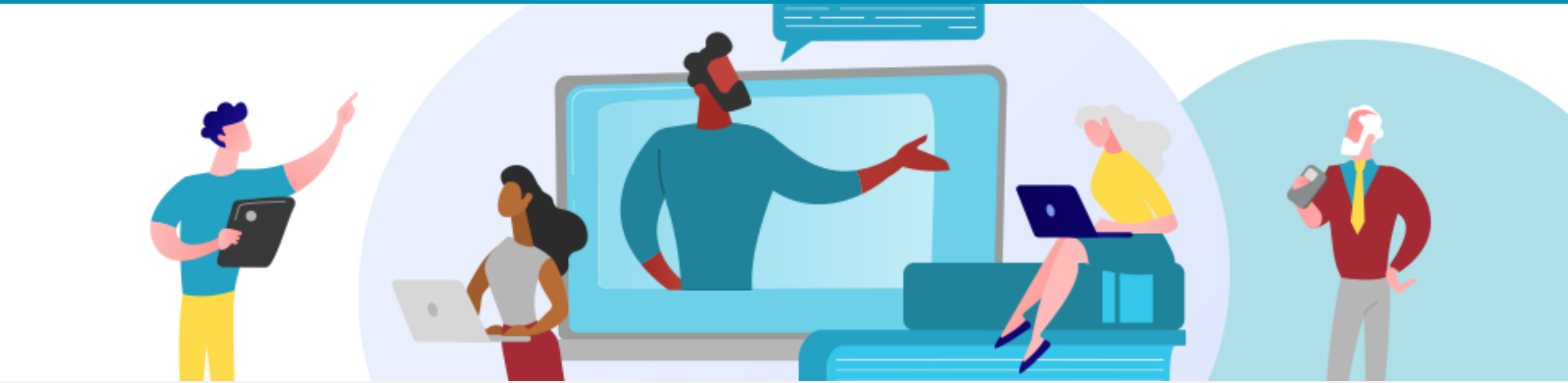
HL7® FHIR® DevDays

JUNE | 2021

Lessons learned with Fhir Paging and with Plain-Server/Façade Fhir Servers and Load Balanced Systems

Sloan Holliday, Optum



HL7 FHIR DevDays 2021, Virtual Edition, June 7–10, 2021 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY firely HL7 International | PARTNER HL7® FHIR® FOUNDATION

Who am I?

- Sloan Holliday
- Sloan Holliday, Lead FHIR Developer @ Optum

Sloan Holliday

Lead FHIR Developer
Optum

Description

Sloan Holliday is a Principal Interoperability Developer with Optum, a part of UnitedHealth Group. He is currently with the Optum Enterprise Clinical Integrations and Interoperability team. Sloan has more than 20 years of experience as a software developer utilizing technologies such as .NET, .NET-Core, Java, MsSqlServer, Azure and more.
Sloan has worked in varied background in the healthcare industry including payer, provider, and HIE software. He is an active member in the local developer user groups and is an active contributor on sites like StackOverflow.

Agenda

- Basic “Anatomy”
- FHIR Server as a Façade Observations
- Hapi FHIR “Plain Server” (aka, Façade) observations

(Questions at the end please) #15MinutesAndRabbitTrailsRNotFriends

20 Slides, 15 minutes, LET’S GO !

*** This is not an “Introduction” to REST Paging.

Basic Anatomy

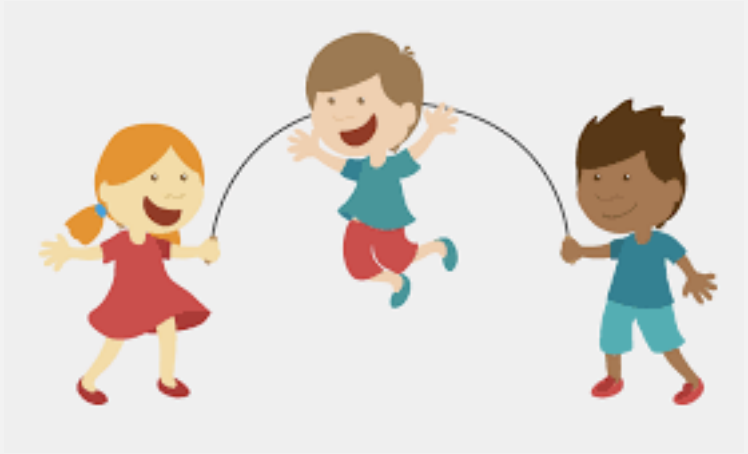
- Requests

  https://vonk.fire.ly/R4/Organization/?_count=10

  http://wildfhir4.aegis.net/fhir4-0-0/Organization/?_count=10

  http://hapi.fhir.org/baseR4/Organization?_count=10

Basic Anatomy



- Response (Vonk)


```
{
  "resourceType": "Bundle",
  "type": "searchset",
  "total": 194,
  "link": [
  {
    "relation": "self",
    "url": "https://server.fire.ly/R4/Organization?_sort=-_lastUpdated&_count=10&_skip=0"
  },
  {
    "relation": "next",
    "url": "https://server.fire.ly/R4/Organization?_sort=-_lastUpdated&_count=10&_skip=10"
    },
  {
    "relation": "last",
    "url": "https://server.fire.ly/R4/Organization?_sort=-_lastUpdated&_count=10&_skip=190"
  }
 ],
 "id": "034773c2-9e1a-428b-97c4-9c0b7af8ee3f",
 "entry": [
```

Basic Anatomy

- Response (wildfhir4.aegis)


```
{
    "resourceType": "Bundle",
    "id": "9a36e9fe-f049-4eaa-96c3-cfab527dd5ca",
    "type": "searchset",
    "total": 35,
    "link": [
        {
            "relation": "self",
            "url": "http://wildfhir4.aegis.net/fhir4-0-0/Organization?_count=10&page=1"
        },
        {
            "relation": "first",
            "url": "http://wildfhir4.aegis.net/fhir4-0-0/Organization?_count=10&page=1"
        },
        {
            "relation": "next",
            "url": "http://wildfhir4.aegis.net/fhir4-0-0/Organization?_count=10&page=2"
        },
        {
            "relation": "last",
            "url": "http://wildfhir4.aegis.net/fhir4-0-0/Organization?_count=10&page=4"
        }
    ],
    "entry": [
```

Basic Anatomy

- Response (hapi.fhir.org/baseR4)


```
{
    "resourceType": "Bundle",
    "id": "a78ac4cb-8d7d-446f-95c9-7e47c04e2648",
    "type": "searchset",
    "link": [
        {
            "relation": "self",
            "url": "http://hapi.fhir.org/baseR4/Organization"
        },
        {
            "relation": "next",
            "url": "http://hapi.fhir.org/baseR4?_getpages=a78ac4cb-8d7d-446f-95c9-7e47c04e2648
                    &_getpagesoffset=20&_count=20&_pretty=true&_bundletype=searchset"
        }
    ],
    "entry": [
        {
            "fullUrl": "http://hapi.fhir.org/baseR4/Organization/1990486",
```

Basic Anatomy (next)

- Response (compare the three, fire.ly, aegis, hapi.fhir.org)

```
{
   "relation": "next",
   "url": "https://server.fire.ly/R4/Organization?_sort=-_lastUpdated&_count=10&_skip=10"
   },


{
   "relation": "next",
   "url": "http://wildfhir4.aegis.net/fhir4-0-0/Organization?_count=10&page=2"
},

{
    "relation": "next",
    "url": "http://hapi.fhir.org/baseR4?_getpages=a78ac4cb-8d7d-446f-95c9-
7e47c04e2648&_getpagesoffset=20&_count=20&_pretty=true&_bundletype=searchset"
       }
```

FHIR Server as a Façade Observations

- The “total count” is a very important value.

```
{
    "resourceType": "Bundle",
    "id": "9a36e9fe-f049-4eaa-96c3-cfab527dd5ca",
    "meta": {
        "versionId": "1",
        "lastUpdated": "2021-05-28T07:10:22.047-04:00"
    },
    "type": "searchset",
     "total": 35,
```

(not shown is that there are 10 “entries” for the current page)

https://www.hl7.org/fhir/bundle-definitions.html#Bundle.total

Element Id	Bundle.total
Definition	If a set of search matches, this is the total number of entries of type 'match' **across all pages in the search**. It does not include search.mode = 'include' or 'outcome' entries and it does not provide a count of the number of entries in the Bundle.
Cardinality	0..1

FHIR Server as a Façade Observations

- The "total count" is a very important value.
  As a metaphor, in RDBMS terms, you kind of need **both**:

  Select COUNT(*) as AccurateOverallCount from Organization where ____

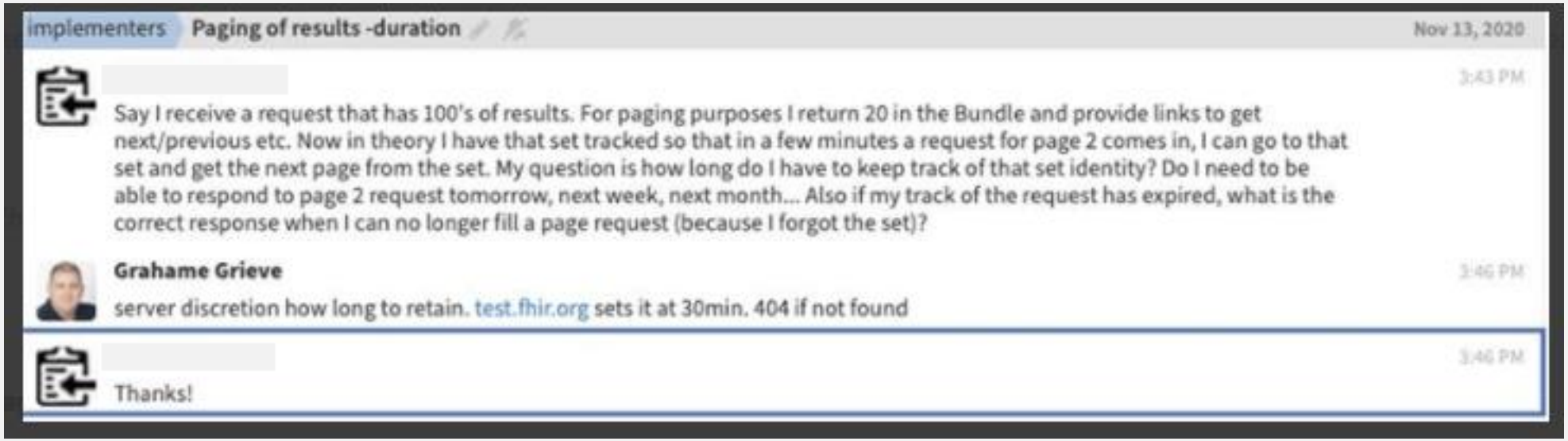
  and (for first page)

  Select (Page1, 10 Rows) from Organization where____

  ("TOP" or "Limit/Skip")

FHIR Server as a Façade Observations

- How long does a ”paging” result last?



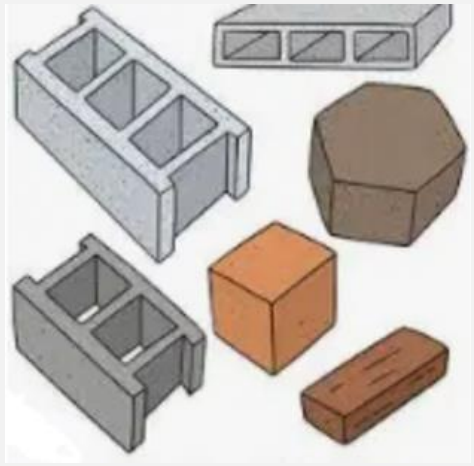
FHIR Server as a Façade Observations

- Default and Max Counts

https://vonk.fire.ly/R4/Organization/? << What is your default count? 10, 50, 100?

https://vonk.fire.ly/R4/Organization/?_count=5555555 << What is your maximum count? 100, 500, 1000?

Hapi Fhir Observations



Hapi Fhir Observations (Documentation)

➢ https://hapifhir.io/hapi-fhir/docs/server_plain/paging.html

- **Paging Providers**
  To support paging, a server **must** have an IPagingProvider implementation set.
  The paging provider is used to store resource return lists *between incoming calls* by clients.

➢ "**must**" ??

Hapi Fhir Observations (**Paging Providers**)

- Short Version:

- If you do not set a paging-provider in Hapi (or set it to null), you get paging links like (#simpleRest):

```
"relation": "next",
"url": "/myr4fhirserver/Patient?_count=13&_offset=13"
```

- If you do set a paging-provider in Hapi, you get paging links like: ***

```
"relation": "next",
"url": "/myr4fhirserver?_getpages=e66cfeef-42e2-48af-8a1b-
0cbce9e371db&_getpagesoffset=13&_count=13&_pretty=true&_bundletype=searchset"
```

Note, the second example does NOT have the /MyFhirResource/ in the URL (as the simple one does)
*** hapi.fhir.org/baseR4 is using this version, but probably is not a plain server implementation

Load Balance Issues

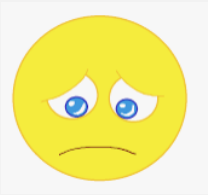
- Remember from earlier : “The paging provider is used to store resource return lists *between incoming calls* by clients.”
- To support “between incoming calls” (“next”, etc), you need to be able to persist “something” to “somewhere”.
  With a Load Balance system, that “somewhere” could be Redis or a RDBMS.
  But (ding ding ding), the “something” must be SERIALIZABLE.
- https://hapifhir.io/hapi-fhir/docs/server_plain/paging.html
- HAPI FHIR contains couple of implementations for IPagingProvider.

- **DatabaseBackedPagingProvider**
  When using DatabaseBackedPagingProvider HAPI FHIR searches may be done asynchronously. This means that the result is also cached to the database and the client may base the cached search result set.
- **FifoMemoryPagingProvider**
  When using FifoMemoryPagingProvider HAPI FHIR server search is persisted on the server memory and when pages are fetched the server returns the results from the cached memory (unless the cache overflowed and the old result set is no longer available).

Load Balance Issues

➢FifoMemoryPagingProvider

When using FifoMemoryPagingProvider HAPI FHIR server search is persisted on the server memory and when pages are fetched the server returns the results from the cached memory.

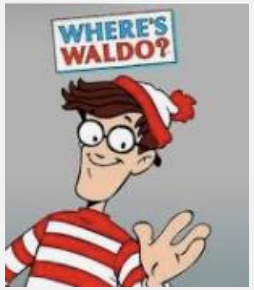


The issue here is that “in memory” runs on ***one*** of the servers/k8.pods,
so when the “next” request comes in, it may not hit the same machine. #sticky

Load Balance Issues

➢DatabaseBackedPagingProvider

Looks Promising, *but* ->




```
public class DatabaseBackedPagingProvider extends BasePagingProvider {

        @Autowired
        private DaoRegistry myDaoRegistry;
        @Autowired
        private SearchBuilderFactory mySearchBuilderFactory;
        @Autowired
        private PersistedJpaBundleProviderFactory myPersistJpaBundleProviderFactory
```

```
public class PersistedJpaBundleProviderFactory {
        @Autowired
        private ApplicationContext myApplicationContext;
```

```
public class SearchBuilderFactory {
        @Autowired
        private ApplicationContext myApplicationContext;
        @Autowired
        private DaoConfig myDaoConfig;
```

Load Balance Issues (Alternate Idea)

- **Custom IBundleProvider**
- **Custom IPagingProvider**

Load Balance Issues (Alternate Idea)

- **Custom IBundleProvider, IPagingProvider**
- **Why? (Motivation)**

There is a *difference* between caching the “search criteria” versus caching the “results”.
We did not want to be responsible for having a redis cache full of PHI data.

Load Balance Issues (Custom IBundleProvider)

```
import ca.uhn.fhir.rest.api.server.IBundleProvider;

public final class IocPayAsYouGoFilterArgsBundleProvider<T extends Serializable> implements IBundleProvider, Serializable {

   /* Must be the TRUE overall size of the resources. */
   private Integer overallSize;

   private T filterArgs; // << Serializable-Pojo holding the "filters" or any paging needs

   /* this MUST be transient to avoid trying to .serialize the entire Ioc Graph */
   private transient org.springframework.context.ApplicationContext iocContext;

   /* track the ioc-type-name in order to pull out the concrete type from the iocContext. */
   private final String iocTypeName; /* Java Developers like to call this the Spring-Bean(Name) */

   @Override
   public List<IBaseResource> getResources(int fromIndex, int toIndex) {
       /* fish out of the Ioc the "bean" that drives paging for this Fhir Resource */
       Class<?> clazz = Class.forName(iocTypeName);
       foundIocItem = this.iocContext.getBean(clazz); /* Not Sonar Friendly :< */

       IFhirPagingMarker castObject = (IFhirPagingMarker) foundIocItem;
       List<IBaseResource> returnItems = castObject.executePaging(fromIndex, toIndex, this.filterArgs);
       return returnItems;
   }

   public void setIocContext(ApplicationContext iocContext) {
      this.iocContext = iocContext;
   }
}
```

```
public interface IFhirPagingMarker<V extends Serializable> {
   List<IBaseResource> executePaging(int fromIndex,
   int toIndex,
   PagingRequestArgsV<V> filterArgs);
}
```

Load Balance Issues (Custom IPagingProvider)

```
import ca.uhn.fhir.rest.api.server.IBundleProvider;
import ca.uhn.fhir.rest.api.server.RequestDetails;
import ca.uhn.fhir.rest.server.IPagingProvider;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;

public final class IocCachePagingProvider implements IPagingProvider, ApplicationContextAware {

   private ApplicationContext iocContext;

   @Override
   public IBundleProvider retrieveResultList(RequestDetails reqDetails, String theId) {
      /* obviously, the responsibility here is to retrieve an IBundleProvider, based on "theId" from  Redis or RDBMS */
      /* it should be a IocPayAsYouGoFilterArgsBundleProvider when it comes out, so type-check it and cast it */
      /* the "trick" here is that after you retrieve it and cast it, you can call the IocPayAsYouGoFilterArgsBundleProvider.setIocContext using "this.iocContext" */
      myIocPayAsYouGoFilterArgsBundleProvider.setIocContext(this.iocContext);
   }

   @Override
   public String storeResultList(RequestDetails reqDetails, IBundleProvider ibp) {
       /* obviously, the responsibility here is to save an IBundleProvider.  in our case, it will be a IocPayAsYouGoFilterArgsBundleProvider */
      /* the "trick" here is that you save it to either Redis or RDBMS, but to save it, it must be serializable */
   }

   /* java DI trick. this is defined by ApplicationContextAware interface */
   @Override
   public void setApplicationContext(ApplicationContext ioc) throws BeansException {
      this.iocContext = ioc;
   }
}
```

Time/Space/Continuum Issues

- This is an in-general “REST” issues. If you have heavy writes/deletes, the “page” can “move” on you.

- One idea. When you get the first request for paging, you can create a list of back-end-surrogate-keys, and keep them in Dictionary/Map with the key being the ”index” (fromIndex, toIndex).
  You cache the Dictionary/Map, not the “result-data”.
  (Your backend system would have to support a “where exists” clause)

MyPagingIndexValue	EmployeeSurrogateKey
1	5cc02721-c2a0-4fe4-ab8c-83828c667ed8
2	64a0eac1-b072-467f-9332-3ff771bf1763
3	a83ebf9b-d339-4583-8850-a598bcfdad2d
4	6bd66b91-6cc4-402d-b683-417f30b8126b
5	80bb86f7-c1e2-49cc-b008-e0e4adf8380f
6	98b71f7e-e3c7-4dbb-a82b-258b968f88de
7	3928ead8-21b7-4473-851f-c6e332e5ac65
8	fca70a6b-7402-43e5-82b3-32b7f58c78ce
9	5374a6e8-5ef3-478d-af2d-b7ff8473cf6a
10	3290f9b3-1a4d-4695-842a-69c616456a38
11	15f8414f-b54f-4608-87c6-d1731ef8b829
12	2a307fe0-0cc6-4831-877f-20af88b5034d
13	be7974c2-2e3b-4ce2-ab29-d5ca05488b4a
14	36878bef-d111-456b-bc52-9801db2c5d5d
15	287e0a37-e91b-4ed6-8266-0d628f05da6f
16	3eb228b1-d78f-4a55-a4dc-84974efd085e
17	73efa4c8-3511-4326-9fe3-ee8c52ebf13a
18	0cf96db8-6e78-48b8-9d1e-6f86cf57105d
19	0787a1be-f61f-4687-a7f3-31fd3d8644dd
20	c496e0b1-947c-4772-9183-0b390f1e4b76
21	904c1086-e15c-4149-b1e3-028fd0cfd924
22	8b415de7-dbfb-4cdc-8761-10924feb3635
23	02ded0a0-7c59-4471-bbb7-421dc72046df
24	7e63c80c-cae2-47c5-ac0f-873454a673fb
25	09c3016d-d767-4f72-8056-c6cfc3f7a5ae
26	5a1af170-d024-449f-aac6-ac04e7734643
27	482086d6-fd8c-4f52-800d-8211cd1b9be1
28	3804e3cd-fa3a-4d2b-a685-24e4ad19c69f
29	5d444057-1cae-4fd9-8a59-64bec55e79f3
30	dee7dce3-798c-435c-8d16-d9462f73daf1
31	895714e7-acae-4058-b180-160629a73ffa
32	78eaf648-d656-4e14-b158-4b44ddf8cecc
33	5ad481f2-737e-4c7f-9b7a-6d0bd50e2454

Time for Demo? (Time allowing)

Run Cancel Disconnect Change Connection HL7FhirDevDaysPagi... Explain

```
1  SELECT [FhirBundleProviderWrapperKey] as 'PrimKey'
2        ,[UniqueIdentifier] as 'AkaTheGetPagesIdThing'
3        ,[BundleProviderPayload] as 'SerializedIBundleProvider'
4        ,[FhirResourceTypeName]
5        ,[IocTypeName] as 'IocTypeNameAkaTheBeanName'
6        ,[CreateDateOffSet]
7        ,[UpdateDateOffSet]
8    FROM [HL7FhirDevDaysPagingDB].[dbo].[FhirBundleProviderWrapper]
```

Results Messages

	PrimKey	AkaTheGetPagesIdThing	SerializedIBundleProvider	FhirResourceTypeName	IocTypeNameAkaTheBeanName
1	1	cc1a8882-3645-4e34-af65-f5af202f129c	0xACED00057372006F636F6D2…	org.hl7.fhir.r4.model.Patient	com.exampleimplementation.fhir.server.myfhirserverimplementation.businesslayer.managers.PatientManager

Contact

- During DevDays, you can find / reach me here:
  - Email: sloan.h.holliday@optum.com

Q&A

ORGANIZED BY

PARTNER