

# 統一データベースの Web API 対応と Excel 関数の開発

大曾根 康裕<sup>1)</sup>, 齋藤 彰一<sup>2)</sup>, 松尾 啓志<sup>1)</sup>

1) 名古屋工業大学 情報基盤センター

2) 名古屋工業大学 サイバーセキュリティセンター

ohzone@nitech.ac.jp

## Web API Support of Unified Database and Development of Excel Function

Yasuhiro Ohzone<sup>1)</sup>, Shoichi Saito<sup>2)</sup>, Hiroshi Matsuo<sup>1)</sup>

1) Information Technology Center, Nagoya Institute of Technology

2) Cybersecurity Center, Nagoya Institute of Technology

### 概要

名古屋工業大学では学内の情報共有と業務効率化を目的に、統一データベース(以後統一DB)<sup>[1]</sup>を2007年度から運用しており、学内全構成員の情報を一元管理している。2009年度からExcelを使って統一DBを検索する「旧Excel関数」が導入されたが、セキュリティ確保の課題(SQLでの直接検索、サブネットでの利用者制限)と、検索速度の課題(1,000件で約25秒)があった。今回Excel関数を更新するにあたり、安全かつ効率的な検索を目指した。まず個人情報漏洩防ぐため、必要最小限の情報を保管する簡易統一DBを作成し、ここから検索するWebAPIを作成した。このWebAPIに問い合わせるExcel関数をAdd-in<sup>[2]</sup>で設計し、さらにExcel関数にシングルサインオンの機能を追加した。Add-inを供給するサーバーにWebAPIを同居させると、Excelがログインに使っているアカウント(Entra ID)がAdd-inを介してWebAPIまで届き、個人認証できる。効率化の面では、新Excel関数では入力をセルのまとまり(A1:A1001等)で受け取る仕組みにより1,000件で10秒ほどまで効率化した。さらに簡易統一DBを検索特化で調整し、さらに高速化できた。具体的には検索用IDにIndexを付けた上で文字コードをutf8からlatin1に変更したところ、1,000件で0.5秒まで高速化した。

### 1. はじめに

大学の情報化を支援するため、名古屋工業大学で多様なシステムが導入されている。統一DBは、学内全構成員の利用者情報とID管理の中核となるデータベースで、2007年度より運用されている。統一DBで学内の主要なシステムに関わる全構成員を管理し、各システムに利用者情報を配信している。さらに、教職員・学生の各種情報を一元管理し、必要とする事務部署や教員・学生に提供している。その結果、学内での情報共有を促進し、部署毎の情報収集を効率化している。学生への情報提供により、学生から事務への問い合わせも抑制し、事務作業の負担軽減となっている(図1)。

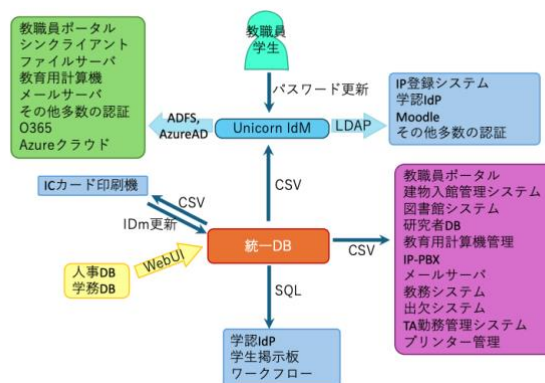


図 1. 統一DBを核とするID情報の連携

### 2. 旧Excel関数

2009年度からはExcelから直接統一DBを検索し、書類作成の手助けをする仕組み(旧Excel関数)の運用を始めた。設計はCOM Add-in<sup>[3]</sup>で、使用者のアカウントIDをデータベースIDに変えて統一DBにログイン

し、SQL で直接検索して結果を得る方式であった。使用対象者は端末のサブネットで制限しており、一部の事務職員のみが使用できた。データベースでのアカウントに閲覧権限を設定することで、閲覧を制御する仕組みであった。万が一の脆弱性に備えて、高度な認証と情報漏洩対策の強化が必要である。また、検索 1 件毎に関数 1 つを要する構造で、1,000 件の検索に 25 秒ほど掛かっていた。

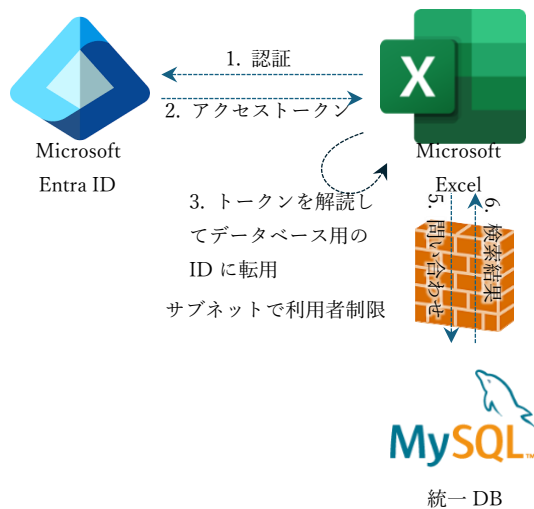


図 2. 旧 Excel 関数の動作図

### 3. 簡易統一 DB と Web API の構築

情報漏洩を抑制するため、必要な情報に絞った簡易統一 DB を MySQL で作成した。この簡易統一 DB は厳密には統一 DB と異なるが、定期的に ETL ツールで同期する事で検索結果が正しくなるよう保全する。この簡易統一 DB を基に情報を提供するサーバーを、Spring Boot で構築した。アプリ同士がデータを率直に連携するための繋ぎ方を API と言い、JSON や XML 規格が標準である。いずれも Spring Boot が扱え、整備が容易である。この API を提供するサーバーを Web API と言い、Web API から Excel に効率よく伝達するのが本改修の目標である。

### 4. Excel による Web API 利用の道筋

Excel で Web API にアクセスする候補として、WebService 関数・Power Query・Add-in が俎上に上がった。ここで必要な機能は Web API で検索できるだけでなく、利用する際に 1 工程程度で結果が得られる事である。

まずは WebService 関数<sup>[4]</sup>だが、これにより Web からデータを読み込むことはできる。しかし、読み込ませたものを利用するには区切り文字や処理方法を指定する必要がある。具体的には検索したい文字を EncodeURL で文字コードを変換して URL に埋め込み、取得した XML データを FilterXML 等で切り出して表示する。これは作業工数が多いので不適切である。

次の Power Query<sup>[5]</sup>も Web からデータを導入できる。Web.Contents でデータを取得し、Json.Document で展開し、結果を一度テーブル等に展開する使い方をしようである。データ連携(ETL)を主に想定していて、先ほどと同様ワンステップで利用できない。

最後に旧 Excel 関数で採用された方法の後継にあたる Add-in だが、これは Excel に導入した簡易アプリで機能を拡張する手法である。検索や展開のための行程は全て Add-in 側が引き受けられることから、本構築の基盤として選定した。

### 5. Add-in 開発の方針

Office 2013 以後は HTML, CSS, TypeScript や JavaScript といった一般的な Web 技術を活用した開発方式が用いられている。実体は Edge ブラウザで動き、Office JavaScript API<sup>[6]</sup>を介して Excel を操作する Web アプリである。Web API との連携は fetch や XMLHttpRequest<sup>[7]</sup>で、JSON を介して容易に実現できる。さらに Add-in では、カスタム関数<sup>[8]</sup>を使うと通常関数のように検索できる。もう一つの柱の認証に関して

SAML ライブラリを利用したシングルサインオン(SSO)の機能<sup>[9]</sup>が提供されている。SSOとは、一度のログインで複数のサービスを利用できる仕組みである。SSO とカスタム関数を組み合わせる方法<sup>[10]</sup>も提供されている。

## 6. Add-in の試作

Add-in プロジェクトの作成に最新版の Yeoman ジェネレーター<sup>[11]</sup>を用いた。この Yeoman ジェネレーターと、生成したプロジェクトの運用に Node.js を要する事から、開発は Node.js v18 を導入した Windows 11 で行った。

ここで必要な機能は、カスタム関数から Web API に繋ぐ構造と、使用者を制御するための認証である。Yeoman ジェネレーターで提示される機能では、カスタム関数とシングルサインオンを併せて選択できなかったため、まずはカスタム関数の Add-in として作成した。

## 7. カスタム関数の作成<sup>[12]</sup>

Add-in の基本規格である manifest.xml にカスタム関数用の functions.js を追記する必要があるが、Yeoman ジェネレーターで導入済みである。TypeScript または JavaScript で記述した関数のコードコメントに

@customfunction を書き入れると、

CustomFunctionsMetadataPlugin<sup>[13]</sup>がビルド時 function.json に翻訳し、カスタム関数が登録される。カスタム関数では一般関数とは異なり、名前空間が必須となる。本件での名前空間は、利用者の入力の手間が少なくなるよう「DB」とした。プロジェクトの開発・作成に関わる設定は package.json に記述してあり、“npm run build:dev”コマンドで Excel が Add-in と認識できるファイル群に変換される(図 3)。この状態で“npm run start:desktop”コマンドを実行すると手元の端末に試験サーバ

(https://localhost:2345)が起動し、Excel に Add-in が読み込まれる。

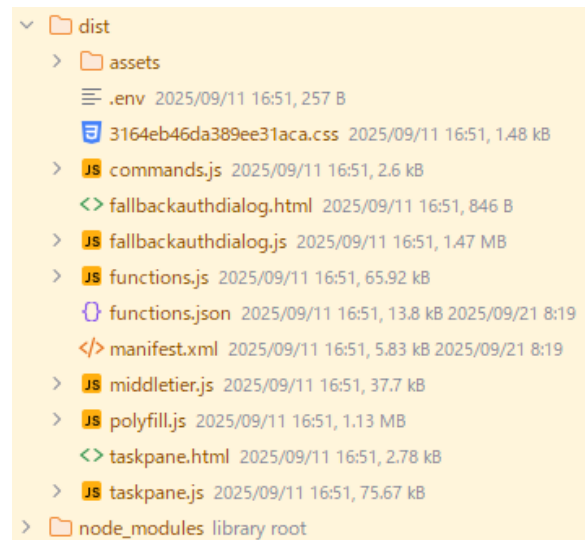


図 3. 作成した Add-in 用ファイル群

## 8. Web API と Add-in を同居

Web API を起動すると、localhost:8765 で問い合わせが可能になる。ポートが異なるため、セキュリティ的には別サーバーとみなされる。5 章で述べたように Add-in の実態は Edge ブラウザである。別サーバーへの問い合わせとして CORS 制約<sup>[14]</sup>で遮断される。遮断を防ぐため Web API を localhost:2345 にて利用する必要がある。これを実現するため、nginx の Proxy を利用した。

## 9. JSON 規格で相互通信

カスタム関数では、Excel のセルを引用すると 2 次元配列として取得できる。この 2 次元配列を整形した後、JSON に変換して Web API に送信する。Web API 側では JSON を配列に復号して簡易統一 DB の検索に利用する。検索は Spring Boot の Repository 機能で簡易統一 DB に対して問い合わせ、結果を再び JSON に変換して返す。カスタム関数は受け取った JSON を配列に復号し、最終的に 2 次元配列として Excel に渡す。Excel はこのデータを元に結果を複数のセルに渡って表示す

る(図 4). 結果に書式を与えたい場合は CellValue の書式で返せば日付や小数点などの表現方法も表現できる.

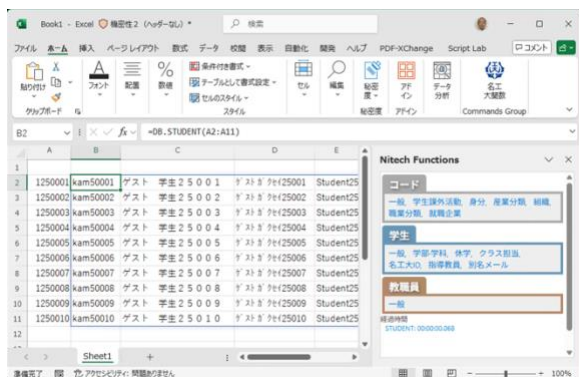


図 4. Excel 関数の使用例

## 10. シングルサインオンの機能を追加

まずはカスタム関数の際と同様, Yeoman ジェネレーターでシングルサインオンの Add-in を作成した. このプロジェクトと, 公式でのシングルサインオン機能の追加法<sup>[9]</sup>を参考に手動で機能を追加した. Azure Portal<sup>[15]</sup>に Add-in を登録する必要がある. 登録はサーバー単位で行い, 開発時は localhost:2345 で登録できる. 登録するとクライアント ID とトークンの鍵が得られるので, これを証明書として Add-in 側にも登録する. この登録によって, Excel を利用する際に使っているアカウント情報をアクセストークンの形で得られるようになる.

このアクセストークンを Bearer トークンの形にして Web API に問い合わせると, この認証情報がさらに伝播する. 8 章の作業で Web API は Add-in と同居している扱いなので, Add-in で登録した localhost:2345 及びクライアント ID とトークンの鍵を Web API にも登録する. これにより, Web API が受け取ったアクセストークンを Microsoft Entra を介して解読できるようになる. 解読で得られたアカウント情報と統一 DB の情報を照合すると, 教職員の身分や所属を判定できる. これ

により Web API へのアクセス可否を判定でき, アクセスログも付けられる(図 5).

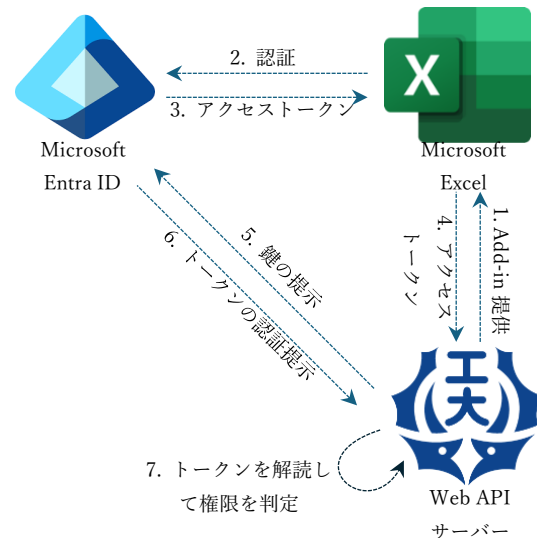


図 5. Excel 関数の認証動線

## 11. 検索の高速化

Web API に設置した簡易統一 DB は検索専用のため, 検索を最適化するための工夫を盛り込みやすい. 効率化前は 1,000 件あたり 10 秒ほども掛かっていたが, 主な検索は ID であるため, ここを効率化した. MySQL はありとあらゆる文字を扱うために UTF8mb4 が標準だが, これは違字体を扱うための余計な機能が付いている. そこで ID だけ latin1 に統一したところ, 1,000 件あたり 0.5 秒ほどまで高速化できた.

## 12. おわりに

Excel の Add-in に Web API を同居させることにより, Excel でログインに利用している Entra ID を Web API でも利用できることを確認した. Add-in ではカスタム関数に加え, タスクペインを組み合わせた入力支援なども行える. 業務効率の向上を支える道具として利用できる. 執筆段階ではまだ情報基盤センター関係者内での試運転の段階だが, 全学教職員にまで手広く公開して利用する予定である.

## 参考文献

- [1] 齋藤彰一, 打矢隆弘, 松井俊浩, 大曾根康裕, 松尾啓志「名工大統一データベース: 学内情報共有・認証基盤データベースの構築と運用」 pp247-251, 2011.
- [2] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/overview/office-add-ins>  
「Office アドインプラットフォームの概要」 2025.9.1.
- [3] <https://support.microsoft.com/ja-jp/topic/91f5ff06-0c9c-b98e-06e9-3657964eec72>  
「Excel COM アドインと Automation アドイン」 2025.9.1.
- [4] <https://support.microsoft.com/ja-jp/office/0546a35a-ecc6-4739-aed7-c0b7ce1562c4>  
「WEBSERVICE 関数」 2025.9.1.
- [5] <https://support.microsoft.com/ja-jp/office/ca69e0f0-3db1-4493-900c-6279bef08df4>  
「Excel クエリを作成、読み込み、編集する (Power Query)」 2025.9.1.
- [6] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/reference/javascript-api-for-office>  
「Office JavaScript API リファレンスドキュメント」 2025.9.1.
- [7] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/excel/custom-functions-web-reqs>  
「カスタム関数でデータを受信して処理する」 2025.9.1.
- [8] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/tutorials/excel-tutorial-create-custom-functions>  
「チュートリアル: Excel でのカスタム関数の作成」 2025.9.1.
- [9] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/develop/sso-in-office-add-ins>  
「Office アドインでシングルサインオン (SSO) を有効にする」 2025.9.1.
- [10] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/excel/custom-functions-authentication>  
「共有ランタイムのないカスタム関数の認証」 2025.9.1.
- [11] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/develop/yeoman-generator-overview>  
「Yeoman ジェネレーターを使用して Office アドインプロジェクトを作成する」 2025.9.1.
- [12] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/excel/custom-functions-overview>  
「Excel でカスタム関数を作成する」 2025.9.1.
- [13] <https://learn.microsoft.com/ja-jp/office/dev/add-ins/excel/custom-functions-json-autogeneration>  
「カスタム関数用の JSON メタデータの自動生成」 2025.9.1.
- [14] <https://developer.mozilla.org/ja/docs/Web/HTTP/Guides/CORS>  
「オリジン間リソース共有 (CORS)」 2025.9.1.
- [15] <https://azure.microsoft.com/ja-jp/get-started/azure-portal>  
「Microsoft Azure portal」 2025.9.1.