DevDays HL7® FHIR®

Hacking FHIR Server for Azure

Brendan Kowitz, Microsoft

November 20-22, Amsterdam | @HL7 @FirelyTeam | #fhirdevdays | www.devdays.com

HL7®, FHIR® and the flame Design mark are the registered trademarks of Health Level Seven International and are used with permission.

Agenda

- Overview of the FHIR Server for Azure
- Hacking and extending
  - Add a data layer
  - Add a feature
- Contributing and what to expect

Two offerings to meet your needs

Open source, full control

Built managed and maintained by the Microsoft Healthcare engineering team.

Designed to enable you to operate and manage compliance.

FHIR Server for Azure

Fully managed Azure offering

Distribution of the open source project, offered as a Platform-as-a-Service in Azure.

Meets regulatory compliance requirements for Protected Healthcare Information (PHI).

Azure API for FHIR

FHIR Server for Azure

Open source implementation of the industry HL7 Fast Healthcare Interoperability Resources (FHIR) specification.

Supports FHIR STU3 and FHIR R4 versions.

Easily deployable into your Azure subscription

An extensible persistence provider model enables multiple options for how data is stored, including Azure Cosmos DB and Azure SQL.

http://github.com/Microsoft/fhir-server

Azure API for FHIR

Fully managed, enterprise-grade FHIR service. Provisioned in the cloud in less than 5 minutes

Enables bringing together health data from disparate systems using industry standard HL7 FHIR (Fast Healthcare Interoperability Resources).

Supports FHIR STU3 and FHIR R4 versions.

Data is isolated and protected with layered, in-depth defense and advanced threat protection according to the most stringent industry compliance standards.

FHIR Server for Azure features

DevDays 2018

- Stu3
- CosmosDB data layer
- Basic search support

DevDays 2019

- R4
- SQL data layer
  - Advanced search (include, chained)
- Bulk export
- Conditional create/update
- Bundle Batch processing
- ?_pretty, $versions
- …

https://github.com/microsoft/fhir-server/releases

Supported features in hosted solution?

Quantity	Yes	Issue #103
URI	Yes	
Special	No	

Modifiers	Supported	Comment
`:missing`	Yes	
`:exact`	Yes	
`:contains`	Yes	
`:text`	Yes	
`:in` (token)	No	
`:below` (token)	No	
`:above` (token)	No	
`:not-in` (token)	No	

Azure API for FHIR

https://docs.microsoft.com/en-us/azure/healthcare-apis/fhir-features-supported

Let’s hack a data layer!

Architectural components

Implementing a Data Layer

Search provider

/Patient?name=smith&address=Seattle

Name	Value
searchOptions.Expression	{(And (Param _type (StringEquals TokenCode 'Patient')) (Param name (StringStartsWithIgnoreCase String 'smith')) (Paran
Expressions	{Microsoft.Health.Fhir.Core.Features.Search.Expressions.Expression[3]}
[0]	{(Param _type (StringEquals TokenCode 'Patient'))}
[1]	{(Param name (StringStartsWithIgnoreCase String 'smith'))}
[2]	{(Param address (StringStartsWithIgnoreCase String 'Seattle'))}
MultiaryOperation	And

T-SQL, CosmosDB SQL, *TableStorage Query*, etc…

Adding a Data Layer demo

- Add an empty data layer, start with an empty capability statement


```
{
    "resourceType": "CapabilityStatement",
    "url": "/metadata",
    "version": "1.0.0.0",
    "name": "Microsoft FHIR Server",
    "status": "draft",
    "experimental": true,
    "publisher": "Microsoft",
    "contact": [
        {
            "telecom": [
                {
                    "system": "url",
                    "value": "https://www.microsoft.com"
                }
            ]
        }
    ],
    "kind": "capability",
    "software": {
        "name": "Microsoft FHIR Server",
        "version": "1.0.0.0"
    },
    "fhirVersion": "4.0.0",
    "format": [
        "application/fhir+json",
        "application/fhir+xml"
    ],
    "rest": [
        {
            "mode": "server"
        }
    ]
}
```

- Add capabilities as required with builder functions

```
public void Build(ICapabilityStatementBuilder builder)
{
    builder.AddRestInteraction("Observation", TypeRestfulInteraction.Read);
}
```


```
{
    "resourceType": "CapabilityStatement",
    "url": "/metadata",
    "version": "1.0.0.0",
    "name": "Microsoft FHIR Server",
    "status": "draft",
    "experimental": true,
    "publisher": "Microsoft",
    "contact": [ … ], // 1 item
    "kind": "capability",
    "software": { … }, // 2 items
    "fhirVersion": "4.0.0",
    "format": [ … ], // 2 items
    "rest": [
        {
            "mode": "server",
            "resource": [
                {
                    "type": "Observation",
                    "interaction": [
                        {
                            "code": "read"
                        }
                    ],
                    "versioning": "versioned",
                    "conditionalCreate": false,
                    "conditionalUpdate": false
                },
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
```

- Add capabilities as required with builder functions

```
public void Build(ICapabilityStatementBuilder builder)
{
    builder.AddRestInteraction("Observation", TypeRestfulInteraction.Read);
}
```


```
{
    "resourceType": "CapabilityStatement",
    "url": "/metadata",
    "version": "1.0.0.0",
    "name": "Microsoft FHIR Server",
    "status": "draft",
    "experimental": true,
    "publisher": "Microsoft",
    "contact": [ … ], // 1 item
    "kind": "capability",
    "software": { … }, // 2 items
    "fhirVersion": "4.0.0",
    "format": [ … ], // 2 items
    "rest": [
        {
            "mode": "server",
            "resource": [
                {
                    "type": "Observation",
                    "interaction": [
                        {
                            "code": "read"
                        }
                    ],
                    "versioning": "versioned",
                    "conditionalCreate": false,
                    "conditionalUpdate": false
                },
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
                { … }, // 5 items
```

- Use defaults to add all known interactions, models and search parameters

```
public void Build(ICapabilityStatementBuilder builder)
{
    builder.AddDefaultResourceInteractions()
        .AddDefaultSearchParameters();
}
```


```
"searchParam": [
  { … }, // 3 items
  {
    "name": "_id",
    "definition": "http://hl7.org/fhir/SearchParameter/Resource-id",
    "type": "token",
    "documentation": "Logical id of this artifact"
  },
  {
    "name": "_lastUpdated",
    "definition": "http://hl7.org/fhir/SearchParameter/Resource-lastUpdated",
    "type": "date",
    "documentation": "When the resource version last changed"
  },
  {
    "name": "_profile",
    "definition": "http://hl7.org/fhir/SearchParameter/Resource-profile",
    "type": "reference",
    "documentation": "Profiles this resource claims to conform to"
  },
  {
    "name": "_security",
    "definition": "http://hl7.org/fhir/SearchParameter/Resource-security",
    "type": "token",
    "documentation": "Security Labels applied to this resource"
  },
```

- Simple file/bundle based data store

```
namespace SamplesFileStorageProvider
{
    public sealed class FileStorageDataProvider : IFhirDataStore, IProvideCapability, IStartable
    {
        private readonly string _folderPath;
        private readonly ISearchIndexer _searchIndexer;
        private readonly IModelInfoProvider _modelInfoProvider;
        private readonly JsonSerializer _serializer = new JsonSerializer();

        public FileStorageDataProvider(
            IOptions<FileStorageSettings> settings,
            ISearchIndexer searchIndexer,
            IModelInfoProvider modelInfoProvider)...

        public Task<UpsertOutcome> UpsertAsync(ResourceWrapper resource, WeakETag weakETag,
            bool allowCreate, bool keepHistory, CancellationToken cancellationToken)...

        public Task<ResourceWrapper> GetAsync(ResourceKey key, CancellationToken cancellationToken)...

        public Task HardDeleteAsync(ResourceKey key, CancellationToken cancellationToken)...

        public ITransactionScope BeginTransaction()...

        public void Build(ICapabilityStatementBuilder builder)...

        public void Start()...
    }
}
```

- Simple file/bundle based data store

```
public Task<ResourceWrapper> GetAsync(ResourceKey key, CancellationToken cancellationToken)
{
    // ...
    if (entry.Location != null)
    {
        // ...

        return Task.FromResult(new ResourceWrapper(
            key.Id,
            key.VersionId ?? "1",
            key.ResourceType,
            new RawResource(token.ToString(), FhirResourceFormat.Json),
            new ResourceRequest(HttpMethod.Post),
            new FileInfo(entry.Location.FilePath).LastWriteTime.ToUniversalTime(),
            false,
            null,
            null,
            null));
    }

    throw new ResourceNotFoundException("Resource not found");
}
```

GET https://localhost:44348/Patient/f2343d85-50d8-4ec2-a779-194b3800b87e

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests | Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body | Cookies | Headers (8) | Test Results — Status: 200 OK Time: 59ms

Pretty | Raw | Preview | Visualize BETA | JSON


```
1  {
2      "resourceType": "Patient",
3      "id": "f2343d85-50d8-4ec2-a779-194b3800b87e",
4      "meta": {
5          "versionId": "1",
6          "lastUpdated": "2019-06-05T18:55:39+00:00",
7          "profile": [
8              "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
9          ]
10     },
11     "text": {
12         "status": "generated",
13         "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Generated by <a href=\"https://github.c
14     },
15     "extension": [
16         {
17             "extension": [
18                 {
```

- Simple in-memory search provider. Uses delegates to interpret AST

```
public class InMemorySearchProvider : SearchService
{
    private readonly IFhirDataStore _fhirDataStore;

    public InMemorySearchProvider(ISearchOptionsFactory searchOptionsFactory, IFhirDataStore fhirDataStore)...

    protected override async Task<SearchResult> SearchInternalAsync(SearchOptions searchOptions, CancellationToken cancellationToken)
    {
        (ResourceLocation, IReadOnlyCollection<SearchIndexEntry>)[] results;

        if (searchOptions.Expression != null)
        {
            var interpreter = new SearchQueryInterpreter();

            var searchFunc = searchOptions.Expression.AcceptVisitor(interpreter, default);

            results = searchFunc.Invoke(InMemoryIndex.Index.Values).ToArray();
        }
        else
        {
            results = InMemoryIndex.Index.Values.Take(10).ToArray();
        }

        var resources = await Task.WhenAll(
            results.Select(async x =>
            new SearchResultEntry(
                await _fhirDataStore.GetAsync(
                    new ResourceKey(x.Item1.ResourceType, x.Item1.ResourceId), cancellationToken))));

        return new SearchResult(
            resources,
            searchOptions.UnsupportedSearchParams,
            searchOptions.UnsupportedSortingParams,
            null);
    }
```

Not Secure | https://localhost:44348/Observation?value-quantity=lt100&code=8302-2&_format=json&_pretty=true


```
{
    "resourceType": "Bundle",
    "id": "7ec2735e-b33b-4974-9cbc-56771830191f",
    "meta": {
        "lastUpdated": "2019-11-17T01:15:53.836607+00:00"
    },
    "type": "searchset",
    "link": [
        {
            "relation": "self",
            "url": "https://localhost:44348/Observation?value-quantity=lt100&code=8302-2&_format=json"
        }
    ],
    "entry": [
        {
            "fullUrl": "https://localhost:44348/Observation/a7cd2527-a1b0-48c6-9683-c45e8663f8c3",
            "resource": {
                "resourceType": "Observation",
                "id": "a7cd2527-a1b0-48c6-9683-c45e8663f8c3",
                "meta": {
                    "versionId": "1",
                    "lastUpdated": "2019-06-05T18:55:39+00:00"
                },
                "status": "final",
```

- Simple in-memory search provider. Uses delegates to interpret AST

```
internal class SearchQueryInterpreter : IExpressionVisitorWithInitialContext<SearchQueryInterpreter.Context, SearchPredicate>
{
    Context IExpressionVisitorWithInitialContext<Context, SearchPredicate>.InitialContext => default;

    public SearchPredicate VisitSearchParameter(SearchParameterExpression expression, Context context)...

    public SearchPredicate VisitBinary(BinaryExpression expression, Context context)...

    private static bool GetMappedValue(BinaryOperator expressionBinaryOperator, ISearchValue first, IComparable second)...

    public SearchPredicate VisitChained(ChainedExpression expression, Context context)...

    public SearchPredicate VisitMissingField(MissingFieldExpression expression, Context context)...

    public SearchPredicate VisitMissingSearchParameter(MissingSearchParameterExpression expression, Context context)...

    public SearchPredicate VisitMultiary(MultiaryExpression expression, Context context)...

    public SearchPredicate VisitString(StringExpression expression, Context context)...

    public SearchPredicate VisitCompartment(CompartmentSearchExpression expression, Context context)...

    public SearchPredicate VisitInclude(IncludeExpression expression, Context context)...

    private SearchPredicate AppendSubquery(string parameterName, Expression expression, Context context, bool negate = false)...

    /// <summary> Context that is passed through the visit.
    internal struct Context...
}
```

Implementing an operation ($validate)

Asp.net MVC Core, Controller / Action

Message Handler

http://hl7.org/fhir/resource-operation-validate.html

Adding an operation demo

- Add controller action for $validate

```
[HttpPost]
[Route(KnownRoutes.ResourceType + "$validate")]
[AuditEventType(AuditEventSubType.Read)]
[Authorize(PolicyNames.ReadPolicy)]
0 references
public async Task<IActionResult> Validate([FromBody] Resource resource)
{
    var response = await _mediator.Send<ValidateOperationResponse>(new ValidateOperationRequest(resource.ToResourceElement()));

    if (resource.ResourceType == ResourceType.Parameters)
    {
        var outcome = new OperationOutcome();
        outcome.Issue.Add(new OperationOutcome.IssueComponent
        {
            Severity = OperationOutcome.IssueSeverity.Error,
            Code = OperationOutcome.IssueType.NotSupported,
            Diagnostics = "Not supported",
        });

        return FhirResult.Create(outcome.ToResourceElement());
    }

    return FhirResult.Create(new OperationOutcome
    {
        Issue = response.Issues.Select(x => x.ToPoco()).ToList(),
    }.ToResourceElement());
}
```

- Add controller action for $validate

```
[HttpPost]
[Route(KnownRoutes.ResourceType + "$validate")]
[AuditEventType(AuditEventSubType.Read)]
[Authorize(PolicyNames.ReadPolicy)]
0 references
public async Task<IActionResult> Validate([FromBody] Resource resource)
{
    var response = await _mediator.Send<ValidateOperationResponse>(new ValidateOperationRequest(resource.ToResourceElement()));

    if (resource.ResourceType == ResourceType.Parameters)
    {
        var outcome = new OperationOutcome();
        outcome.Issue.Add(new OperationOutcome.IssueComponent
        {
            Severity = OperationOutcome.IssueSeverity.Error,
            Code = OperationOutcome.IssueType.NotSupported,
            Diagnostics = "Not supported",
        });

        return FhirResult.Create(outcome.ToResourceElement());
    }

    return FhirResult.Create(new OperationOutcome
    {
        Issue = response.Issues.Select(x => x.ToPoco()).ToList(),
    }.ToResourceElement());
}
```

- Add rules in the built-in validation layer

```
public class ValidateResourceOperationValidator : AbstractValidator<ValidateOperationRequest>
{
    0 references
    public ValidateResourceOperationValidator(INarrativeHtmlSanitizer htmlSanitizer, IModelAttributeValidator modelAttributeValidator)
    {
        RuleFor(x => x.Resource).SetValidator(new ResourceValidator(htmlSanitizer, modelAttributeValidator));
    }
}
```

- Add success response to $validate handler

```
public class ValidateOperationHandler : IRequestHandler<ValidateOperationRequest, ValidateOperationResponse>
{
    16 references
    public Task<ValidateOperationResponse> Handle(ValidateOperationRequest request, CancellationToken cancellationToken)
    {
        return Task.FromResult(new ValidateOperationResponse(
            new OperationOutcomeIssue(
                OperationOutcomeConstants.IssueSeverity.Information,
                OperationOutcomeConstants.IssueType.Informational,
                "OK")));
    }
}
```

- Resources posted to $validate will now be run through internal validation pipeline

POST {{FhirServer}}/Patient$validate Send Save

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON


```
1 {
2   "resourceType": "Patient",
3   "id": "example!!",
4   "text": {
5     "status": "generated",
6     "div": "<p><div xmlns=\"http://www.w3.org/1999/xhtml\">\n\t\t\t<table>\n\t\t\t\t<tbody>\n\t\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Name
        </td>\n\t\t\t\t\t\t<td>Peter James \n              <b>Chalmers</b> (&quot;Jim&quot;)\n            </td>\n\t\t\t\t\t</tr
        >\n\t\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Address</td>\n\t\t\t\t\t\t<td>534 Erewhon, Pleasantville, Vic, 3999</td>\n\t\t\t\t
        </tr>\n\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Contacts</td>\n\t\t\t\t\t\t<td>Home: unknown. Work: (03) 5555 6473</td
        >\n\t\t\t\t</tr>\n\t\t\t\t<tr>\n\t\t\t\t\t\t<td>Id</td>\n\t\t\t\t\t\t<td>MRN: 12345 (Acme Healthcare)</td
```


Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 859ms Size: 848 B Save Response

Pretty Raw Preview Visualize BETA JSON


```
1 {
2     "resourceType": "OperationOutcome",
3     "id": "e96f405a-5c3f-41a8-bb72-d391d358e230",
4     "issue": [
5         {
6             "severity": "error",
7             "code": "invalid",
8             "diagnostics": "Id must be any combination of upper or lower case ASCII letters ('A'..'Z', and 'a'..'z'), numerals (
9             "location": [
10                "Resource.Id"
11            ]
12        },
13        {
```

Deploy into AppService

Deploying the FHIR Server Template

To deploy the backend Cosmos DB, Azure Web App, and FHIR server code, use the buttons below to deploy through the Azure Portal. If you would like to protect the FHIR API with token authorization, you will need to supply application registration details as described above.

Deploy to Azure | Deploy to Azure Gov

The template can also be deployed using PowerShell. Here is an example of how the authorization details from above can be provided:

```
$rg = New-AzResourceGroup -Name "RG-NAME" -Location westus2

New-AzResourceGroupDeployment `
-TemplateUri "https://raw.githubusercontent.com/Microsoft/fhir-server/master/samples/templates/default-a
-ResourceGroupName $rg.ResourceGroupName `
-serviceName $fhirServiceName `
-securityAuthenticationAuthority $apiAppReg.Authority `
-securityAuthenticationAudience $apiAppReg.Audience
```

https://github.com/microsoft/fhir-server/blob/master/docs/DefaultDeployment.md

Issues and up for grabs

16 Open ✓ 15 Closed Author Labels Projects Milestones Assignee Sort

_type search parameter should move in metadata Bug Up for grabs
#723 opened 6 days ago by brandonpollett

ValidateResourceIdFilterAttribute returns 'TypeName' in error message location instead of 'Id' Bug Up for grabs
#664 opened on Sep 23 by brendankowitz

Deployment template fails in Azure Gov with App Insights Bug Up for grabs
#662 opened on Sep 21 by hansenms

Validation messages need reformatting Bug Up for grabs
#660 opened on Sep 20 by rotodd

AadSmartOnFhirProxyController.cs logs a caught null redirect URL as a NullRef exception Bug Up for grabs
#628 opened on Aug 27 by feordin

Requesting token from AadSmartOnFhirProxy fails fo the token "scope" is null Bug Up for grabs
#615 opened on Aug 15 by namalu
4

ArgumentException() thrown when search request contains "-_count" Area-Search Bug Up for grabs
#580 opened on Jul 12 by jackliums
3

Contributing!

Fork!

Branch and code

Submit a PR

Build system

microsoft / fhir-server

Unwatch 65 | Unstar 346 | Fork 114

Code | Issues 85 | Pull requests 13 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Added null check for missing scopes #704

Edit

Merged hansenms merged 4 commits into `master` from `personal/mihansen/missing-scope-bug` 15 days ago

Conversation 9 | Commits 4 | Checks 3 | Files changed 2 | +55 −12

response to review comments b43673f

Azure Pipelines

- PR Build
- PR Build (Linux)
- PR Build (Windows)

Azure Pipelines / PR Build
succeeded 17 days ago in 9m 55s

Build #pr704-20191026-1 succeeded

0 errors / 0 warnings

DETAILS

Tests

- Failed: 0 (0.00%)
- Passed: 4,524 (100.00%)
- Other: 0 (0.00%)
- Total: 4,524

View more details on Azure Pipelines

Going forward

- We'll continue to add features based on community and partner input
- We invite you to collaborate on features
- Examples
  - GraphQL
  - Advanced search
  - New $operators

Recap

- Overview of the FHIR Server for Azure
- Hacking and extending
  - Add a data layer
  - Add a feature
- Contributing and what to expect

Questions?

- @brendankowitz
  #fhirserverforazure
- https://github.com/brendankowitz/fhir-server/tree/devdays/demo-2019
- http://github.com/Microsoft/fhir-server
- https://stackoverflow.com/questions/tagged/fhir-server-for-azure
- http://aka.ms/fhir
- Or see us at the tables

DevDays HL7® FHIR®

www.devdays.com