" };

docStore.Initialize();

Get["/"] = \_ =>

{

var session = docStore.OpenSession();

return Response.AsJson(session.Query<Person>().ToList());

};

& test this.

POST[""] = \_ =>

{

var newPerson = this.Bind<Person>();

var session = docStore.OpenSession();

session.Store(newPerson);

session.SaveChanges();

return Response.AsText(entity.Id.ToString()).WithStatusCode(HttpStatusCode.Created);

};

HTTP Client

[go to browser, do thing]

RavenDb Questions