

Extra DataPower Development Exercises

HTTP Service

Create an HTTP Service that will serve up static content (html, css, ...) files.

These files will be stored in the folder `local:///rates`.

Steps:

- in your application domain, create the folder `local:///rates`
- create and upload the 3 html files below into this new folder
- create HTTP Service: *Rates-HTTP-Service*
 - port `8080`
 - Base Directory: `local:///rates`
 - Start Page: `local:///rates/index.html`

index.html

```
<html>
<body>
Welcome to the <i>Todays Rates</i> service.<br/>
Rates may be found on todays-rates.xml or todays-rates.json.<br/>
</body>
</html>
```

todays-rates.xml

```
<todays_rates>
  <gbp>1.4</gbp>
  <usd>1.2</usd>
</todays_rates>
```

todays-rates.json

```
{"gbp": 1.4, "usd": 1.2}
```

Test that the service is operational by:

- using a browser
- using curl

For the following exercise, the above will be considered an internal service, only to used by other services.

MPGW Service

MPGW *Rates-MPGW*

Create a MPGW service *Rates-MPGW*

- that can be called by an end user using either of the following URIs
 - `/html`
 - `/json`

The two rules implied above will have code that accesses the appropriate information from the HTTP service and return it in an appropriate format.

Both of these rules use code that make a sideways call to the above HTTP service to access the relevant values in the relevant format. Both rules will also loop-back, meaning there is no call to a backend.

To make this sideways call:

- the /html rule will include an XSL (Transform) that makes an `url-open` call
- the /json rule will include a GWS (GatewayScript) that makes an `urlopen` call.

(These file are to be stored in a subfolder with the same name as this service.)

The code of each enhances the data received from the side call as follows and is returned to the client:

/html

```
<html>
  <body>
    Todays rates are: <br/>
    British pounds: 1.4 <br/>
    US dollars: 1.2 <br/>
  </body>
</html>
```

/json

```
{
  "Todays_rates": {
    "British_pounds": 1.4,
    "US_dollars": 1.2
  }
}
```

where the output is produced by the same XSL / GWS that makes the sideways call, and the values in red above are extracted from the `url-open` / `urlopen` responses from the HTTP service.