

DataFlex

DATAFLEX WEB API FRAMEWORK

NIELS DE BOER

01

WHY?

WHY?

CURRENT SITUATION

- WebHttpHandler
- Time consuming
- Manual work
 - Performing data dictionary operations
 - Converting to JSON
 - Writing documentation
- Repeat for each verb (GET,POST,PUT,PATCH and DELETE)

WHY?

CURRENT SITUATION ENDPOINT

```
072 // Build up the SQL query to get the projects
073 Move (@SQL ""
074     SELECT ${LIMIT}
075         Project.ProjectID      AS ID,
076         Project.Name          AS Name,
077         ProjectType.Description AS TYPE,
078         ProjectStatus.Code     AS StatusCode,
079         ProjectStatus.Description AS StatusDescription,
080         Project.Description    AS Description,
081         Customer.Company       AS Customer
082     FROM
083         Project
084     JOIN
085         ProjectStatus
086         ON Project.PrjStatus = ProjectStatus.ProjectStatusID
087     JOIN
088         Customer
089         ON Project.CustomerID = Customer.CustomerID
090     JOIN
091         ProjectType
092         ON Project.PrjType = ProjectType.PrjTypeID
093     "" to sSQL
094
095 // Add a result set limit if the limit is specified
096 Move (Replace("${LIMIT}", sSQL, If(sLimit <> "" and IsInteger(sLimit), ("TOP" * sLimit), ""))) to sSQL
097
098 // Add filter on project type if type is specified
099 If (sType <> "") ;
100     Move (sSQL * @SQL "WHERE ProjectType.[TYPE] = ${projecttype}") to sSQL
101
102 // Prepare the SQL statement
103 Send SQLPrepare of ghoSQLExecutor sSQL
104
105 // Set the parameter
106 If (sType <> "") ;
107     Send SQLSetParameter of ghoSQLExecutor "projecttype" sType
108
109 // Execute the SQL statement to the struct
110 Get SQLExecute of ghoSQLExecutor to taProjects
111
112 // Add Project Allocations for each project
113 For iCount from 0 to (SizeOfArray(taProjects)-1)
114     Get GetProjectAllocations taProjects[iCount].IID to taProjects[iCount].taAllocations
115 Loop
```

WHY?

CURRENT SITUATION ENDPOINT

```
// Set Response header to JSON
Send AddHttpResponderHeader "Content-Type" "application/json"

// Convert the struct to JSON and output it
Send StructToUchar taProjects (&ucData)
Send OutputUChar ucData
```

02

WE CAN DO BETTER

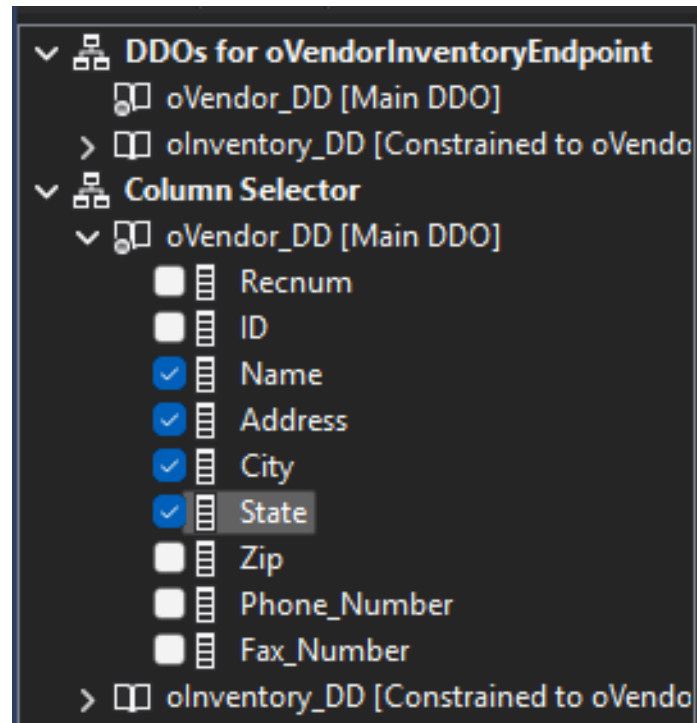
WE CAN DO BETTER

HOW DID WE IMPROVE?

- cHttpApi
- Automatic data dictionary operations
- Automatic data conversion (JSON/XML)
- DDO Drag and drop support
- Hooks for implementing security on multiple levels
 - Globally
 - Endpoint
- Documentation (Swagger)

WE CAN DO BETTER CONFIGURE YOUR ENDPOINT

- cRestDataset
- Drag and drop
- Entry_Item
- Fields are configurable



```
Object oVendorInventoryEndpoint is a cRestDataset
  Set psPath to "VendorInventory"
End_Object

Object oVendor_DD is a cVendorDataDictionary
End_Object

Object oInventory_DD is a cInventoryDataDictionary
  Set Constrain_file to Vendor.File_number
  Set DDO_Server to oVendor_DD
End_Object

Set Main_DD to oVendor_DD
Set Server to oVendor_DD

Object oVendor_Name is a cRestField
  Entry_Item Vendor.Name
End_Object

Object oVendor_Address is a cRestField
  Entry_Item Vendor.Address
End_Object

Object oVendor_City is a cRestField
  Entry_Item Vendor.City
End_Object

Object oVendor_State is a cRestField
  Entry_Item Vendor.State
End_Object
```


WE CAN DO BETTER TEST YOUR ENDPOINT

- OpenApi specification
- Try it out feature
- Example response
- Status codes
- Authentication

The screenshot shows an API Explorer interface for the endpoint `/ApiKey/VendorInventory`. The interface is divided into several sections:

- GET /ApiKey/VendorInventory**: A light blue header bar with the text "Get a list of all VendorInventory".
- POST /ApiKey/VendorInventory**: A green header bar with the text "Create a new VendorInventory".
- Parameters**: A section with a "Try it out" button and the text "No parameters".
- Request body**: A section with a dropdown menu set to "application/json".
- Example Value | Schema**: A dark grey box containing a JSON object:

```
{  "Name": "Vendor Name.",  "Address": "Street Address.",  "City": "City Name.",  "State": "Two letter state ID.",  "Inventory": [    {      "Description": "Inventory Part Description",      "Unit Price": "Retail unit price",      "On Hand": "Units currently available"    }  ]}
```
- Responses**: A table with columns "Code", "Description", and "Links".

Code	Description	Links
201	A VendorInventory	No links
- Media type**: A dropdown menu set to "application/json" with the text "Controls Accept header." below it.
- Example Value | Schema**: A dark grey box containing a JSON object (partially visible):

```
{  "Name": "Vendor Name.",
```

WE CAN DO BETTER CUSTOM BEHAVIOUR

- cCustomEndpoint
- No entry items
- OpenApi support
- OnDefineSchema
- OnHttpGet

```
Procedure OnDefineSchema tEndpointDefinition ByRef endpointDefinition
    tFieldDefinition imageField
    tVerbDefinition genericVerbInformation getVerb postVerb
    tParameterDefinition personIDParam

    //Configure the requested data
    Move "application/octet-stream" to genericVerbInformation.asRequestMediaTypes[-1]
    Move "image/jpeg" to genericVerbInformation.asRequestMediaTypes[-1]

    Move "image" to imageField.sFieldName
    Move REST_BINARY_FIELD to imageField.eFieldType
    Move True to imageField.bRequired

    Move imageField to genericVerbInformation.requestFields[0]

    //Configure the response data
    Move imageField to genericVerbInformation.responses[0].responseFields[-1]
    Move C_OK to genericVerbInformation.responses[0].iStatusCode
    Move "OK" to genericVerbInformation.responses[0].sStatusCodeDescription
    Move "application/octet-stream" to genericVerbInformation.responses[0].asResponseMediaTypes[-1]

    Move genericVerbInformation to getVerb
    Move genericVerbInformation to postVerb

    Move C_GET to getVerb.sVerb
    Move C_POST to postVerb.sVerb

    Move "Upload a image belonging to a person" to postVerb.sDescription

    Move "ID" to personIDParam.sName
    Move "ID of the person this image belongs to" to personIDParam.sDescription
    Move REST_QUERY_PARAMETER to personIDParam.eParameterIn
    Move REST_INTEGER_FIELD to personIDParam.eParameterType
    Move True to personIDParam.bRequired

    Move personIDParam to postVerb.parameters[-1]

    Move getVerb to endpointDefinition.verbDefinitions[-1]
    Move postVerb to endpointDefinition.verbDefinitions[-1]
End_Procedure
```

WE CAN DO BETTER MODIFIERS

- cHttpRequestModifier
- Middleware
- Extra functionality
- Logging
- Pre and Post request events

```
Object oRequestLogger is a cHttpRequestModifier
```

```
Object oApiLogs_DD is a cApiLogsDataDictionary  
End_Object
```

```
//Logs the incoming request time
```

```
Procedure OnPreRequest tHttpRequestContext ByRef httpRequestContext  
String sIncomingTime
```

```
Send Clear of oApiLogs_DD
```

```
Move (CurrentDateTime()) to sIncomingTime
```

```
Set Field_Changed_Value of oApiLogs_DD Field ApiLogs.Verb to httpRequestContext.sVerb
```

```
Set Field_Changed_Value of oApiLogs_DD Field ApiLogs.TimeReceived to sIncomingTime
```

```
Set Field_Changed_Value of oApiLogs_DD Field ApiLogs.Endpoint to httpRequestContext.sPath
```

```
End_Procedure
```

```
//Logs outgoing request time, status code and saves the record
```

```
Procedure OnPostRequest tHttpRequestContext ByRef httpRequestContext
```

```
Boolean bErr
```

```
String sResponseTime
```

```
//GetTickCount
```

```
Move (CurrentDateTime()) to sResponseTime
```

```
Set Field_Changed_Value of oApiLogs_DD Field ApiLogs.TimeResponse to sResponseTime
```

```
Set Field_Changed_Value of oApiLogs_DD Field ApiLogs.StatusCode to httpRequestContext.iStatusCode
```

```
Get Request_Validate of oApiLogs_DD to bErr
```

```
If (not(bErr)) Begin
```

```
Send Request_Save of oApiLogs_DD
```

```
End
```

```
End_Procedure
```

WE CAN DO BETTER AUTHENTICATION/AUTHORISATION

- cHttpAuthModifier
- Authentication
- OnDefineAuthRules
- OnAuth

```
Procedure OnDefineAuthRules Variant ByRef vSecurityStruct  
    bearerAuth bearerauth  
    Forward Send OnDefineAuthRules (&vSecurityStruct)  
  
    Move "http" to bearerauth.type  
    Move "bearer" to bearerauth.scheme  
    Move "JWT" to bearerauth.bearerFormat  
  
    Move bearerauth to vSecurityStruct  
End_Procedure
```

```
Procedure OnAuth tHttpContext ByRef httpcontext  
    String sBearerToken sErrorMessage  
    Boolean bIsValid  
    Integer iRetVal  
  
    Forward Send OnAuth (&httpcontext)  
  
    Get HttpRequestHeader "Authorization" to sBearerToken  
    Move (Replace("Bearer ", sBearerToken, "")) to sBearerToken  
  
    Get DecodeString of oJWT sBearerToken to iRetVal  
  
    If (iRetVal <> C_JWT_VALID) Begin  
        Move ('JWT was invalid for the following reason: ' + (ErrorCodeToErrorMessage(oJWT, iRetVal))) to sErrorMessage  
        Move True to httpcontext.bErr  
        Move C_UNAUTHORIZED to httpcontext.iStatusCode  
        Move "Unauthorized" to httpcontext.sShortStatusMessage  
        Move sErrorMessage to httpcontext.sErrorMessage  
        Procedure_Return  
    End  
End_Procedure
```

WE CAN DO BETTER

EXAMPLES

- Example workspace
- Three types of authentication/authorisation
- Example logging class
- One-to-many relationships (Child)
- One-to-one relationships (Parent)

03

DEMO!

04

FUTURE CONSIDERATIONS

FUTURE CONSIDERATIONS

FUTURE

- Wizards
- OAuth2 example