



# CA Database: Business API Strategy

DAVID ROSS – SOFTWARE ARCHITECT  
SHEILA MILLER – PRODUCT MANAGER

# Agenda

- System vs business APIs
- Review API strategy
- Creating business APIs
- Feedback
- General Announcements

# Disclaimer

Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of September 2020 and **is subject to change or withdrawal by CA at any time without notice. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion.**

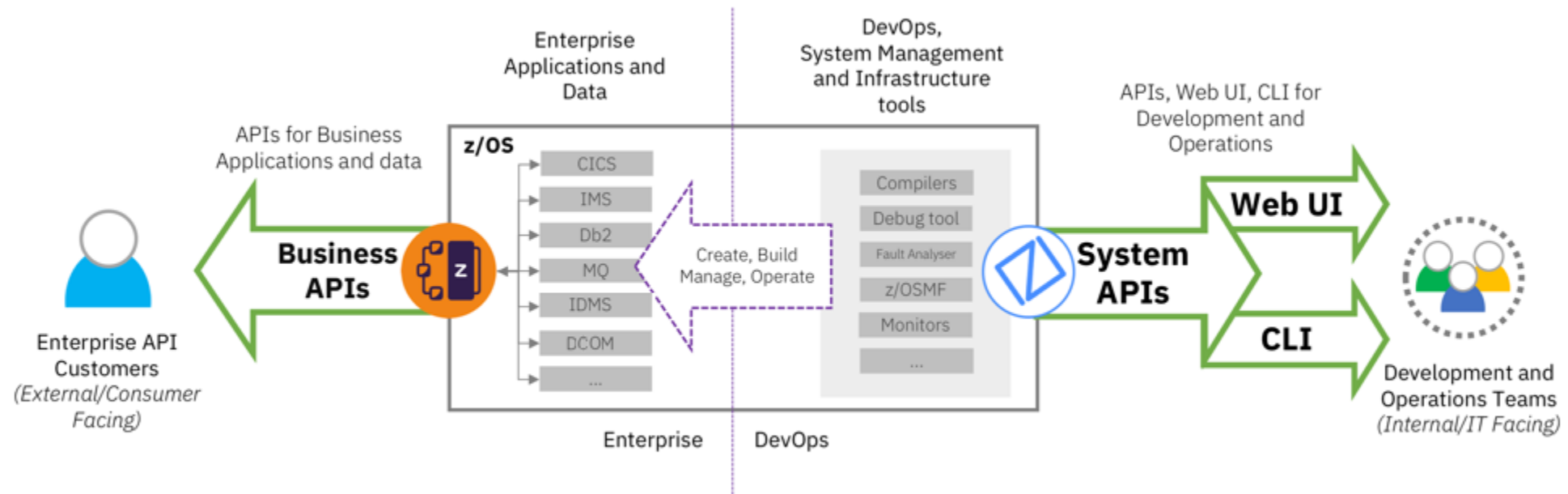
Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

Copyright © 2021 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

**THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY.** CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. **In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages.**

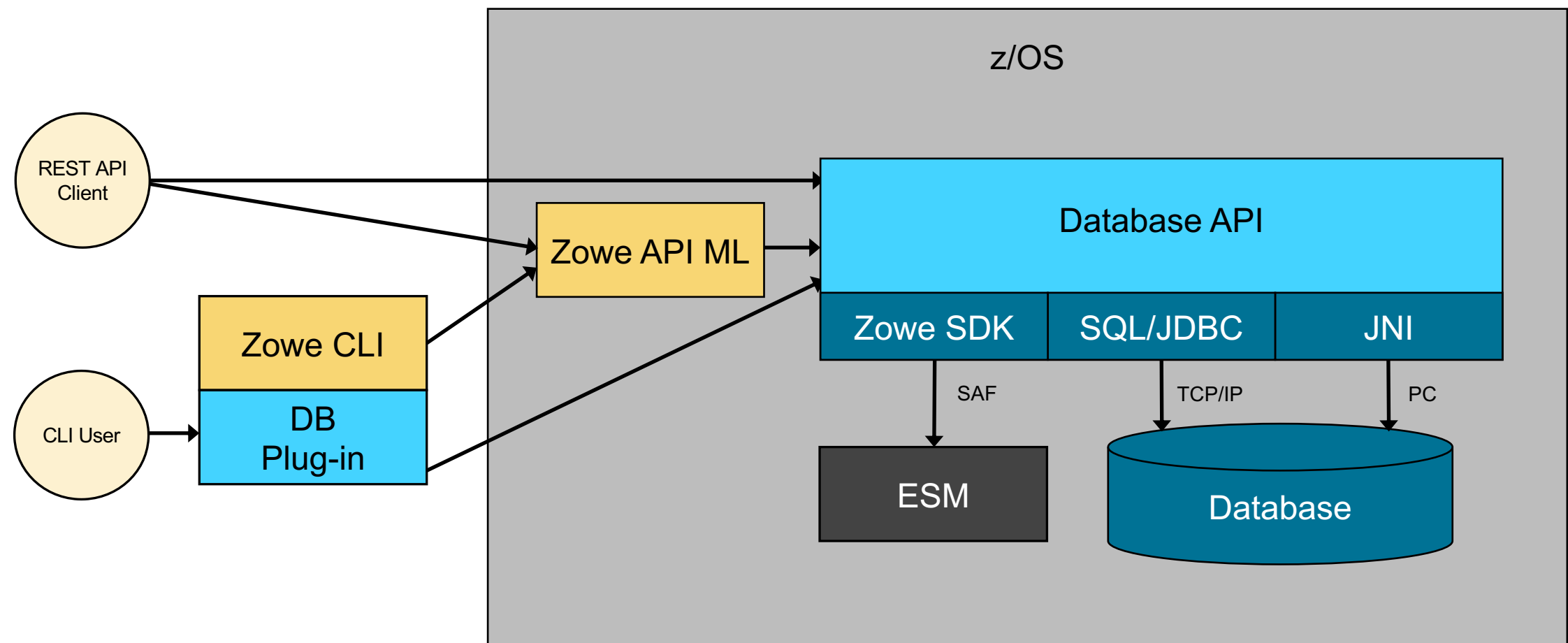
# Why APIs?

- Modern, scalable access to z/OS services, tools and data
- Foundation for modern DevOps workflows and user experiences





# Database API Architecture



# Database API Modernization Strategy

## User Personas

---

- Experienced DBA
- Next generation DBA
- Application DBA
- Systems DBA
- Experienced developer
- Next generation developer
- API developer
- Web app developer

## Use Cases

---

- Automate and simplify database administration
- Optimize operations with capacity planning and performance tuning
- Maintain existing applications with scripting and modern DevOps tools
- Leverage existing applications in new service-based applications

## Solutions

---

- Provide APIs for DBAs
  - Scripting and UI Integration
- Provide API data for analytics
  - Performance monitoring metrics
  - Enable user developed interfaces and dashboards
- Provide APIs for developers
  - DevOps scripting
  - IDE Integration
- Provide tools to create business APIs

# Roger

## Traditional DBA\*

**I LIKE** maintaining procedural administration of the data - availability, consistency, and data integrity are the name of the game!

**I LACK** time to spend on higher level work, because way too much of my day is spent on manual tasks.



Click avatar to download



### GOALS

- Ensure database integrity and availability
- Create and maintain physical database entities, including instances, databases, and table and index spaces
- Design, implement, and monitor regular database housekeeping
- Ensure recoverability of databases
- Monitor and evaluate performance tests and ongoing production performance
- Grant authorizations to database objects
- Guide, review, and approve the design of new databases
- NOT responsible for the actual content of databases

### SKILLS

- **SME for one MF database (IDMS, DB2, IMS or Datacom) and a general understanding of others**
- Knowledgeable in database structures, SQL constructs, troubleshooting, and backup
- Deep understanding of database design principles for performance and availability, security, and the factors affecting those principles.
- General understanding of the application design and process flow
- **General understanding of distributed servers that can connect to MF applications and how to set them up**

### TOOLS

- **Query and database access tools**
- Database ISV tools (administration, performance monitoring, utilities, reporting, etc)
- Strong allegiance to one toolset
- Knowledge in many other tools in z/OS that are not dedicated to database

### PAIN POINTS

- **Having to manage multiple database technologies from different vendors**
- Pressure to reduce wait-times
- “Walk-in” questions that require immediate attention
- Taking on more responsibility with less staff

### WORKS WITH

- Security Admin to grant authorizations
- App Devs for SQL tuning
- QA Tester for access to test data
- Capacity Planner for database-related planning
- System programmers for DB2, z/OS, and hardware platforms
- Storage Admin to control the storage allocated to app databases and create new spaces
- Distributed Server Admins to tune how their servers connect and interact with DB2 in Mainframe
- DevOps Architects to outline new solutions for incoming applications to Mainframe and DB2
- Data Scientists (see last scenario)

### FUTURE SCENARIOS

- **Roger will leverage automation to:**
  - Plan, manage, and initiate reorgs
  - Plan and coordinate recovery
  - Assess and add indices
  - Assess and improve SQL performance
- Roger will define standards and create processes for dev teams that enable “shift-left”
- Roger will Increase technical understanding and usage of core analytics and ML/data science concepts

\*This is a unified DBA persona. Many MF shops have an operations-side DBA and an application-side DBA.

# Ralph

## Systems Performance Engineer

**I LIKE** to maximize system utilization, while maintaining an acceptable level of service.

**I LACK** the time and personnel to accomplish my main tasks while routinely getting calls for Sev 1 alerts.



Click avatar to download



### GOALS

- Track and tune high level resources associated with application performance
- Conduct RCA (triage) of performance problems
- Assess application and systems stability
- Make recommendations on infrastructure enhancements and solutions - main influencer
- Determine KPIs to establish system health

### SKILLS

- Degree in CS (or equivalent)
- Expertise in MF app development or systems engineering
- SME in real-time performance monitoring and tuning in a z/OS Parallel Sysplex Environment
- Knowledgeable in z/OS operating system and subsystems (eg, Batch, CICS, DB2, IMS, IDMS, Datacom and TSO)

### TOOLS

- System tool (eg, consoles, WTOs)
- Monitors (SYSVIEW, Vantage, Netmaster, Omegamon, Mainview)
- Performance analysis (SAS products, MXG)

### PAIN POINTS

- Associating resources to applications
- Understanding KPIs in monitors - each is different
- Limited in-depth view of CICS
- Setting effective thresholds
- Setting effective WLM policy
- Translating KPIs and system level info
- Identification of critical workloads

### WORKS WITH

- Capacity Planner to ensure sufficient capacity to meet service level requirements
- DevOps Architect to assess architecture level modifications for performance or capacity risks
- DBAs and CICS app teams to identify problems and tuning opportunities
- IT, Dev, and BU leads to keep abreast of major systems initiatives

### FUTURE SCENARIO

- Ralph will get freed up to do his main jobs by minimizing the amount SMEs like him need to get engaged as alerts get elevated.

Suggest changes to this persona <https://bit.ly/3it1JLz>

# APIs for DBAs

- Command line APIs
  - Execute DCMT master terminal commands
- Dictionary APIs
  - Maintain database, record, application definitions
  - IDMS IDD, SCHEMA, SSC,SYSGEN compiler APIs
  - Datacom Data Dictionary APIs under consideration
- Performance metrics APIs
  - IDMS Performance Monitor
  - Datacom System Tables
  - SYSVIEW information
  - Datacom SQL performance (with Precision UI)
- Zowe CLI plugins
- Enable use of modern tools

Roger  
DBA



Ralph  
Performance Engineer





# Database Administration APIs

**Compilers** API to execute compiler commands to manage IDMS dictionary components

**POST** `/schema/{datasource}` Executes IDMS SCHEMA commands

**POST** `/idd/{datasource}` Executes IDMS IDD commands

**DC Log** REST API to get CA IDMS DC log messages and information

**GET** `/printLog/{datasource}` Returns a log report

**DCMT** API to execute DCMT commands to manage IDMS system operations

**POST** `/dcmt/quiesce/{datasource}` Executes a DCMT QUIESCE command for one or more target areas

**GET** `/dcmt/{datasource}` Executes a DCMT DISPLAY command

**PUT** `/dcmt/{datasource}` Executes a DCMT VARY command

# Performance Metrics APIs

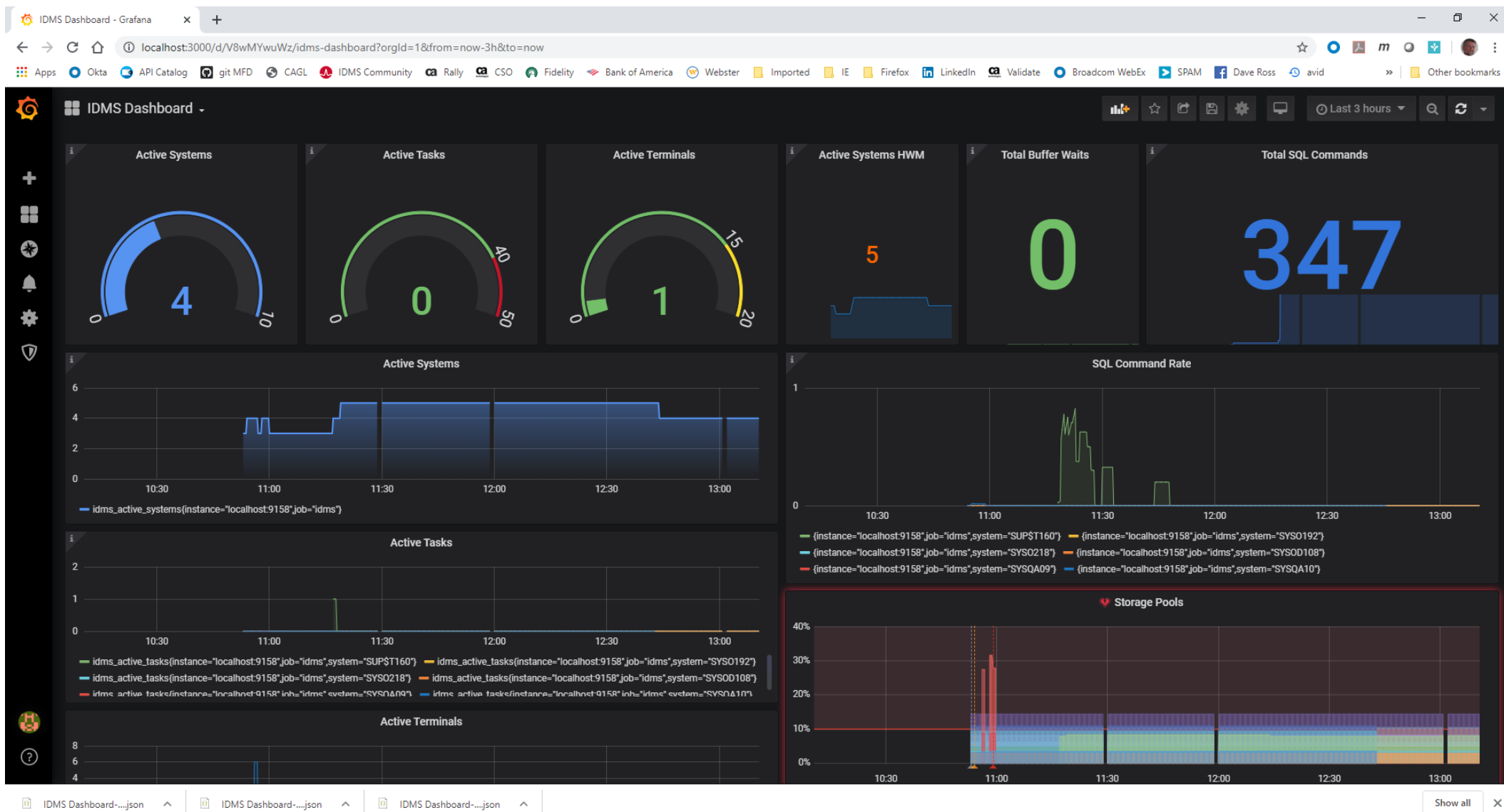
## Performance Monitor REST API to get Realtime Monitor statistics and information

GET	/programPools/{jobname}	Returns a list of program pools
GET	/userTasks/{jobname}	Returns a list of active user tasks
GET	/systems	Returns a list of all active IDMS systems

## The Datacom API Multi User Controller

GET	/mufs/{jobname}/sqltables/DRB	Returns the Directory Database Tables for a specific MultiUserFacility's Directory
GET	/mufs/{jobname}/sqltables/DRA	Returns the Directory Areas Tables for a specific MultiUserFacility's Directory
GET	/svw/ActiveSystems	Returns a list of all Datacom MultiUserFacility instances on an LPAR

# Using the Performance Metrics API



# Rob

## Traditional Mainframe Developer

**I LIKE** seeing all the new people coming onto the mainframe. I want to hand over my baby [the mainframe] to someone who really cares about it.

**I LACK** a clear understanding of where the mainframe is headed. I don't see why we need to change systems that have been so reliable for so long. If it ain't broke, don't fix it!



Click avatar to download



### GOALS

- Deliver quality code faster according to business requirements and code standards
- Scope the code change effort level
- Have complete app-understanding
- Transfer ownership of app change request

### SKILLS

- CS (or related degree) and on-the-job training
- Highly skilled in COBOL, JCL, DB2, VSAM, CICS, CA-7, and ISPF in a z/OS mainframe environment
- Advanced knowledge of mainframe version and revision control practices, data structures, and architecture

### TOOLS

- Endevor, File Master, Sysview, MQ, Intertest, mainframe debugging and incident management tools, and JIRA

### PAIN POINTS

- Bad publicity for COBOL
- [Mainframe modernization efforts](#)

### WORKS WITH

- Business Analyst to understand sizing of work
- Dev Manager to understand business objectives

### FUTURE SCENARIO

- [It seems likely that in the 2020s Rob's role will slowly be replaced by NextGen Devs](#) (both in-house and offshore)
- As Rob mentors the younger generation, he may also be interested in being mentored by them on modern development tools and interfaces.

Suggest changes to this persona <https://bit.ly/3it1JLz>



Mainframe Design Labs

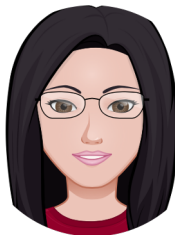


# Michelle

## Next Gen Mainframe Developer

**I LIKE** being a backend person, and the mainframe is the ultimate backend!

**I LACK** many of the tools and coding practices I need to become productive on the mainframe.



### GOALS

- Deliver quality code on-time according to business requirements and code standards
- Scope the code change effort level
- Have sufficient app-understanding
- Help QA folks develop test plan based on app changes

### SKILLS

- **CS degree in modern languages, eg, Java, .NET, C++, and Javascript**
- On-the-job experience in COBOL or PL/I, JCL, and mainframe systems (<1 yr)
- Getting as comfortable with ISPF as with modern IDEs
- Learning how to interact with apps as a mainframe user

### TOOLS

- IDEs, ISPF, MF testing and debugging tools
- Previous experience with open source dev tools (eg, GIT for source control, task runners like Gulp, testing tools like JUnit)

### PAIN POINTS

- Discomfort with MF architecture and operating systems
- Low access to SMEs - need to prioritize questions
- Can't Google MF how-to's
- Hopping between IDEs and ISPF
- Lack full view of the SDLC
- No access to CI/CD automation

### WORKS WITH

- PM to understand sizing of work
- QA to create test plans
- Offshore Devs/Testers
- Dev Manager to understand business objectives

### FUTURE SCENARIO

- **Michelle will use her newly acquired MF skills in combination with her modern coding skills to work on full-stack apps across platforms.**
- **Michelle won't need to switch between IDEs and ISPF, because her dev tools will be integrated into IDEs.**



# APIs for Developers

- Dictionary access APIs
  - IDD compiler APIs
- Code editor for VS Code and Eclipse Che
  - Language sensitive editors
  - Based on LSP (Language Server Protocol)
  - Open source COBOL parser
  - Standard COBOL for Datacom
  - Extensions for IDMS DML
- SQL API
  - Execute SQL statements
  - Get database metadata
- Enable use of modern development tools

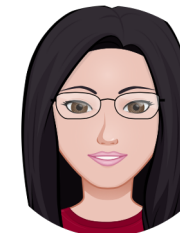
Rob

Traditional Mainframe Developer

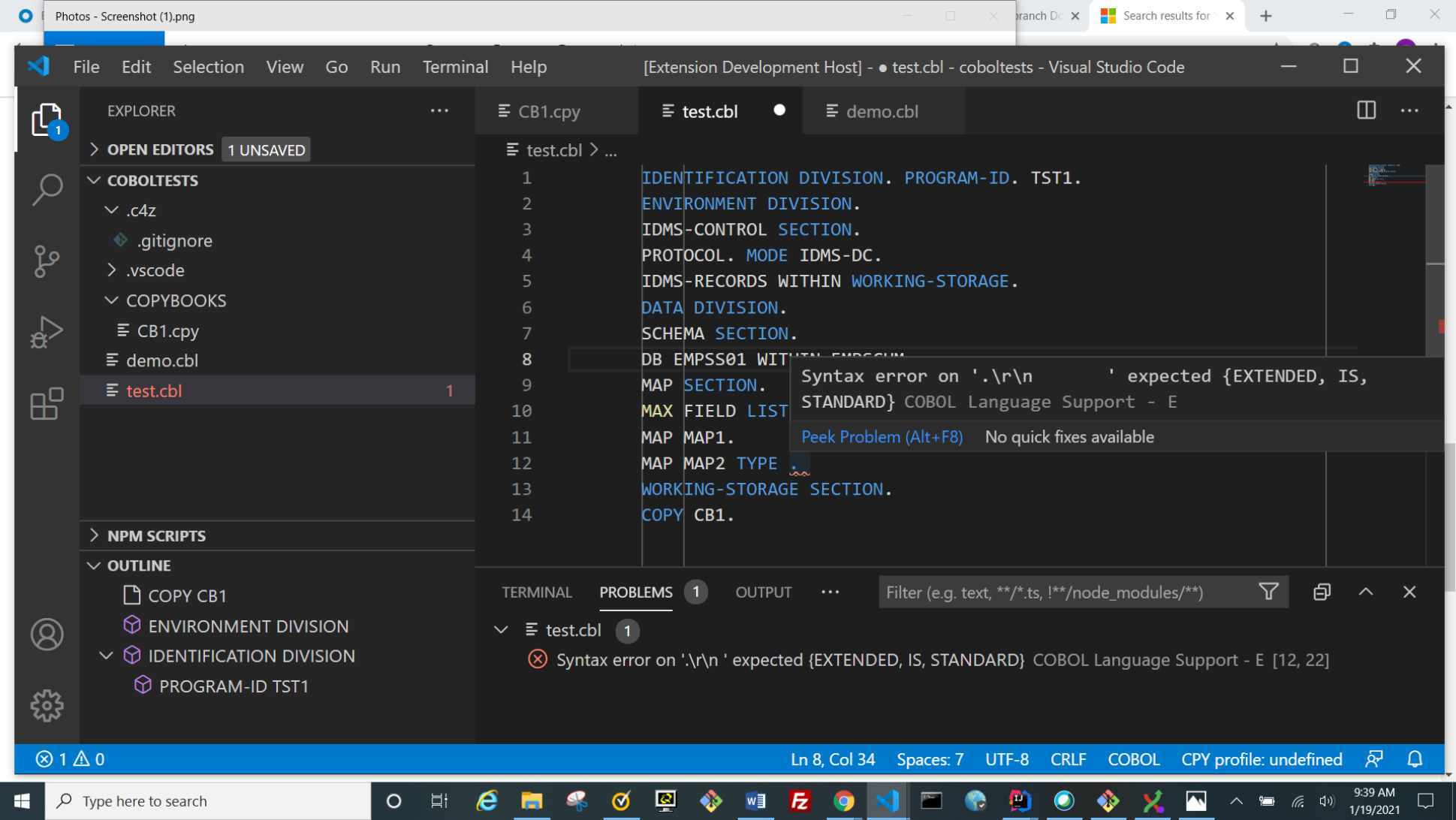


Michelle

Next Gen Mainframe Developer



# Editing COBOL Programs with VS Code



# SQL API for User Data and Business Logic

- Execute SQL statements directly
- Execute a batch of SQL statements from a file
- Endpoints expose standard JDBC metadata
- Discover records, tables, relationships, and procedures

## SQL REST API to execute freeform SQL statements and database metadata functions



POST	/sql/execute/{datasource}	Executes SQL statements	
GET	/sql/schemas/{datasource}	Returns a list of schemas	
GET	/sql/tables/{datasource}	Returns a list of tables	

# SQL Metadata API

## Request URL

```
https://localhost:10080/api/v1/sql/tables/SYSDemo?catalog=null&schemaPattern=DEMOEMPL%25&tablePattern=EMP%25
```

## Server response

### Code

### Details

200

OK

Example Value | Model

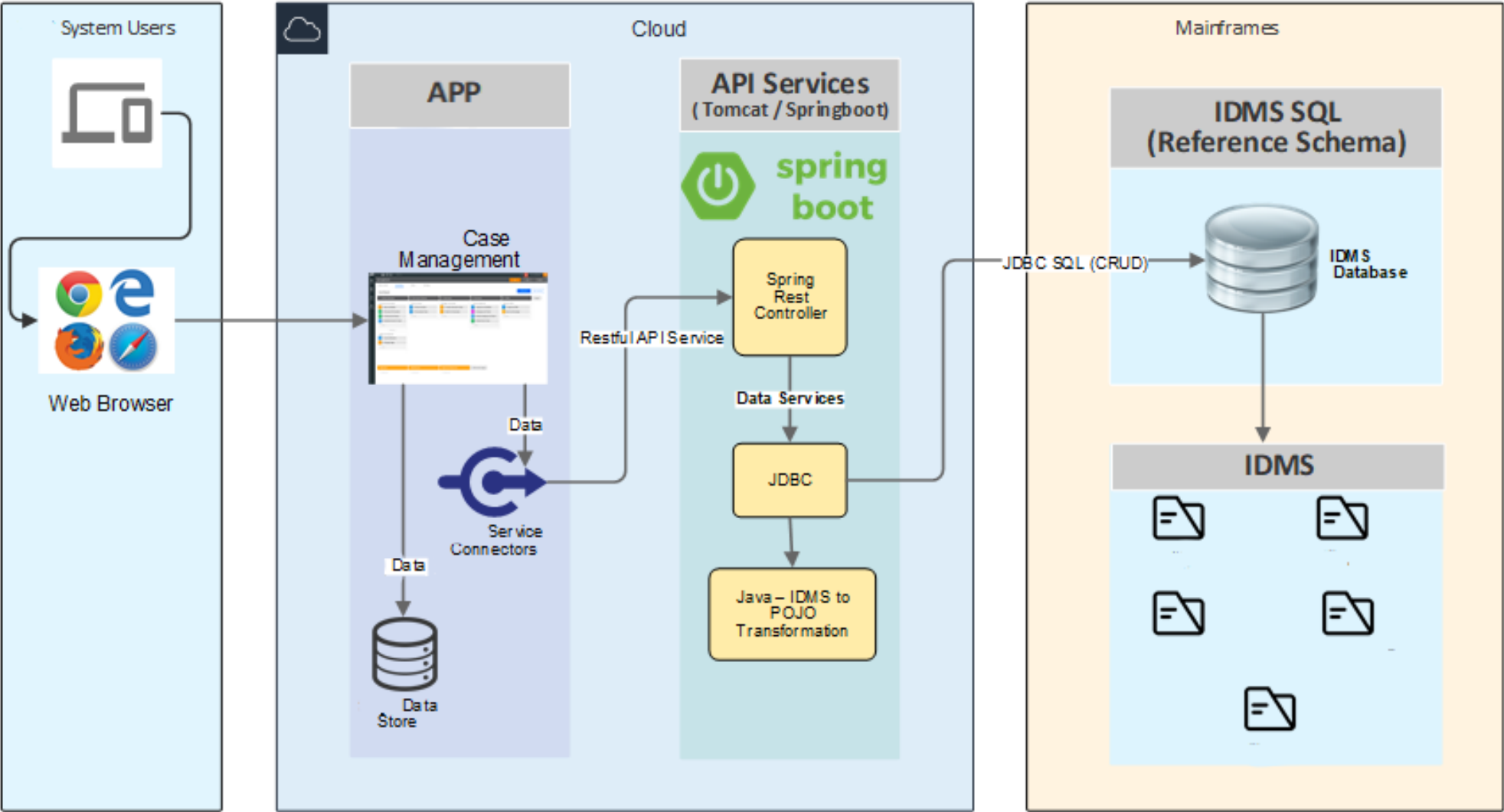
```
{
  "tableResultSet": [
    {
      "tableCatalog": "null",
      "tableSchema": "DEMOEMPL",
      "tableName": "EMPLOYEE",
      "tableType": "TABLE",
      "tableRemarks": "null",
      "typeCatalog": "null",
      "typeSchema": "null",
      "typeName": "null",
      "referencingColumnName": "null",
      "referencingGeneration": "null"
    }
  ],
}
```

# Business API Hill Statements

- Problem
  - Users need better new user interfaces to transaction processing applications
  - Developers need to create new web based applications that maintain the data in existing databases
- Hill statements
  - An IDMS or Datacom developer can easily create REST APIs that access native data and business logic
  - A web developer can discover and consume REST APIs that access these native database



# Case Study: Leveraging a Database for New Applications



# Exposing a Database Record as a REST API

Payroll Record  
COBOL Copy Book

Payroll Record  
JSON Document

```
02 PAY-IDENTIFICATION.  
03 PY-NUMBER          PIC 9(5).  
03 PY-ACTIVITY-CODE   PIC X(1).  
03 PY-ACTIVITY-STATUS PIC X(1).  
02 PAY-FIGURES.  
03 FG-CURRENT-RATE    PIC 9(6)V9(2).  
03 FG-YTD-WAGE        PIC 9(6)V9(2).  
03 FG-YTD-COMMISSION  PIC 9(6)V9(2).  
03 FG-YTD-TAX         PIC 9(6)V9(2).  
02 PAY-RECORD.  
03 RC-NUMBER          PIC 9(5).  
03 RC-ACTIVITY-CODE   PIC X(1).  
03 RC-ACTIVITY-STATUS PIC X(1).  
03 RC-CURRENT-RATE    PIC 9(6)V9(2).  
03 RC-YTD-WAGE        PIC 9(6)V9(2).  
03 RC-YTD-COMMISSION  PIC 9(6)V9(2).  
03 RC-YTD-TAX         PIC 9(6)V9(2).
```

HTTP GET /payroll/90901

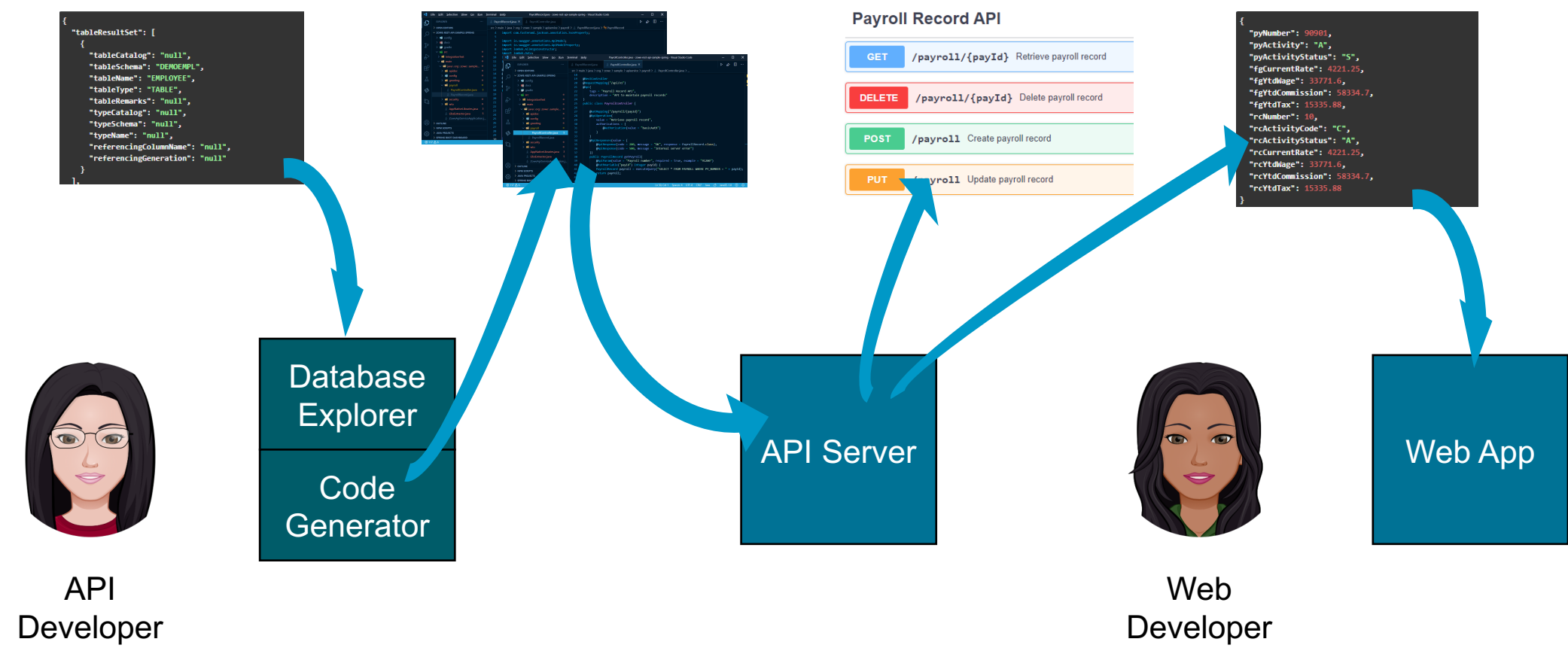
HTTP PUT /payroll

```
{  
  "pyNumber": 90901,  
  "pyActivity": "A",  
  "pyActivityStatus": "S",  
  "fgCurrentRate": 4221.45,  
  "fgYtdWage": 33771.60,  
  "fgYtdCommission": 58334.70,  
  "fgYtdTax": 15355.88,  
  "rcNumber": 10,  
  "rcActivityCode": "C",  
  "rcActivityStatus": "A",  
  "rcCurrentRate": 4221.45,  
  "rcYtdWage": 33771.60,  
  "rcYtdCommission": 58334.70,  
  "rcYtdTax": 15355.88  
}
```

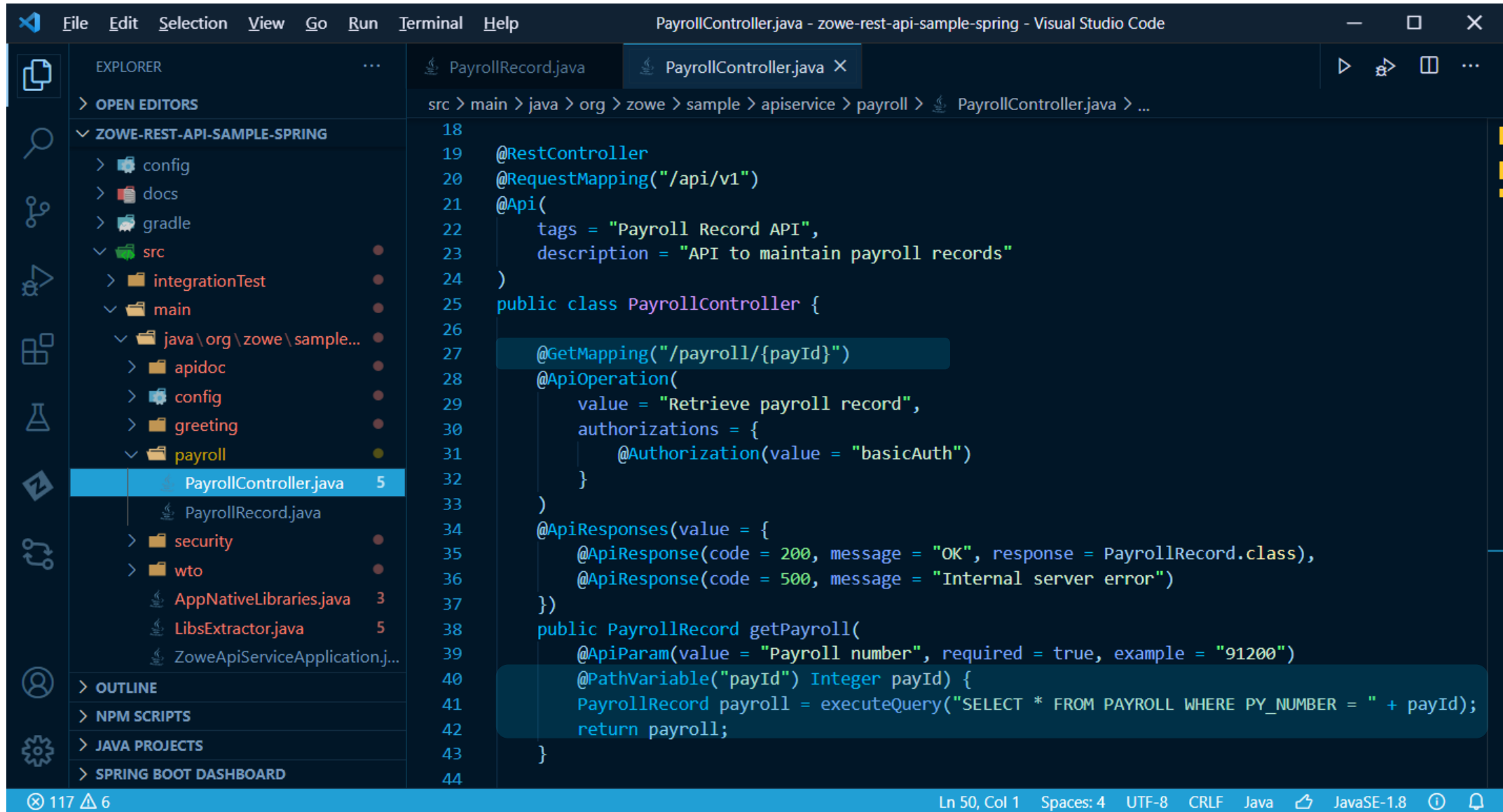
# Business API Solution

- Generate business REST APIs from existing database definitions
- Expose records, tables, system tables, procedures as REST resources
- Enable CRUD operations on database resources with HTTP POST, GET, PUT, DELETE methods
- Integrate with External Security Manager and Zowe API Mediation Layer

# Using APIs for Web Application Development



# Generated Controller Class



The screenshot shows the Visual Studio Code editor with the file `PayrollController.java` open. The Explorer sidebar on the left shows the project structure, with the `payroll` folder selected. The code in the editor is as follows:

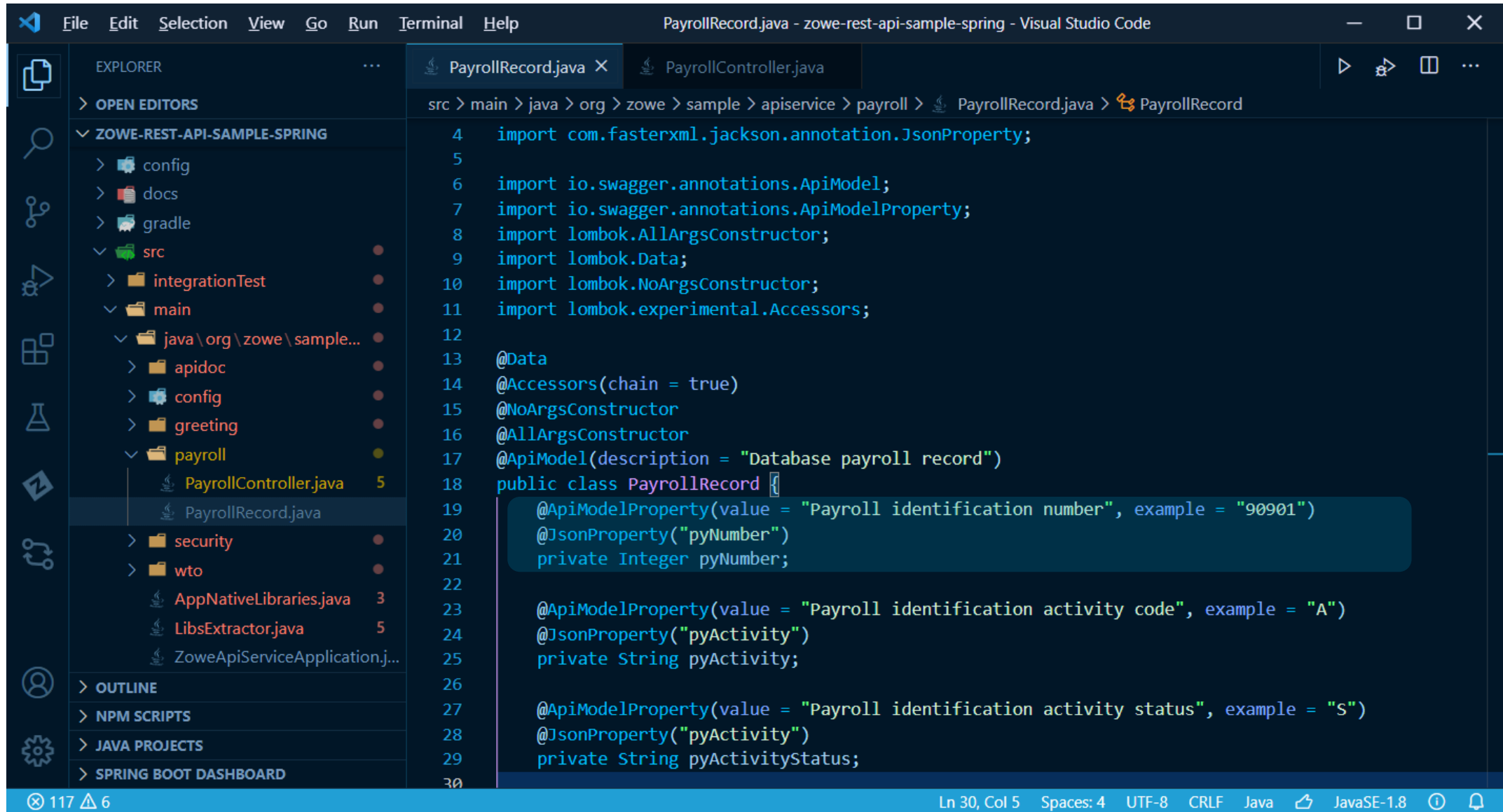
```
18
19 @RestController
20 @RequestMapping("/api/v1")
21 @Api(
22     tags = "Payroll Record API",
23     description = "API to maintain payroll records"
24 )
25 public class PayrollController {
26
27     @GetMapping("/payroll/{payId}")
28     @ApiOperation(
29         value = "Retrieve payroll record",
30         authorizations = {
31             @Authorization(value = "basicAuth")
32         }
33     )
34     @ApiResponses(value = {
35         @ApiResponse(code = 200, message = "OK", response = PayrollRecord.class),
36         @ApiResponse(code = 500, message = "Internal server error")
37     })
38     public PayrollRecord getPayroll(
39         @ApiParam(value = "Payroll number", required = true, example = "91200")
40         @PathVariable("payId") Integer payId) {
41         PayrollRecord payroll = executeQuery("SELECT * FROM PAYROLL WHERE PY_NUMBER = " + payId);
42         return payroll;
43     }
44 }
```



# Generated Swagger API Endpoint Documentation

Payroll Record API			▼
GET	/payroll/{payId}	Retrieve payroll record	🔒
DELETE	/payroll/{payId}	Delete payroll record	🔒
POST	/payroll	Create payroll record	🔒
PUT	/payroll	Update payroll record	🔒
Models			▼
PayrollRecord >			↩

# Generated Model Class



```
PayrollRecord.java - zowe-rest-api-sample-spring - Visual Studio Code

src > main > java > org > zowe > sample > apiservice > payroll > PayrollRecord.java > PayrollRecord

4  import com.fasterxml.jackson.annotation.JsonProperty;
5
6  import io.swagger.annotations.ApiModel;
7  import io.swagger.annotations.ApiModelProperty;
8  import lombok.AllArgsConstructor;
9  import lombok.Data;
10 import lombok.NoArgsConstructor;
11 import lombok.experimental.Accessors;
12
13 @Data
14 @Accessors(chain = true)
15 @NoArgsConstructor
16 @AllArgsConstructor
17 @ApiModel(description = "Database payroll record")
18 public class PayrollRecord {
19     @ApiModelProperty(value = "Payroll identification number", example = "90901")
20     @JsonProperty("pyNumber")
21     private Integer pyNumber;
22
23     @ApiModelProperty(value = "Payroll identification activity code", example = "A")
24     @JsonProperty("pyActivity")
25     private String pyActivity;
26
27     @ApiModelProperty(value = "Payroll identification activity status", example = "S")
28     @JsonProperty("pyActivity")
29     private String pyActivityStatus;
30 }
```

# Generated Swagger API Model Documentation

PayrollRecord ▾ {

pyNumber	number	example: 90901	Payroll identification number
pyActivity	string	example: A	Payroll identification activity code
pyActivityStatus	string	example: S	Payroll identification activity status
fgCurrentRate	number	example: 4221.25	Payroll figures monthly base pay rate
fgYtdWage	number	example: 33771.6	Payroll figures year to date base pay
fgYtdCommission	number	example: 58334.7	Payroll figures year to date commision

# Using the Generated API

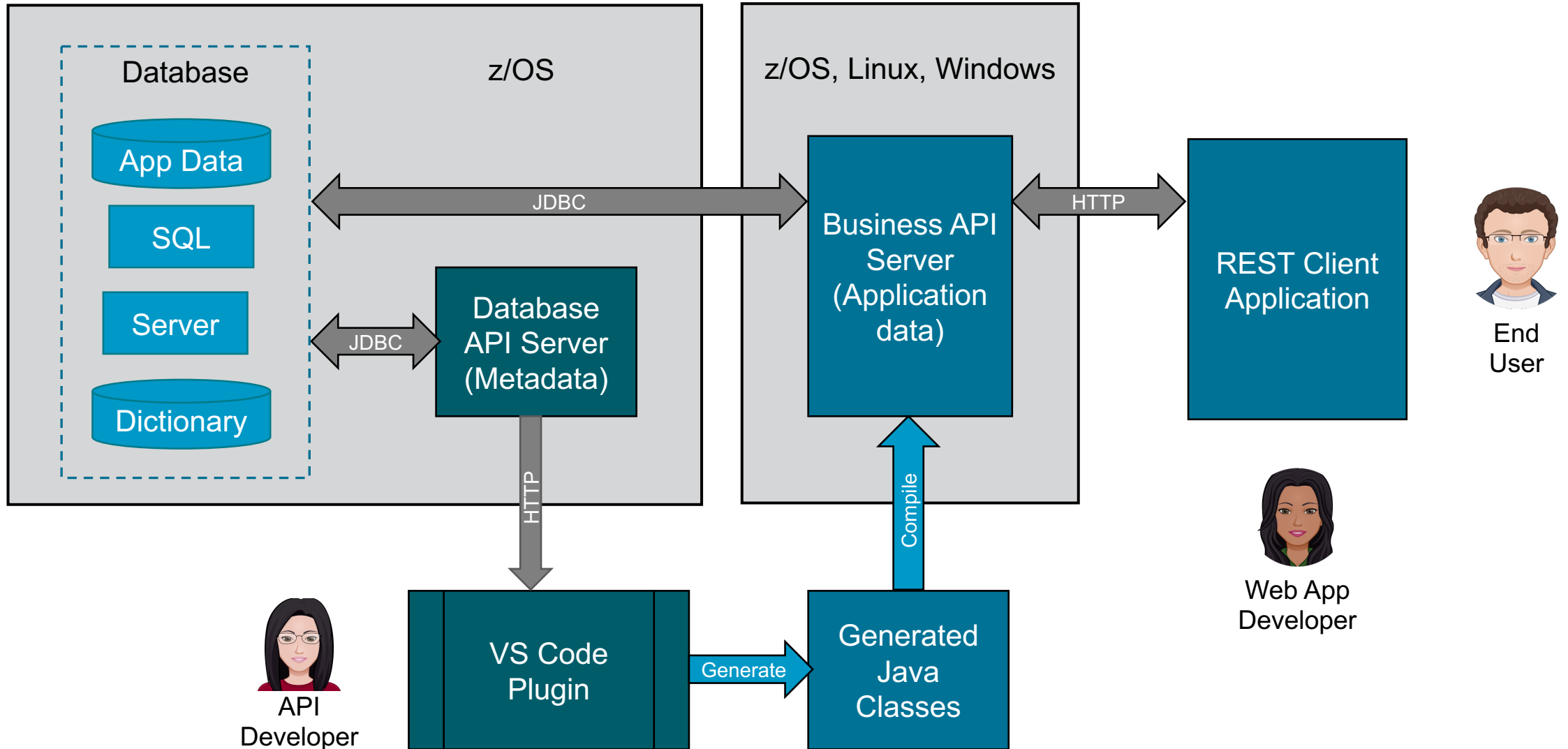
## Request URL

```
https://localhost:10080/api/v1/payroll/90901AS
```

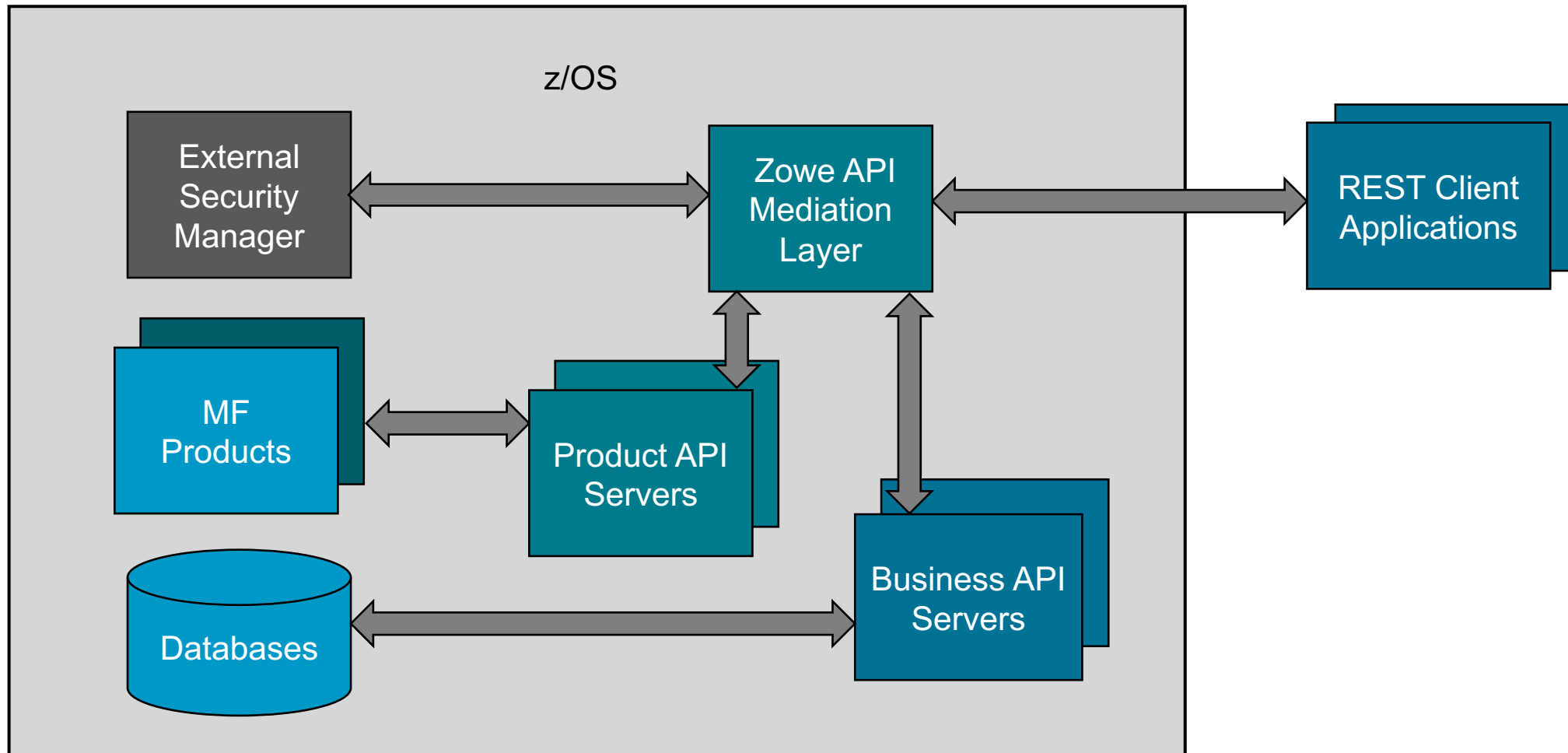
## Server response

```
{
  "pyNumber": 90901,
  "pyActivity": "A",
  "pyActivityStatus": "S",
  "fgCurrentRate": 4221.25,
  "fgYtdWage": 33771.6,
  "fgYtdCommission": 58334.7,
  "fgYtdTax": 15335.88,
  "rcNumber": 10,
  "rcActivityCode": "C",
  "rcActivityStatus": "A",
  "rcCurrentRate": 4221.25,
  "rcYtdWage": 33771.6,
  "rcYtdCommission": 58334.7,
  "rcYtdTax": 15335.88
}
```

# Architecture of Tools to Create Business APIs



# Business API Integration with Zowe



# Comparison with IDMS Web Services

## IDMS Web Services

- SOAP and XML
- IDMS is the server
- User coded services, COBOL or ADS
- Manual XML parsing and generation
- Enables IDMS apps to consume web services

## REST APIs

- REST and JSON
- Spring Boot Tomcat server
- Generated service, Java
- Automatic JSON parsing and generation
- Integration with Zowe API gateway



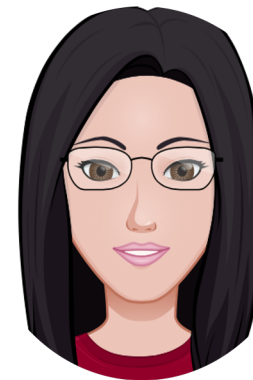
# Summary

- Personas and use cases
- System APIs to manage databases
- Business APIs to expose application data
- Creating business APIs and applications

# Polling Question

Are you interested in creating APIs that expose data in new applications?

- CA IDMS
- CA Datacom





# General Announcements

# Partnering with You – How you Can Get Involved

We're Interested In Your Thoughts



Join us on our **journey** to achieve our ongoing **vision**, we are **actively seeking feedback and suggestions from our customers**. We want to hear your voice!

- Submit your ideas [communities.broadcom.com](https://communities.broadcom.com)
- Vote and comment on ideas that are important to you

- Join our [validation program](#) to influence our product direction
  - End-of-Sprint review meetings
  - Early access to new features and enhancements

# Polling Question

Are you willing to participate in Customer Validation?

- CA IDMS
- CA Datacom

# Monthly Database Community Webcasts

- Mark your Calendars!
- **\*NEW\*** Monthly webcasts!
  - CA Datacom 2<sup>nd</sup> Thursday of each month @ 11:00 (ET)
    - [CA Datacom System Performance REST API](#) (March 11)
  - CA IDMS 3<sup>rd</sup> Thursday of each month @ 11:00 (ET)
    - CA IDMS (March 18 – TBD)
- What topics would you like us to cover?
  - Please indicate within the session survey





# Events

- [Mainframe Technical Exchange](#): **SAVE THE DATE!**

**October 5-7, 2021**

We're excited to announce that the Mainframe Technical Exchange will be held October 5-7, 2021. This no-cost technical educational event will be hosted virtually and available to all Broadcom customers. As the year progresses, we will consider supplementing the virtual event with regional in-person events if possible.

## **What can you expect from our 2021 Technical Exchange?**

- Engage with Broadcom technical experts
- Connect and share Mainframe knowledge with more than 1,000 attendees from 56 countries
- Obtain the latest information on product updates and roadmaps
- Gain new skills and tips & tricks, that can be used immediately

Registration information and a detailed session agenda will be provided closer to the event. Bookmark this site for the latest event information.

Follow [\*\*#BroadcomMTE2021\*\*](#) on LinkedIn and Twitter for the latest event updates.



# Questions





# Thank You