



A Close Look at Two Development Approaches to Interacting with FHIR APIs

Benjamin Xue | Microsoft



HL7 FHIR DevDays 2020, Virtual Edition, November 17–20, 2020 | @FirelyTeam | #fhirdevdays | www.devdays.com/november-2020

ORGANIZED BY **firely**



Who am I?

- Benjamin Xue
- Senior Program Manager at Microsoft



A Close Look at Two Development Approaches to Interacting with FHIR APIs

- Solution
 - Synchronize data between FHIR format and relational data
- First Approach
 - Mapping data schema using FHIR Path
 - Working with data using FHIR Path
- Second Approach
 - Mapping data schema using JSON Path
 - Working with data using JSON path

Note: C# code tested on Azure API for FHIR which uses Firely's HL7.Fhir packages

Mapping using FHIR Path

Data field in a relational table

emailaddress1

FHIR Path for Patient

```
Patient.telecom.where(system='email').select(value.first())
```

Code using FHIR Path

- Get value from FHIR resource to upsert database record

```
public static FhirClient InitializeFHIRClient(string accesstoken, string
resource)
{
    var _fhirClient = new FhirClient(resource);
    _fhirClient.PreferredFormat = HL7.Fhir.Rest.ResourceFormat.Json;

    _fhirClient.OnBeforeRequest += (object sender,
        BeforeRequestEventArgs e) =>
    {
        e.RawRequest.Headers.Add("Authorization", "Bearer " +
            accesstoken);
    };

    return _fhirClient;
}
```

```
//_m is a JObject containing mapping detail
var _fhirClient = FHIRResourceHelper.InitializeFHIRClient(fhiraccesstoken,
fhiraudience);
var _fhirClientResource = _fhirClient.Read<Patient>(_fhirRequestUrlResource);
if (_fhirClientResource == null) return null;
var _f = _fhirClientResource.ToTypedElement();

var _a0 = _m["fhirElementName"];
var _a1 = _f.Scalar((string)_a0);
...
```

Code using FHIR Path

- Get value from database record to upsert FHIR resource

```
var _fhirClient = FHIRResourceHelper.InitializeFHIRClient(fhiraccesstoken, fhiraudience);
Patient _patient = _fhirClient.Read<Patient>("Patient/" + _fhirResourceId);
...
if (_fhirElement == "Patient.telecom.where(system ='email').select(value.first())"){
    _count = 0;
    _entryfound = false;
    foreach (ContactPoint _cp in _patient.Telecom){
        if (_cp.System == ContactPoint.ContactPointSystem.Email) {
            _patient.Telecom[_count].Value = _cdsAttributeValue;
            _entryfound = true;
            break;};
        _count++;}
    if (!_entryfound) _patient.Telecom.Add(new ContactPoint(ContactPoint.ContactPointSystem.Email, null, "emailhere"));
```

Pros and Cons using FHIR Path

- Get value from FHIR resource to upsert database record
 - Easy to do
- Get value from database record to upsert FHIR resource
 - Easy if the FHIR Path is found
 - Difficult to construct a FHIR resource object if the FHIR Path is not found
 - FHIR resource object is strongly typed, for example Patient

Mapping Using JSON Path

- Patient's telecom | email ↔ database: emailaddress1

```

{
  "s": "$.telecom[?(@system=='email')].value",
  "c": {
    "p": "telecom[0]",
    "a": [
      {
        "system": "email"
      },
      {
        "value": "%"
      }
    ]
  }
}

```

Note:

s: select

[0]: first array element

c: create

p: parent

a: attribute

?: placeholder for value

Code Using JSON Path

- Get value from FHIR resource to upsert Database record

```
//_m is a JObject containing mapping detail  
//_fhirJsonObject is a JObject of a FHIR resource content  
//_cdsEntityObj is a JObject for a database table e.g. contact or patient
```

```
var _a0 = _m["fhirElementName"];  
_objElementMapping = JObject.Parse(_a0.ToString());  
var _selectPath = _objElementMapping["s"].ToString();  
var _a1 = _fhirJsonObject.SelectToken(_selectPath);  
var _a3 = _m["cdsAttributeName"];  
_cdsEntityObj.Add((string)_a3, _a1.ToString());
```

Code Using JSON Path

- Get values from database record to upsert FHIR resource

```
// objElementMapping is a object containing mapping detail for a specific FHIR resource element
//_cdsAttributeValue is value of a database field
//_fhirJObject is a object containing FHIR resource data
//_obj is a object used to store data which is later added to _fhirJObject

//scenario 1: Json path found - replace/update it
var _selectPath = _objElementMapping["s"].ToString(); //skipping "$." in defined mapping
_srchToken = _fhirJObject.SelectToken(_selectPath);
_fhirJObject.SelectToken(_selectPath).Replace(_cdsAttributeValue);
```

Code Using JSON Path

- Get values from database record to upsert FHIR resource

//scenario 2: Json path not found but it's a top-level property or its parent property exists, so create the property

```
var _objAttributesCreate = _objElementMapping["c"];
```

```
_fhirJObject.Add(new JProperty(_selectPath.Substring(2), _cdsAttributeValue));
```

```
var _newPropValue = _cdsAttributeValue;
```

```
JProperty _prop = (JProperty)(_a.Children().First());
```

```
_newPropName = _prop.Name; //code handling array element is not shown here
```

```
_obj.Add(new JProperty(_newPropName, _newPropValue));
```

//scenario 3: parent element not found, so create the parent element (single element vs. array), and

//create child element, one level or multiple levels, depending on the FHIR resource type

//There are code examples on how to create and work with jobject, jarray and jproperty

Pros and Cons using JSON Path

- Get values from FHIR resource to upsert database record
 - Easy to do
 - Code not specific to FHIR resource type
- Get values from database record to upsert FHIR resource
 - Easy if the Json Path is found
 - Doable but requiring more mapping detail

FHIR Path vs. JSON Path

- Mapping is key
- Either approach works but complexity varies

Contact

- During DevDays, you can find / reach me here:
 - Via Whova App – Speaker’s Gallery
 - Email: benjamin.xue@microsoft.com
 - Twitter: [zxue](https://twitter.com/zxue)

Q&A