

Automating API Development to Deliver Faster Business Results

Database API Generator Increases Productivity to Drive Faster Time-to-Market Deliverables

TABLE OF CONTENTS

[Introducing the Database API Generator](#)

[Why Create APIs and What Problems Do They Solve?](#)

[Embrace Open and Deliver at the Speed of Business](#)

[What is Behind the Curtain?](#)

[Actionable Insights](#)

[Uncertain on Where to Begin?](#)

[Leveraging Investments for Future Growth](#)

Introducing the Database API Generator

The Database Application Programming Interface Generator simplifies the development of new web-based applications by providing direct access to existing mainframe data. It automates the creation of Representational State Transfer (REST) APIs that access data. Developers can easily create APIs that can be called from the web applications, providing easy access to database resources using modern, standard techniques.

This document outlines the importance of APIs, shares insights on how the Database API Generator can be used, and provides real-world examples and benefits.

Why Create APIs and What Problems Do They Solve?

Data is a business's most valuable asset and mainframe databases house this data, fueling mission-critical systems. Organizations must have the ability to leverage their existing investments, which includes skills to build and maintain new and existing mission-critical applications to compete and succeed in their respective industries. They need to digitize both internal and external customer interactions in a more modern way.

Organizations leveraging the mainframe to run their business continue to build new applications to meet ongoing business demands. Some organizations might be in a *status quo* or *business as usual* mode, with no new application growth. However, even if no net new development efforts are part of your mainframe strategy, you still need to maintain existing applications.

Opening up and making mainframe databases more extensible by using APIs and services is key to simplifying and automating everyday user experiences. APIs provide modern, scalable access to z/OS services, tools, and data.

Most APIs today are based on the principles of REST, which uses HTTP protocols to invoke the API, and JSON documents to represent the data exchanged with the service. These requests are text based, allowing easy interchange between different platforms.

BY PROVIDING TOOLS THAT HELP DEVELOPERS USE THEIR EXISTING SKILLS, ORGANIZATIONS CAN BETTER LEVERAGE THE MAINFRAME TO RUN THEIR BUSINESSES AND UTILIZE EXISTING BUSINESS ASSETS MORE QUICKLY AND EFFECTIVELY.

There are two types of APIs:

- *Business APIs* are for business applications and data, which is the focus of this document.
- *System APIs* focus on systems management, performance, and infrastructure. For information about systems APIs, see the [IDMS™ Systems API white paper](#) and the [Datacom™ Systems API white paper](#).

The Database API Generator automates the creation of REST APIs that access business data, so that developers can quickly and easily create new or integrate with existing web, mobile, or cloud-based applications.

Next-generation developers are familiar with modern tools, such as Microsoft Visual Studio Code (VS Code) to develop RESTful APIs and browser-based applications, in languages such as Java or JavaScript. Often, these developers have limited knowledge about the mainframe. By providing tools that help developers use their existing skills, organizations can better leverage the mainframe to run their businesses and utilize existing business assets more quickly and effectively.

Embrace Open and Deliver at the Speed of Business

The Broadcom® IDMS and Datacom *Embrace Open* approach aligns with the Broadcom *Open Mainframe* strategy. By leveraging open-source technologies and taking an API approach, it further enables customers to integrate the mainframe with everything (data, apps, and services) from mobile to mainframe. This approach empowers end users to increase productivity, using tools they are familiar with, shielding them from the perceived complexity of the mainframe.

With the inception of [Zowe](#), an open-source framework for mainframe, the platform enables customers to securely develop and manage applications with an experience that is similar to distributed and cloud environments. The framework and capabilities accelerate mainframe application development through modern DevOps tooling while introducing new levels of automation and infrastructure and systems management. By arming new and emerging mainframers with these valuable skills, Zowe helps to sustain the mainframe environment by attracting and retaining next-generation talent.

What is Behind the Curtain?

Modern tooling is the name of the game and a key component of simplifying and automating the user experience. In this case, the Database API Generator leverages VS Code, which is a source-code editor made by Microsoft. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and integration with [Git](#).

By automating the creation of REST APIs that access data, developers can easily create web applications to discover and consume the data using modern, standard techniques.

THE VS CODE EXTENSION INVOKES THE APIS THAT THE DATABASE METADATA SERVICE PROVIDES TO DISCOVER THE DATABASE OBJECTS, INCLUDING TABLES, VIEWS, AND PROCEDURES.

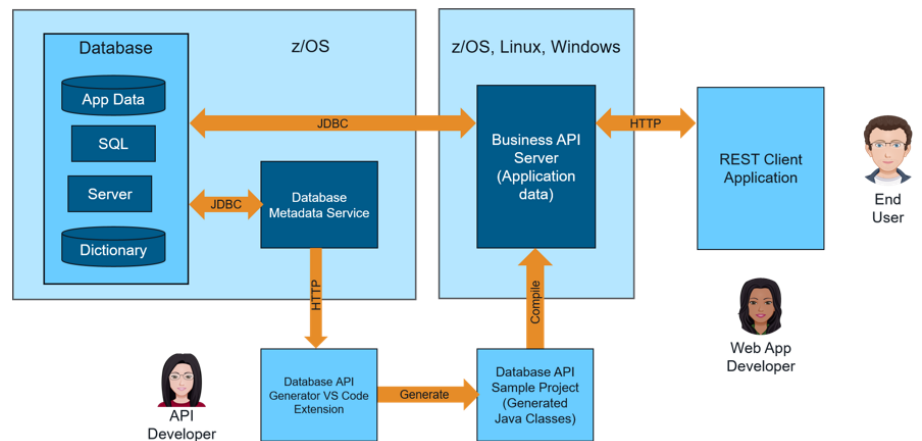
Architecture

The Database API Generator is composed of several components, all of which are leveraged to enable the overall functionality and automation:

- **Database metadata service:** A standalone component that is used by the Database API Generator VS Code extension explorer. The REST API endpoints of this service expose standard JDBC metadata, which allows for the discovery of schemas, tables, relationships, procedures, and other metadata.
- **Database API Generator VS Code extension:** Leverages the VS Code extension that is used from within a Spring Boot project. Using the explorer functionality, developers can discover and select user-defined database resources for generating the Java classes and methods used to create the REST API endpoints to access the data.
- **Database API Generator sample project:** A sample Spring Boot project that provides a framework for creating a business API server. The Database API Generator is used from within the project to discover and generate the components of resulting REST API endpoints that access user-defined data. The project is then compiled into the resulting business API server.
- **Business API server:** Deployed on z/OS or distributed platforms. The REST API endpoints have been generated by the developer in the Database API Generator and access the user-defined data. These endpoints can be used to develop the web applications for the end user.

The following figure depicts the Database API Generator overall architecture flow.

Figure 1: Database API Generator Architecture



The VS Code extension invokes the APIs that the Database Metadata Service provides to discover the database objects, including tables, views, and procedures. The metadata service is an API wrapper on the JDBC database metadata methods. This architecture allows the extension to be very lightweight.

The sample project is included with the download of the Database API Generator. The starter project provides a file structure, common code, build script, and other configuration files—a foundation to get users *started*. After the Java classes are generated to implement the API, the API server can be built immediately with no additional coding other than modification of configuration files with user-specific settings.

THE BUSINESS APIS GENERATED BY THE DATABASE API GENERATOR ARE AUTOMATICALLY COMPATIBLE WITH ZOWE, AND WHEN DEPLOYED ON Z/OS, CAN BE EASILY CONFIGURED AND INTEGRATE INTO THE ZOWE ENVIRONMENT.

Common libraries are included that support common functions, provide integration with Zowe, and satisfy dependencies on open-source components. Systems REST APIs that Broadcom provides for both IDMS and Datacom, including the metadata service, use these same common libraries. If you are familiar with the open-source ecosystem, you are aware that vulnerabilities are often found in open-source software. Broadcom proactively monitors and addresses these vulnerabilities in the common libraries for Broadcom-supported APIs. Database API Generator users can also take advantage of the open-source ecosystem to check for vulnerabilities with the custom APIs that are generated.

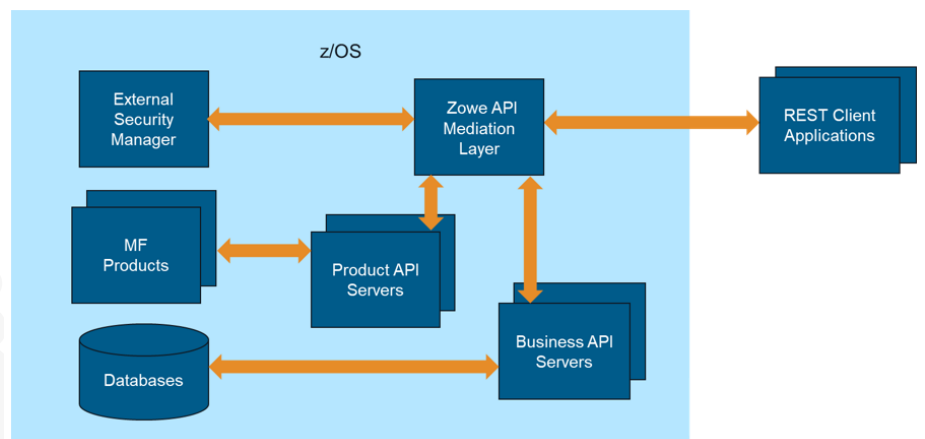
Integration with the Zowe API Mediation Layer

The Zowe open-source framework provides a gateway for APIs deployed on z/OS. This gateway is called the **Zowe API Mediation Layer (API ML)**. The API ML provides several advantages for these APIs, including enhanced security, scalability, and load balancing. The gateway server exposes all of its APIs through a single port, limiting network exposure. The gateway server supports single sign-on integrated with the mainframe external security manager, whether it be Top Secret™, ACF2™, or RACF. The Zowe API ML also provides a catalog that supports API discovery and documentation for developers.

The business APIs generated by the Database API Generator are automatically compatible with Zowe. When the business APIs are deployed on z/OS, they can be easily configured and integrate into the Zowe environment. Our product API servers, such as the Systems REST APIs server and Metadata API server, as well as user-defined business API servers, can all be accessed through the Zowe API ML gateway.

Using Zowe provides several advantages, but it is not required. The APIs that the Database API Generator creates can be deployed without integration to the Zowe API ML on z/OS, and the endpoints are secured by the external security manager (Top Secret, ACF2, or RACF). The APIs can also be deployed on a distributed system server or in the Cloud. Because the generated API uses JDBC to access the database, the database management system performs its normal authentication and authorization checking.

Figure 2: Zowe Integration (Optional)



When deployed off the mainframe, the use of a security product of choice to secure the endpoints themselves is critical.

THE USER INTERFACE IS SUPPLIED USING A WEB APPLICATION RUNNING ON AN APPLICATION SERVER.

Actionable Insights

Use Case 1: Then and Now

In this sample use case, the requirement is to build a new user interface for a distributed web-based application. The interface leverages existing IDMS network data by manually coding the API integration (server) between the front end and the back end. Although this use case is based on IDMS, the same techniques apply to Datacom databases.

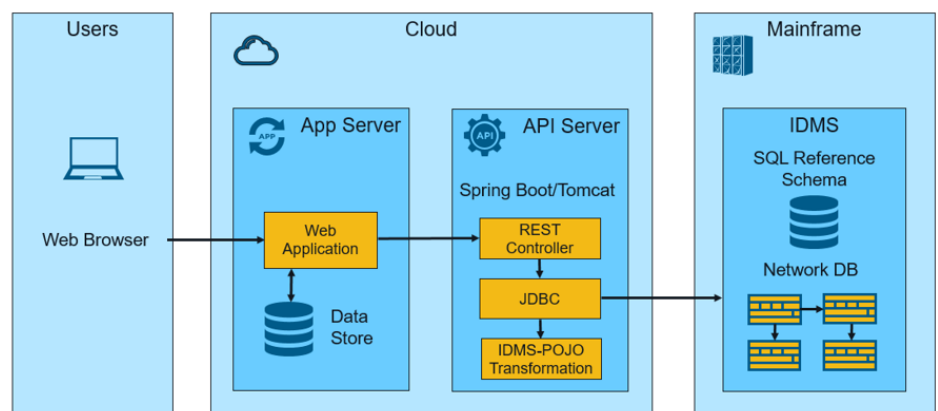
The end users of this application are presented with a user interface in a browser. The user interface is supplied using a web application running on an application server. The application server is implemented as an on-premise, distributed server that can, in principle, run in the cloud.

The web application accesses the database through an API that hides the details about IDMS and the database structure. The API is provided by a server running on a distributed system, with the potential to run in the cloud as well.

An IDMS SQL schema is defined to enable access to the IDMS network database. This special type of SQL schema references the network database definition (for example, the network schema) to expose the existing network records directly as SQL tables. This schema does not require any application or database structural changes.

The API server is implemented using the popular open-source Spring Boot REST API framework with an embedded Apache Tomcat server. It uses JDBC to execute SQL statements to access the records in the IDMS network database. The API server converts the IDMS records to plain old Java objects (POJOs), and the Spring Boot framework returns these POJOs to the API caller as JSON documents.

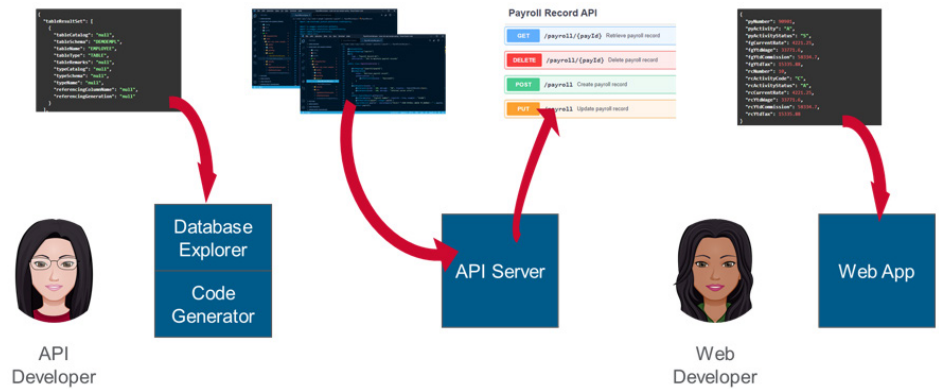
Figure 3: API Server



WE DISTINGUISH THE DEVELOPERS USING TWO PERSONAS: AN API DEVELOPER AND A WEB APPLICATION DEVELOPER.

To provide context in this use case, we distinguish the developers using two personas: an API developer and a web application developer. In some cases, these personas might be the same person with combined skills. Typically, an API developer has Java skills with experience developing REST APIs. This persona is familiar with modern coding editors such as VS Code. They might also have SQL database knowledge, although not necessarily with IDMS or Datacom.

Figure 4: API Developer and Web Application Developer Personas



API Development Scenario before Database API Generator

In the *before* scenario, developers were required to understand the existing IDMS or Datacom object structures. They needed to understand how to manually create SQL queries and then integrate them with the Java code to implement the API server. This manually intensive development effort is tedious and error prone. To modify or enhance the existing APIs, they needed to understand existing API code, which might have been written by a different developer and have code that is difficult to understand or change. Manually written APIs might not follow the standard approach across all objects.

API Development Scenario after Database API Generator

With the available Database Metadata Service and Database API Generator VS Code extension, APIs can be generated with the click of a button. It is a straightforward process to generate an API that exposes database objects as resources that can be retrieved and updated using the API. This process of API development improves the overall user experience and delivers results faster with superior code quality. The generated API can be enhanced and modified easily, providing both short-term and long-term value to any project initiative.

The API developer would use the VS Code extension provided as part of the Database API Generator to create a REST API to access the database. The extension includes a Database Explorer that uses database metadata to display the database structure in a common explorer style *tree view*. The API developer uses the explorer to select the database objects to be included in the API. The VS Code extension generates the Java code to implement the REST API. The generated API server provides [Swagger documentation](#) defining the API endpoints and JSON models that define the data. Swagger documentation helps developers with designing, building, documenting, and consuming REST APIs.

LEVERAGING THE DATABASE API GENERATOR, BROADCOM ENGINEERING TEAMS ARE REALIZING QUICKER TIME TO MARKET DEVELOPING APIS AS PART OF OUR INNOVATION STRATEGIES.

The web developer is a front-end developer with experience in creating browser-based applications and user interfaces. Leveraging the Database API Generator, this developer requires little to no knowledge of the mainframe, mainframe databases, or even SQL because their view of the data is displayed in the Swagger API documentation. The web developer uses whatever framework, language, and development environment they prefer—providing flexibility of choice. The only requirement is the ability to invoke a REST API.

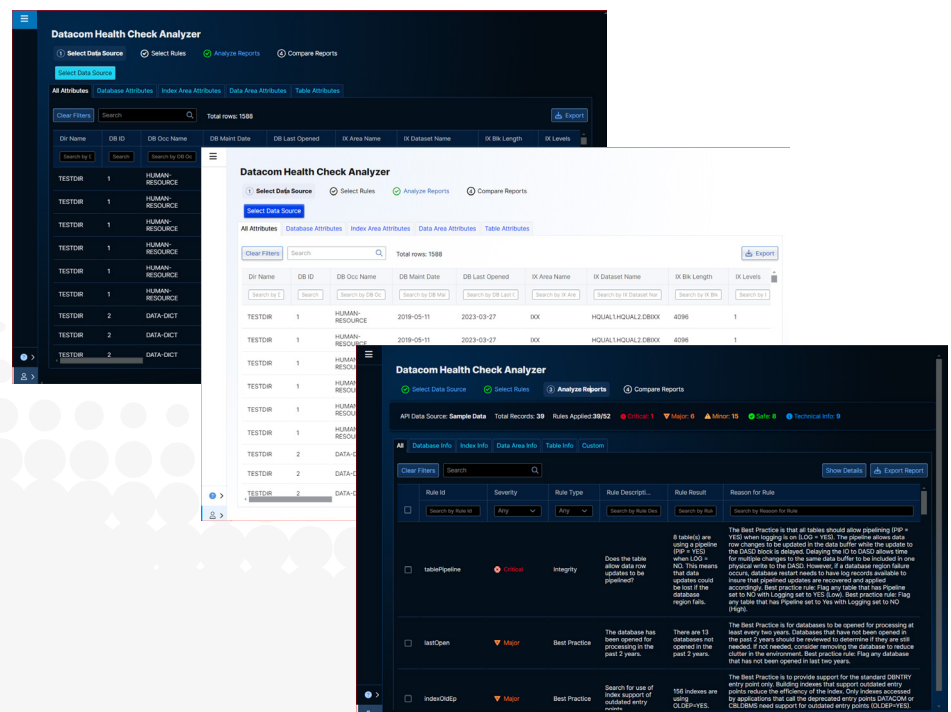
Use Case 2: Drinking Our Own Champagne

Leveraging the Database API Generator, Broadcom engineering teams are realizing quicker time to market developing APIs as part of our innovation strategies. One example is the new Datacom Health Check Analyzer application. Our teams used the Database API Generator to develop the Datacom Systems REST API to integrate with a modern front-end web application. The Datacom Systems REST API was designed for a database administrator to gain valuable insights, best practices, and recommendations within their environment with the click of a button.

The development and delivery of this new application was accelerated by 80% compared to a manual coding effort, which can take significantly longer with the risk of human coding errors. The key aspect was the ability to regenerate the REST API when new data requirements were found, giving the ability to quickly refactor without any development efforts. Not only does this capability allow our teams to deliver superior product value to our customer sooner, but it also frees up engineers to focus on additional product investments and innovations in the development pipeline.

Broadcom is reaping the productivity benefits of our own tooling to deliver new product capabilities to our customers sooner. With faster time-to-market, our customers can realize the same value using the Database API Generator by delivering results to meet project goals and objectives sooner.

Figure 5: Faster Development and Delivery



WE INVITE OUR IDMS AND DATACOM LICENSED CUSTOMERS TO ENGAGE IN AN INTERACTIVE EMBRACE OPEN WORKSHOP.

Uncertain on Where to Begin?

Broadcom is here to support you on your modernization journey. Whether you are using the Database API Generator or any of the database *Embrace Open* product offerings, we can get you kick-started on your projects.

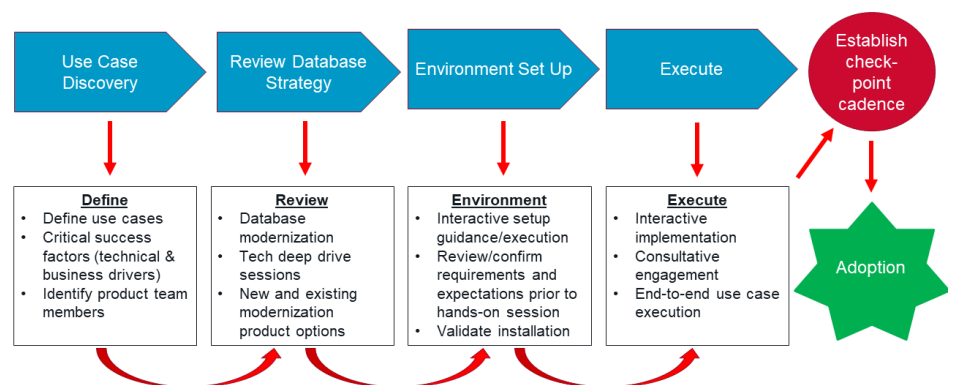
To ensure the user experience is positive, we invite our IDMS and Datacom licensed customers to engage in an interactive Embrace Open Workshop, at no additional cost. The workshop is designed to get you started faster and benefit sooner from new product innovations being delivered in the databases. The workshops use a phased approach and are customized to meet each customer’s unique project needs. The ultimate objective is to make customers comfortable with the new product features or functionality so that they can start using the new innovations upon completion of the workshop.

Figure 6: Value that Goes Beyond Code

Embrace Open Workshops

- Hands on workshops to assist with onboarding of modernization technologies
- Help with adoption and learning
- Facilitated by Broadcom (Technical Services and Product teams)
- Install/deploy in customer’s environment

High-level Summary of Phases



THE END GOAL IS TO ENABLE BROADCOM CUSTOMERS TO LEVERAGE THEIR EXISTING INVESTMENT AND TRANSFORM USER EXPERIENCES OPENING THE DATABASES THROUGH APIS.

Leveraging Investments for Future Growth

APIs build upon the existing value already invested in mission-critical systems that have grown and supported their businesses for decades. IDMS and Datacom are proven, reliable, scalable, and highly performant database management systems. The end goal is to enable Broadcom customers to leverage their existing investment and transform user experiences opening the databases through APIs.

How to Get Started

If you are licensed for IDMS/DB (19.0), Datacom/DB (15.1), or Datacom/AD (15.1), the Database API Generator is available to you right now as part of the base license.

For Database API Generator installation and use, refer to the following information:

- Database Metadata Service ([Installing the Database Metadata Service](#) and [Using the Database Metadata Service](#))
- Database API Generator ([Installing the Database API Generator](#))
- Database Sample API Project ([Configure the Database API Sample Project](#))
- Database Business API Server ([Install and Configure the Business API Server](#))

If you have any questions or require assistance, contact your local technical support group at www.broadcom.com/support/software/contact or visit the [Support Portal](#).

Please visit the [IDMS IUA EIUA Community](#) and [Datacom CADRE Community](#) sites and participate in conversations.