Angular Provider Usage

When developing libraries, you will come across the need for your library to somehow, convey or pass on, any registration needs it has to the consuming application, or consuming library.

Simple Example

The typical way to do this, is to include a lib-providers file with a function that returns a providers array.

Here's a simple example of a lib-providers.ts:

```
import { ApplicationConfig } from '@angular/core';
import { provideHttpClient, withFetch } from '@angular/common/http';

export const LIB_OGAWEBUISHAREDKERNEL_PROVIDERS: ApplicationConfig =
{
  providers: [provideHttpClient(withFetch())]
};
```

The above is a simple lib-providers.ts file that a library would contain.
It includes a provider call to register the HttpClient, as the library uses it.

And, the consuming application would simply add that provider call to its providers block, like this:

```
import { LIB_OGAWEBUISHAREDKERNEL_PROVIDERS } from 'lib-oga-webui-sharedkernel';

export const appConfig: ApplicationConfig = {
  providers:
  [
    provideZoneChangeDetection({ eventCoalescing: true }), provideRouter(routes),
    ...LIB_OGAWEBUISHAREDKERNEL_PROVIDERS.providers
  ]
};
```

The above example is how an app would include a library's needed providers, for DI registration.

> NOTE: The usage of the spread operator (...) on the providers line.
> This flattens the received array to one-dimension.

Revision #4
Created 16 March 2025 21:08:19 by glwhite
Updated 17 March 2025 00:49:43 by glwhite