

# Node.js on IBM i

## What is it and what's the big deal?



**Brian May**  
**Director, Customer Solutions**  
Profound Logic

# Node.js on IBM i: What is it and what's the big deal?

## Objectives for this session:

- What is Node.js
  - Overview/History
  - Language basics
  - Editors and dev tools
- What's the big deal?
  - Ecosystem
  - NPM
- What does it mean for IBM i
  - Modern perception
  - More developers



# Product and Services Discussion



**Hang around after the main presentation to learn how **Profound Logic** can make developing in **Node.js** easier!**



<https://openjsf.org/about/members/>

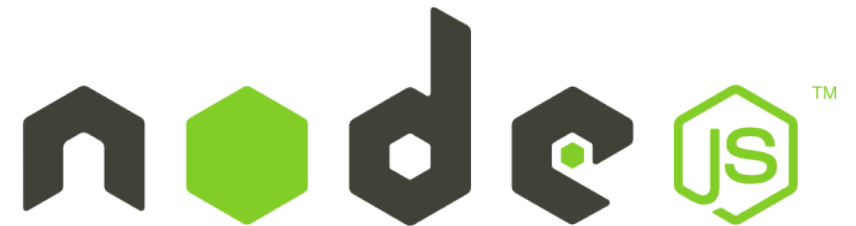
# What is Node.js?



# What is Node.js?

## Simple Definition

Node.js is server-side  
JavaScript



# About Node.js

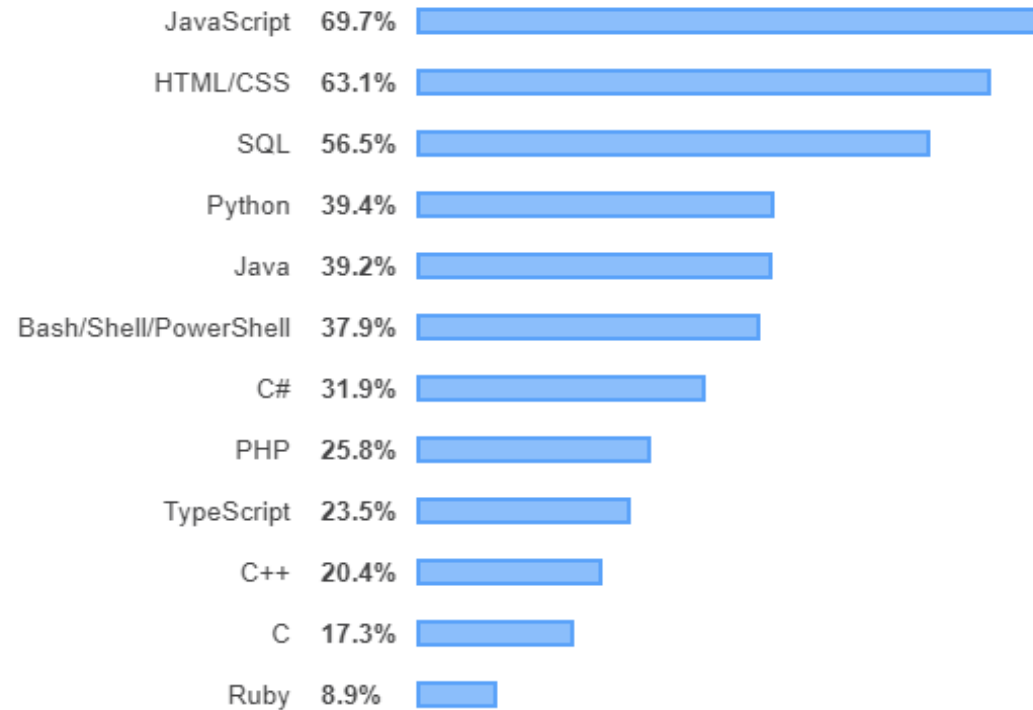
- JavaScript is now the most popular development language in the world
  - Has superseded Java and PHP
- Node.js is truly modern
  - Released for Linux in 2009
- Designed for web and mobile application development
- Same set of skills for server side and client-side development

## Stack Overflow Developer Survey 2019

### Programming, Scripting, and Markup Languages

All Respondents

Professional Developers





# Node.js Usage

- Node.js is used by companies you know

Walmart 

ebay™

PayPal®

DOW JONES

intuit.

NETFLIX

LinkedIn

The New York Times

 Microsoft

 U B E R

YAHOO!

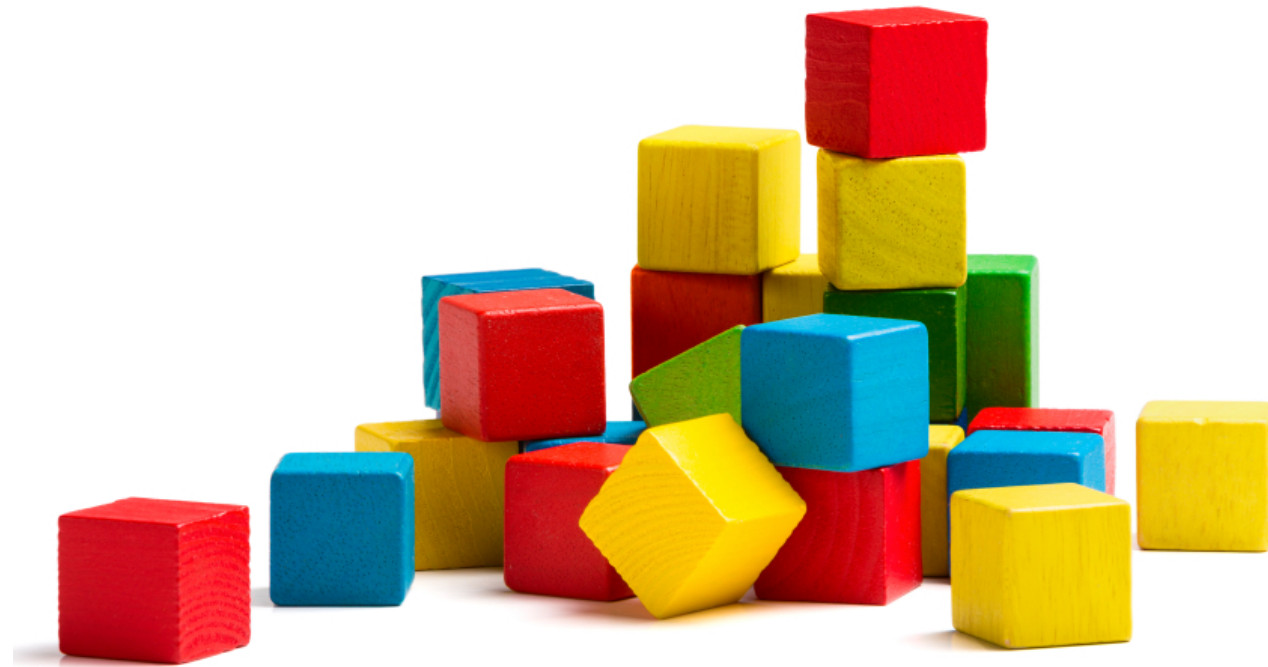
Kingfisher

# Node.js Basics

- We're not going to discuss JS syntax
  - There are thousands of resources on JS coding
  - The syntax for Node.js is the same as JS in the browser
  
- We will however talk about how a Node.js application is structured

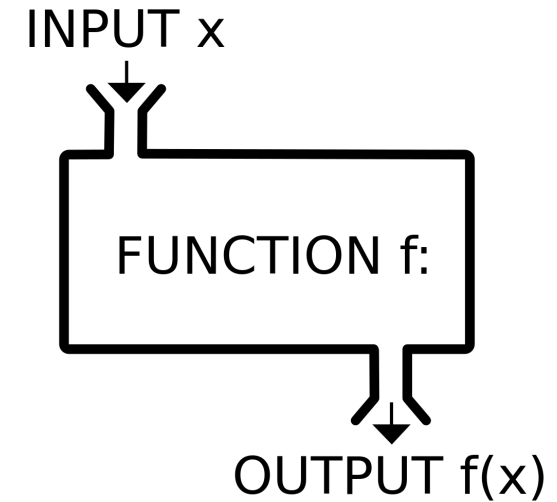
# Building blocks of Node.js

- Functions
- Modules
- Packages



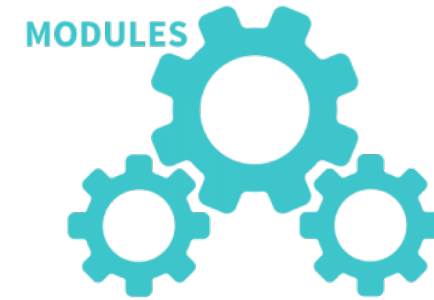
# Functions

- Problem → Write JavaScript Code
- Similar Problem → Reuse the Code
- JavaScript functions can be used to reuse code
  
- Similar to subprocedures in RPG
- Can take parameters and return values



# Modules

- In Node.js, 1 or more functions can be placed in a .js file that represents a Module
- Everything is by default internal to the module
- Unless the **exports** keyword is used



# Exporting from a module

mymodule.js

```
function add(a, b) {  
  return a + b;  
}
```

```
function subtract(a, b) {  
  return a - b;  
}
```

```
exports.add = add;  
exports.subtract = subtract;
```

# Packages



- A file or a set of files / modules in a directory
- One of the files is the main file
- Directory will contain a file named *package.json* with information about the package
  - Name of package
  - Version of package
  - Dependencies
  - Etc.

# Node.js Editors

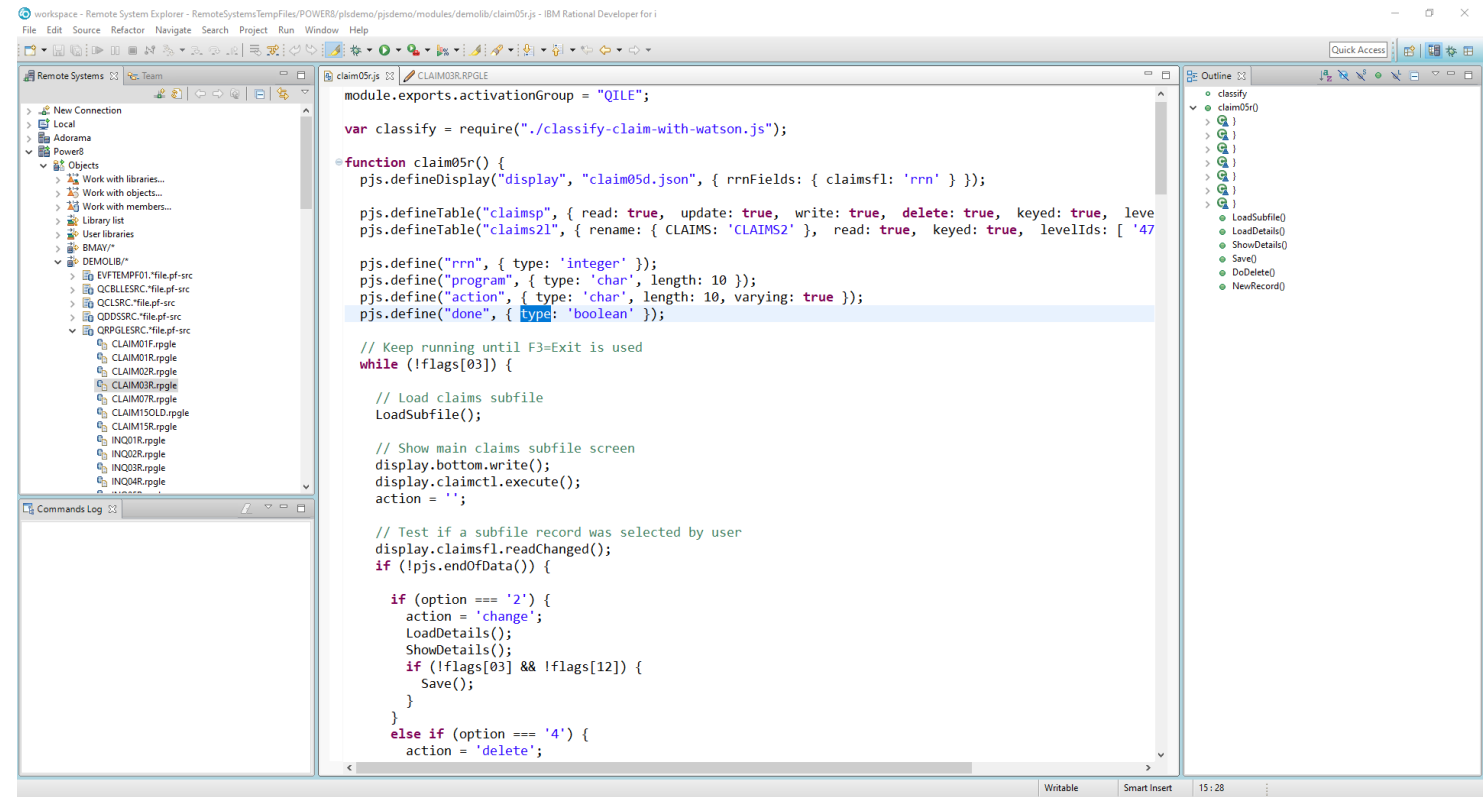


And many more!



# Editor Options

- Rational Developer for i
  - IBM Product based on Eclipse
  - Used for ILE development
  - Familiar for most IBM i developers
  - Great for ILE development
  - Serviceable for Node.js editing



The screenshot displays the Rational Developer for i interface. The main editor window shows a Node.js script named 'claim05r.js' with the following code:

```
module.exports.activationGroup = "QILE";

var classify = require("./classify-claim-with-watson.js");

function claim05r() {
  pjs.defineDisplay("display", "claim05d.json", { rrnFields: { claimsfl: 'rrn' } });
  pjs.defineTable("claimsp", { read: true, update: true, write: true, delete: true, keyed: true, level: 1 });
  pjs.defineTable("claims2l", { rename: { CLAIMS: 'CLAIMS2' }, read: true, keyed: true, levelIds: [ '47' ] });
  pjs.define("rrn", { type: 'integer' });
  pjs.define("program", { type: 'char', length: 10 });
  pjs.define("action", { type: 'char', length: 10, varying: true });
  pjs.define("done", { type: 'boolean' });

  // Keep running until F3=Exit is used
  while (!flags[03]) {

    // Load claims subfile
    LoadSubfile();

    // Show main claims subfile screen
    display.bottom.write();
    display.claimctl.execute();
    action = '';

    // Test if a subfile record was selected by user
    display.claimsfl.readChanged();
    if (!pjs.endOfData()) {

      if (option === '2') {
        action = 'change';
        LoadDetails();
        ShowDetails();
        if (!flags[03] && !flags[12]) {
          Save();
        }
      }
      else if (option === '4') {
        action = 'delete';
      }
    }
  }
}
```

The interface includes a 'Remote Systems' tree on the left, a 'Commands Log' at the bottom left, and an 'Outline' view on the right showing the project structure. The status bar at the bottom indicates 'Writable', 'Smart Insert', and '15:28'.

# Editor Options

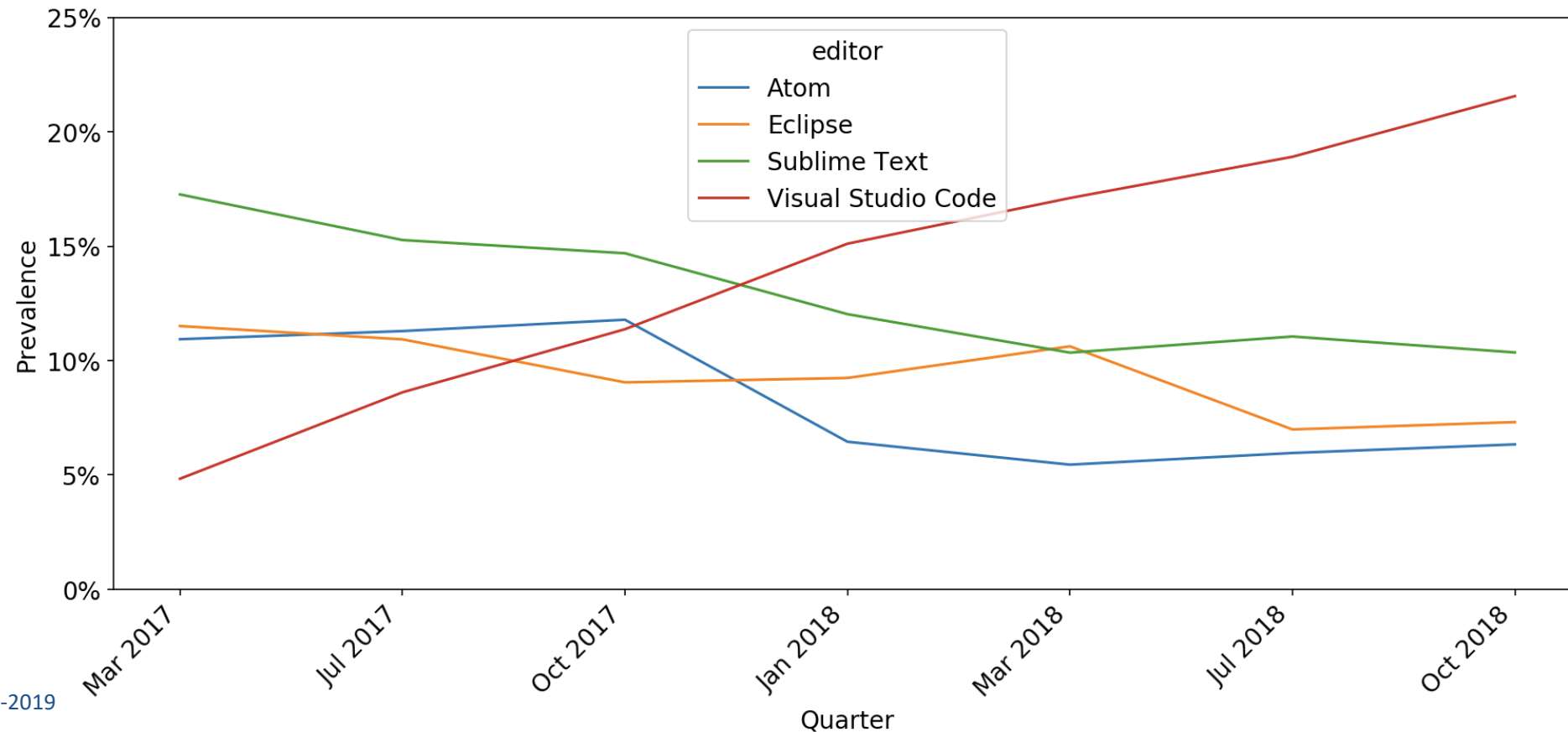
- Visual Studio Code
  - Free and Open Source
  - MIT License
    - Can be used for commercial development
  - Extremely popular
  - Can connect to IBM i
  - Not good for ILE development

```
SSH FS - power8 > pjsdemo > modules > demolib > JS claim03r.js
1
2 exports.activationGroup = "QILE";
3
4 function claim03r() {
5   pjs.defineDisplay("display", "claim03d.json", { rrnFields: { claimsfl: 'rrn' } });
6
7   pjs.defineTable("claims", { read: true, update: true, write: true, delete: true, keyed: true, levelIds: [ '475A4A84923B0' ] });
8   pjs.defineTable("claims2", { rename: { CLAIMS: 'CLAIMS2' }, read: true, keyed: true, levelIds: [ '475A4A84923B0' ] });
9
10  pjs.define("rrn", { type: 'integer' });
11  pjs.define("blahvar", { type: 'char', length: 10 });
12  pjs.define("program", { type: 'char', length: 10 });
13  pjs.define("action", { type: 'char', length: 10, varying: true });
14  pjs.define("done", { type: 'boolean' });
15
16  // * Keep running until F3=Exit is used
17  while (!flags[03]) {
18
19    // * Load claims subfile
20    loadSubfile();
21
22    // * Show main claims subfile screen
23    display.bottom.write();
24    display.claimctl.execute();
25    action = '';
26
27    // * Test if a subfile record was selected by user
28    display.claimsfl.readChanged();
29    if (!pjs.endOfData()) {
30
31      if (option === '2') {
32        action = 'change';
33        loadDetails();
34        showDetails();
35        if (!flags[03] && !flags[12]) {
36          save();
37        }
38      }
39      else if (option === '4') {
40        action = 'delete';
41      }
42    }
43  }
44 }
```

# Editor Usage

- VS Code usage is skyrocketing

– <https://triplebyte.com/blog/editor-report-the-rise-of-visual-studio-code>



# Editor Options

- Profound.js Node Designer
  - Part of the free Profound.js framework
  - Hosted on any platform including IBM i
  - Runs on IBM i
  - No ILE editing at this time
  - Zero footprint (Runs in browser)

```
1 exports.activationGroup = "QILE";
2
3
4 function claim03r() {
5   pjs.defineDisplay("display", "claim03d.json", { rrrnFields: { claimsf1: 'rrn' } });
6
7   pjs.defineTable("claimsp", { read: true, update: true, write: true, delete: true, keyed: true, levelIds: [ '475A4A84923B0' ] });
8   pjs.defineTable("claims2l", { rename: { CLAIMS: 'CLAIMS2' }, read: true, keyed: true, levelIds: [ '475A4A84923B0' ] });
9
10  pjs.define("rrn", { type: 'integer' });
11  pjs.define("blahan", { type: 'char', length: 10 });
12  pjs.define("program", { type: 'char', length: 10 });
13  pjs.define("action", { type: 'char', length: 10, varying: true });
14  pjs.define("done", { type: 'boolean' });
15
16  // * Keep running until F3=Exit is used
17  while (!flags[03]) {
18
19    // * Load claims subfile
20    loadSubfile();
21
22    // * Show main claims subfile screen
23    display.bottom.write();
24    display.claimctl.execute();
25    action = '';
26
27    // * Test if a subfile record was selected by user
28    display.claimsf1.readChanged();
29    if (!pjs.endOfData()) {
30
31      if (option === '2') {
32        action = 'change';
33        loadDetails();
34        showDetails();
35        if (!flags[03] && !flags[12]) {
36          save();
37        }
38      }
39    }
40  }
41 }
```

What's the big deal?



What's the big deal?

**THE COMMUNITY!**

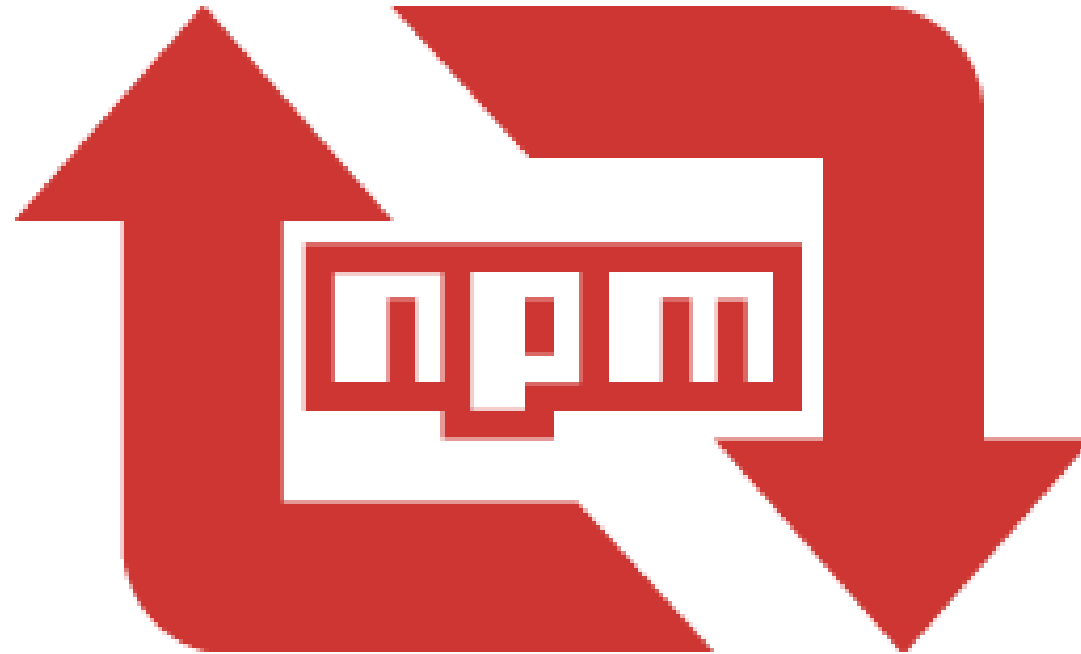
**THE ECOSYSTEM!**

# Ecosystem

- The Node.js language is very small
  - In some languages, developers depend on the language to provide all functionality
  - Node.js provides only the foundation required to develop
  - All other features are provided by packages created and shared by developers



# Why Node.js



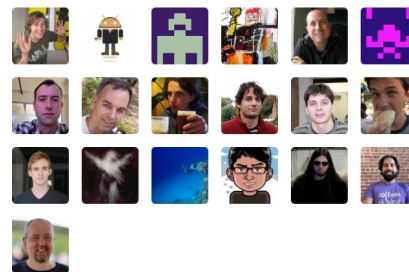


# NPM – Node Package Manager

- Node.js ships with the npm tool
- Uses a public registry of packages
- Packages often depend on each other
- Usage stats for each package gives it a social aspect



## Collaborators [list](#)



## Stats

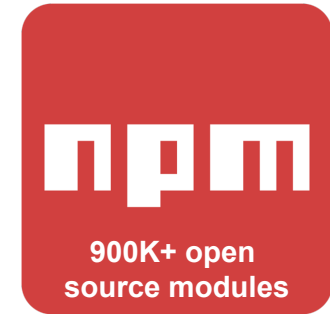
25,015 downloads in the last day

108,628 downloads in the last week

468,587 downloads in the last month

# Wild NPM statistics

- Various packages:
  - Web Services Capabilities
  - PDF, Excel, Image generation and processing
  - Email
  - IoT, AI, Watson API, Cloud
  - Encryption / Security
  - JSON, XML processing, Web, Mobile Tools
  - Integration: drivers and API to access data on any other system
- Close to 1 million packages
- More than 1 billion downloads every day (1,000,000,000)



# Use Emerging Technology Now



## Accessing DB2 for i in Node.js

- IBM created a Node.js module (open source, available via npm) for IBM i database access. The name of the module is idb-connector

<https://www.npmjs.com/package/idb-connector>

```
cd /path/to/myproject
npm install idb-connector
```

- Like all modules, you require it – then can use it.

```
var db2i = require("idb-connector");

var dbconn = new db2i.dbconn();

dbconn.conn("*LOCAL"); // connect to local database

// Same as naming = *SYS in RPG
dbconn.setConnAttr(db2i.SQL_ATTR_DBC_SYS_NAMING, db2i.SQL_TRUE);
```

# Running an SQL Statement

- The SQL results are returned asynchronously in a JavaScript array of objects

```
var stm = new db2i.dbstmt(dbconn);

var sql = "select ORDERREF, ORDERLINE, PRODREF, ORDERQTY," +
          "SALETOTAL from SALESDDL";

stm.exec(sql, function(result) {

    // statement is run in the background
    // this function is called when it is ready

    result.forEach(function(row) {

        // this function is repeated for each row available
        // row.ORDERREF, row.ORDERLINE, etc will contain the fields
        // in string format
    })
})
```

What does it mean for IBM i?



# Perception

- Modern Language
  - IBM i can do more than run 30-year-old green screen applications
  - New workloads
  - New solutions to business needs



# The Cloud





# Developers

- Having Node.js and other modern languages available on IBM i, increases the developer pool available to companies who run their business on IBM i
  - Helps address retiring IBM i workforce
  - Brings new talent and ideas to IBM i



Don't forget!

- After Q&A, I would love to show you some of the technology Profound Logic offers to make developing business applications in Node.js painless



Profound.js



NodeRun

# Questions?



Thank you!

**Web Site:**

[www.profoundlogic.com](http://www.profoundlogic.com)

**Contact us:**

[bmay@profoundlogic.com](mailto:bmay@profoundlogic.com)

 profoundlogic

 company/profound-logic-software

 @ProfoundLogic

## Extending Node with an NPM package

- We decided to build an NPM package to make it easier to build business applications with Node.js
  - Free package
  - Commercial components for advanced capabilities

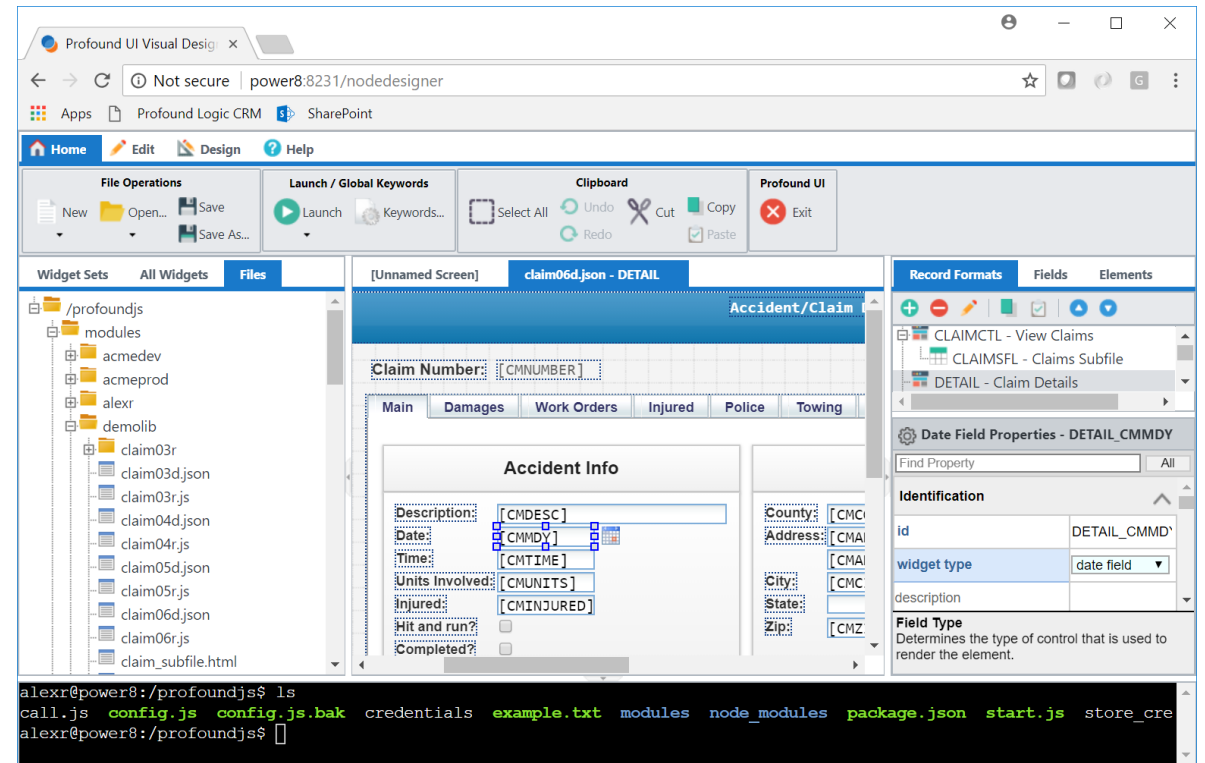


# Free Profound.js Framework on NPM

- Free NPM Package that gives Node.js capabilities similar to RPG
  - Top-down transactional business programming
    - Eliminates Callbacks
    - Eliminates Promises
    - Eliminates Async/Await
  - Support for IBM i data types
    - Packed
    - Zoned
    - Fixed Length Char
    - Data Structures
    - Pointers
  - Various API to work with these IBM i data types

# Free Profound.js Framework on NPM

- Free NPM Package that gives Node.js capabilities of RPG
  - Support for Rich Display or HTML interfaces
    - Rich Display: Many Widgets
    - Manual HTML: More Control
  - Visual Designer Tool
    - Little manual code
    - Drag-and-drop / Point-and-click
  - Complete Browser-based IDE
  - Built-in PASE Terminal
  - Display API similar to RPG's EXFMT



# NodeRun.com – Free Node.js Playground

- NodeRun.com
  - Free Node.js Playground in the Cloud
  - Node.js Preinstalled
  - Profound.js Preinstalled
  - Full IDE and runtime environment with design, editing, and debugging tools
  - Connects the cloud to your IBM i and RPG code via Node.js

