Introducing **DFws**

API for **DFdiscover Services**

Samer Chaudhary

DF/Net Software ULC
Current DataFax Server Architecture

- **DFedcservice** (Open HTTPS API with limitations)
  - Single channel used for all requests
    - Client tools required to maintain connection for each session
  - Data interchange in proprietary format
    - Client tools aware of business logic and database structure
Current DataFax Server Architecture

- Works well for existing DataFax client tools
Current DataFax Server Architecture

• Not ideal for Web/Mobile apps or 3rd party apps

• Difficult to maintain such connections
• Requires detailed understanding of DF backend
Solution?

• A web API that is easy to use
  • Maintain sessions across multiple requests
  • Data interchange in a standard format
  • Accessible by HTTP clients like web browsers & mobile apps
  • Simple requests that do not require backend knowledge
What is a web API?

• Provide interface to access resources from a closed system

• DFdiscover Resources:
  • **Studies** (List of Studies registered on the server)
  • **Records** (One or more data records for a subject or site)
  • **Keys** (List of all primary keys that exist for a given site)
  • **Users** (List of registered users)
Why?

• Popular demand from existing clients

• Industry standard (Google, Amazon, eBay, Yahoo, etc.)

• Available web APIs grown consistently over the past decade

• Widely used with Big Data, Cloud Computing, Cloud Apps, etc.
Why in DFdiscover?

• Support development of Mobile apps

• Expose DFdiscover resources to developers for:
  • integration with other systems
  • develop client’s own apps

• Overcome network disruptions
Solution: DFws

- Features:
  - A web API to expose DFdiscover services
  - RESTful web API that uses full features of HTTP
  - Maintain sessions to DFdiscover backend
  - Data interchange in JSON (standard for mobile/web apps)
Request: https://api.dfdiscover.com/v1/users/datafax

Response: {
  "affiliation": "DF/Net Software ULC",
  "city": "Dundas",
  "country": "Canada",
  "email": "datafax@dfnetresearch.com",
  "fullName": "DataFax admin",
  "phone": "+1 (905) 522-3282",
  "postalCode": "L9H 3H8",
  "routerAccess": true,
  "state": "Ontario",
  "streetAddress": "55 Head St",
  "userLogin": "datafax"
}
DFws - Architecture

- Lightweight architecture for devices with memory constraints
- End-to-end SSL - Securely transmits data to backend encrypted
- Accessible to authorized and registered API clients
- JWT Authentication system (tastier version of web cookie)
DFws - Web API endpoints

• Access data using web API endpoints:
  • Web API URI: https://api.dfdiscover.com/v1
  • Resource name: /studies

• Resource request to fetch list of studies:
  https://api.dfdiscover.com/v1/studies
DFws - HTTP methods

- DFws support following operations:
  - GET - Fetch resource
  - PUT/POST - Update/Save resource
  - LOCK - Acquire resource lock
  - UNLOCK - Release resource lock
## DFws - Endpoints

Base URI: [https://api.dfdiscover.com/v1](https://api.dfdiscover.com/v1)

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Resource</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>/studies</td>
<td>List of studies</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/sites</td>
<td>Sites database for study 154</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/setup</td>
<td>Setup for study 154</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/253/lut/meddra</td>
<td>Lookup table named ‘meddra’ from study 253</td>
<td>GET</td>
</tr>
</tbody>
</table>
## DFws - Endpoints

Base URI: [https://api.dfdiscover.com/v1](https://api.dfdiscover.com/v1)

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Resource</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>/studies/154/missing_map</td>
<td>Missing map for study 154</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/qcproblem_map</td>
<td>QC Problem map study 154</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/visit_map</td>
<td>Visit map for study 154</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/setup/plates/5</td>
<td>Setup for plate 5 of study 154</td>
<td>GET</td>
</tr>
</tbody>
</table>
# DFws - Endpoints

**Base URI:** https://api.dfdiscover.com/v1

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Resource</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>/studies/154/ptlist/2</td>
<td>List of subjects from site 2 with records in database</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/reclist/2001</td>
<td>Existing records keys for the subject id 2001</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/keys/2</td>
<td>Existing primary keys for site 2</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/records/1001/1/4</td>
<td>Existing record for Keys: Subject:1001, visit:1, plate: 4</td>
<td>GET</td>
</tr>
<tr>
<td>/studies/154/records/1001/1/4</td>
<td>Update record with Keys: Subject:1001, visit:1, plate: 4</td>
<td>PUT</td>
</tr>
</tbody>
</table>
## DFws - Endpoints

Base URI: [https://api.dfdiscover.com/v1](https://api.dfdiscover.com/v1)

<table>
<thead>
<tr>
<th>Endpoint (for Admin)</th>
<th>Resource</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>/users</td>
<td>All active users</td>
<td>GET</td>
</tr>
<tr>
<td>/users/datafax</td>
<td>User ‘datafax’ record</td>
<td>GET</td>
</tr>
<tr>
<td>/users/datafax/userroles/2</td>
<td>User-role record for ‘datafax’</td>
<td>GET</td>
</tr>
<tr>
<td>/roles</td>
<td>All active roles (filter by study)</td>
<td>GET</td>
</tr>
<tr>
<td>/roles/general/roleperms</td>
<td>Role-perms for role ‘general’</td>
<td>PUT</td>
</tr>
</tbody>
</table>
DFws - Web API server

• Server-side RESTful Web Listener application (Qt based)

• Fully configurable (ports, connections, database, sessions)

• Admin interface for access control and connections log

• Multi-threaded sessions handler

• Interact with multiple API-enabled DFdiscover servers
Architecture

Web Clients --> HTTPS --> DFws API --> HTTPS --> DF Client Tools --> HTTPS --> DFdiscover Server
Licensing - New ‘Features’ property

- **Enable API:** add DFws feature to DFdiscover license

- Feature usage can be controlled and monitored

- Future ready to support additional features
DFws - Security

- Security Features:
  - HTTPS (SSL encrypted)
  - API Token (Authentication to Access API)
  - JWT as API request token
    - URI, Session token, Expiry & Request verification
  - Session Expiration handler
  - Independent threads per user per session
DFws - Usage examples

- Curl-based calls from scripts
- PHP script for web apps or web CMS systems like WordPress
- Cloud applications: AWS, Azure, Google apps
- Desktop applications
Mobile/Web Apps using DFws

• DFcollect V1
  Mobile application for data entry

• DFdiscover Web
  NodeJS based web application for data validation

• DFdiscover Reporting Web Portal
  PHP based web portal for cross-studies reports
DFcollect - iOS/Android app
DFdiscover Web - NodeJS web app
DFdiscover reporting - PHP web portal
Availability?

• DFws API will be released with DFdiscover 1.0
• Additional license feature required to enable API
Thank You

• Questions?