BASIC KNOWLEDGE Of CLOUD SERVICES

IBM Watson IOT Platform

Over the past few years, AI has become a buzzword both inside and outside of the tech community. AI systems are designed to operate in a very human context, on human forms of expression — our language, both written and oral, our physical movement and even our facial expression.Working with Watson is an investment in an AI system built for business. Embracing AI as complimentary to the role humans play in your business will prevent misunderstandings and allow for accelerated growth and business value.

3 things to know about IBM Watson

1.IBM Watson in the competitive world of AI

This unique integration allows you to seamlessly integrate Watson services into your digital applications, products and operations, giving you access to data-driven insights that improve business outcomes and produce real value, in addition to allowing you to store, train and manage your data on the most secure cloud.

2.Domain Expertise

Watson provides the largest base of industry offerings across the board. From healthcare to finance and transportation to energy, Watson understands the language of your industry. Combine your data, industry knowledge and subscription data to train your AI on what you care about in order to make the most informed decisions.

3.Data privacy

Watson can ingest, cleanse and comprehend a variety of data types while optimizing for the robust AI workloads required to extract meaningful insights. With Watson, you own your data, insights and IP. In a competitive marketplace, IBM believes the perspective and insights you derive from applying Watson should not be shared or sold. Your Intellectual Property and insights stay with you — period.In 2016, IBM invested more than $5 billion in research and development and received a record 8,088 patents. Nearly 2,700 of these patents are A.I. related. By choosing Watson you are the direct beneficiary of the advancements and progress that come

from IBM Research.

In order to keep pace and with the evolving world of AI, it is important to stay informed about how the worlds largest technology firms are thinking about and deploying AI in service of people and society.

Some of the best ways to do this are:

Google and Microsoft

Visit Future of Life Institutes AI News, where you will find a wide breadth of current events promoted by the worlds leading AI researchers and scholars

Stay up-to-date on where Watson technology is heading at IBM Research: AI and Cognitive Computing

For a unique perspective on IBM Researchs work, consider following IBM Research Frontiers Institute

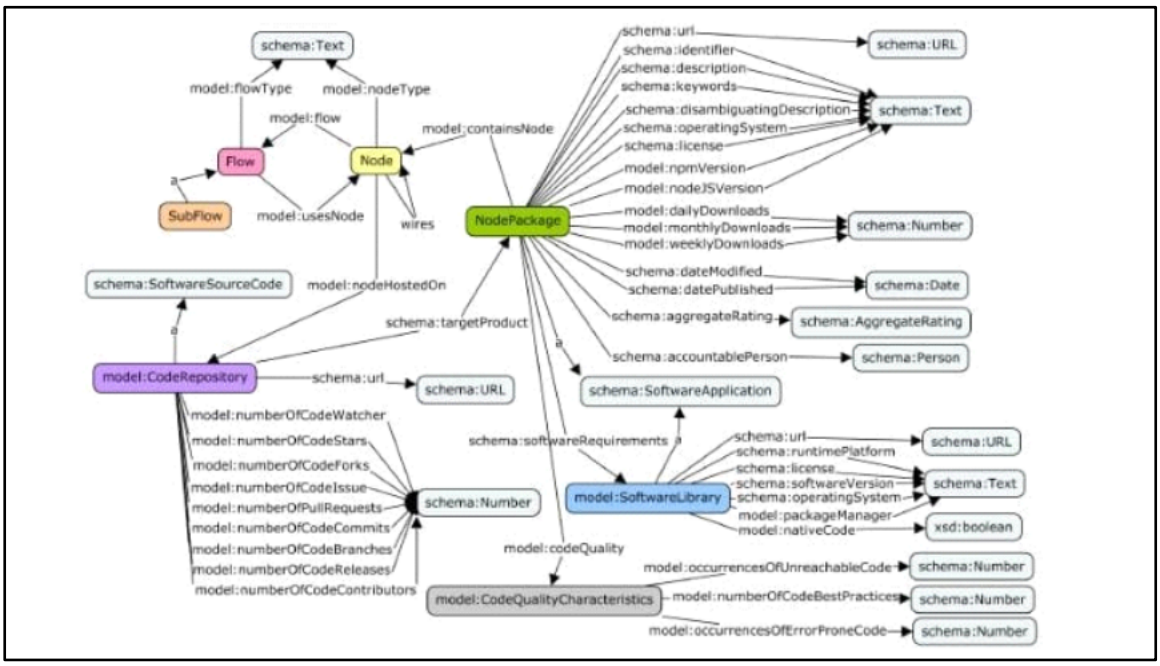
Finally, learn about one of the largest university and industry AI collaborations, and what they are up to, at the MIT-IBM Watson AI Lab

Node-RED Service

Node-RED is comprised of a large ecosystem of nodes forIoT devices and services that makes it a powerful tool for IoT applica-tion development. In order to facilitate the usage of this heterogeneousecosystem in industrial settings, we present here the NOde-red libraryeVAluation (NOVA) approach for gathering the relevant metadata in aknowledge base and first analyses of the data.

[1] A Knowledge Base for Node-REDToday, Node-REDs catalog comprises over 3.000 nodes and flows. A node im-plementation can be seen as an adapter of a device or service that makes theirfunctionality available in the Node-RED ecosystem. Using the Node-RED graph-ical user interface, nodes can then be combined in so-called flows. A flow, whichcan also comprise sub-flows, represents an IoT application [1]. Node-REDs cat-alog contains nodes for IoT platforms and devices (e.g., Xively or Raspberry Pi),Web services (e.g., Twitter), smart home products (e.g., Philips Hue or Ama-zon Alexa), industrial automation (e.g., Siemens S7, OPC UA, or ModBus) oranalytics and machine learning (e.g., IBM Watson).The easy usage of the Node-RED UI to mash-up IoT applications in combina-tion with the rich ecosystem makes Node-RED valuable for industrial enterprisesand their customers. However, descriptions of nodes and flows are currently notaccessible as a structured metadata. This hinders discovery, and does not allowfiltering based on the quality rating or licensing information. More importantlymetadata, containing explicit links from flows to contained nodes, does not existat all. All these shortcomings hamper the usage of Node-RED nodes and flowsin industrial settings.In order to address these challenges, we have developed the NOde-red libraryeVAluation (NOVA)

approach1, which automatically collects large amounts ofmetadata to feed a knowledge base for further analyses.At its centre, NOVA comprises a Web crawler that harvests available meta-data about nodes and flows, and stores them in a triple store. This is howthe NOVA knowledge base is created. The process works similarly to DBPe-dias extraction manage



[2].While typical crawler approaches for the IoT (e.g. [4]) focus onharvesting sensor data from public IoT platforms, we address here the collectionof metadata for IoT device adapters and application workflows.The entry point for our crawler is the Node-RED catalog2, from which adescriptive Web page for each node package can be found. For example, onthe page of the node that connects Particle devices3, a list of contained nodescan be found, as well as general metadata (e.g., version, rating, keywords, ormaintainers). This metadata is stored in the class NodePackage. The crawlerfollows links to listed Web resources: (1) the NPM repository and (2) the GitHubproject, and harvests further metadata from these pages.Using NPM, the crawler downloads the software package and installs it lo-cally. Thereby, all dependent libraries of the node package are downloaded, andtheir metadata is stored using the SoftwareLibrary class. This includes informa-tion on license and version, and gives a hint whether the native code is included.This report is relevant for the choice of execution environment. Using GitHubsREST API, the crawler gathers all available metadata regarding the source code(e.g., quality rating and number of commits), and stores this information in theCodeRepository class. Additionally, the source code itself is downloaded by thecrawler and an automated analysis is triggered using PMD4.

This static source a Knowledge Base for the Node-RED IoT Ecosystem 3code analyzer detects unreachable code, broken best practices, and error pronecode. The number of each kind of incident is stored in the CodeQualityCharac-teristics

class.Finally, all flows are accessed via the Node-RED catalog and their metadatais stored in the Flow class. Each flow references multiple nodes that make up itsstructure. Thereby, a node that is part of a flow specifies, via the wires property,to which other nodes its outputs are connected.2 Data AnalysesThe crawling described above runs for several hours and produces currentlyaround 650.000 triples. Using SPARQL, we analyzed the resulting knowledgebase. As a starting point, we counted node packages (1.593) and their (directlyand transitively) dependent libraries (4,482). In order to make these usable incommercial settings, a license clearing is required. With the installation of theNPM package (Section 1), we gathered those license-related data. In the post-processing, we took the SPDX license list as a reference5and cleaned the licensenames by ignoring case, spaces and - characters. Thereby, 105 out of overall 128licenses match an SPDX license. The most used license is MIT (41.528), followedby ISC (7897) and Apache 2.0 (4140). The long tail of licenses (113) have lessthan 10 occurrences. Now, having the license metadata available enables us toautomatically check the license compatibility for a specific node package.Next, we address a known issue with the Node-RED specification of fl ows:when a flow refers to some node, reference is not done by node package identifi erbut simply by name of the node type. Consequently, it can be difficult for the userto find the correct node package, which is required to run a desired flow. Thisis an even more pressing issue, as the name of the node type can be ambiguous.In the overall 984 flows, we found 20.311 node references to the 5.010 nodesthat are available in the 1.593 node packages. From these node references, 8.112are ambiguous and coming from 90 different node packages. For instance, in aflow6implementing smart metering on a Raspberry Pi, the listed "server" nodeis ambiguous, as it is used in 2 node packages. In order to reuse this flow, a userwould have to manually determine the correct node package.

We address this issue by disambiguating nodes in flows. Listing 1.1 links nodespackages to node references in flows, if they are unambiguous (i.e. if flows refer toa unique node type value). The query in Listing 1.2 builds upon it and attemptsto resolve ambiguity by looking at ambiguous and unambiguous nodes from thesame repository (or node package) used in the same flow. Rationale: nodes ofthe same package tend to be used together. 311 flows could be disambiguatedthis way (out of 984). In the dataset, 120 nodes with a valid repository URL areambiguous, 75 of them are used in at least one flow.3 Conclusions & Future WorkWe present the NOVA approach, and showcase e.g. the automated license clear-ing that can be extended similarly to [5], by checking composite license character-istics for compatibility with the planned usage of a node. Second, we addressedthe issue of missing links from flows to contained nodes. In fact, the name oflisted nodes can be ambiguous. Using SPARQL, we show a two-fold queryingprocess that addresses this issue and solves ambiguity for many nodes.The possibilities for future use and research on the created knowledge baseare broad. Automated quality checks or indexes can be developed based ondifferent input parameters from the NOVA model. This indicates to the usersif a component can be utilized or not. Discovery could be improved too.

Thiscan be achieved by semantically annotating nodes and flows with categories,

ortransforming their keywords into links to well-defined terms. This way, links toa more general knowledge base, such as wikidata, could be built up rmissions, replicate a sample database, querydata, work with the HTTP API, and access documentation and support resources.Your first step in your Cloudant journey is to sign up for a free account. Provide the requested informationto register. Your first step in your Cloudant journey is to sign up for a free account. Provide the requestedinformation to register. Cloudant offers a number of data centers to choose from for your account, andrecommends you choose the cluster that is geographically closest to your application.When you sign in, you are immediately brought to your dashboard, so your next step is to create adatabase.The database name should contain only lowercase alphanumeric characters and no spaces.Cloudant is a NoSQL document database meaning that a database contains a collection of JSON formatteddocuments. So, to add data to your database, you add a new document to the database.The dashboard includes an easy-to-use JSON editor that helps you ensure the documents use the correctJSON syntax. A JSON document includes a set of key value pairs: a name and its value. When you create adocument, Cloudant automatically generates a unique document identifier, or you can provide your own.

The value of the id key is how the database system identifies each document and must be unique in eachdatabase.Cloudant allows you to use a flexible schema. Each document can use the same schema or use a uniqueschema. The values in a document can contain numbers, strings, nested objects, arrays, or Boolean data.Save this document, and go back to the dashboard. Youll notice that Cloudant automatically addedanother key-value pair beginning with rev which tracks document revision history.Add more documents in the same way.Cloudant lets you set permissions for an individual database. If you share this database with users, thenthey will be able to access the database from their dashboard.Generate an API key to provide credentials for programmatic access to a database.You can perform basic create-read-update-delete operations on documents by directly referencing thedocument ID.Additionally, Cloudant creates the primary index for every database out-of-the-box and stores it in a b-tree data structure. Cloudant uses the document ID as the primary key. The primary index is most usefulwhen you can find documents based on their ID.Cloudant builds a secondary index using MapReduce, and stores it in a b-tree data structure, also. Thesecondary index is useful when you need to analyze data or get a range of keys.

For example, to countdata fields, sum or average numeric results, gather advanced statistics, and group by date.The search index is built using Lucene search which is a unique feature of Cloudant. The search index isuseful to perform ad-hoc queries, find documents based on their contents, or work with groups, facets, orgeographies.The Geospatial index is also unique to Cloudant and is stored in an r-tree data structure. The geospatialindex is most useful for complex geometries, advanced relations, and GeoJSON

Cloudant DB

Cloudant Query uses mongo-style querying and is useful for ad-hoc queries, when using many logicaloperators, or if you are familiar with querying data using MongoDB or SQL.The Cloudant web site includes examples and tutorials for each type of index to help you get started. Ifyou sign in, then you will be able to add any of the sample databases to your Cloudant account.This process uses the replication API under the covers to replicate the selected sample database to yourdashboard. You can create as many databases as you need in your account — each with any number ofdocuments.The next time you load your dashboard, youll see the animaldb sample database in your list of databases.The animaldb database includes a design document defining any secondary indexes that the databaseshould have. And the rest of the documents contain the information for each animal.Cloudant leverages an HTTP API with the API URL giving you direct programmatic access from anapplication or from the command line with the cURL utility.

From here, you can also view the JSONdocument at the specified URL.The Cloudant HTTP API follows this hierarchical model.AccountDatabaseDocumentAttachmentA URL for the Cloudant API is made up of an account name, the database within that account, and theendpoints to manipulate data within that database.This example references the ablanks account, the employee_directory database, and for all documents inthe database, show the document body. You could use this programmatically to populate a web page thatshows all employees.You make HTTP requests using these verbs:GETPUTPOSTDELETECOPYTypically, when you access the data from a browser, you perform a GET; however, you can use browseradd-on tools or a command line tool to PUT, POST, DELETE, or COPY data.You can use the API to perform all of these requests:Insert dataRead dataCreate indexesMake queriesMonitor the databaseCreate replication jobsOr Create databasesHeres what an HTTP API command looks like.The first part indicates that this is an HTTP request

Next, include a verb such as GET or PUT.You can include headers, data that you are passing, then the URI and any parameters.This cURL example executes a GET, passes the user credentials, and includes the base URI to access.To start, its important to have a set of useful tools installed on your environment for accessing theCloudant HTTP API.cURL is a readily-available command line tool. You can use a Linux shell to issue cURL commands. ManyLinux distributions have cURL preinstalled, so its best check your environment before installing this utility./jq is a useful cURL add-on tool for manipulating the JSON response from the Cloudant HTTP API so it isformatted in a more readable way.Given that the API is HTTP, you can also access it directly from your browser. JSONView is a useful plug-infor formatting the JSON response, equivalent to jq for cURL. JSONView is great tool for both FireFox andChrome browsers. Instead of seeing raw JSON text, JSONView formats the JSON text to make it morereadable.RESTClient and POSTMan provide an easy user interface to manipulate the API, such as sending requeststo the Cloudant database, parsing a response, specifying a URL with credentials to authenticate, or addingheaders. This screen shows POSTMan in Chrome, but RESTClient in FireFox

works similarly and are bothgreat tools for those who prefer to use a browser over a command line.And if you prefer a visual experience, the Cloudant Dashboard provides an intuitive user interface for themajority of the API functionality.If you need help at any point, the dashboard provides access to the documentation.Or you can contact support in one of three ways: submit a new support case, emailsupport@cloudant.com, or join the #cloudant channel on Internet Relay Chat (IRC). Follow@CloudantStatus on Twitter to get information on the status of Cloudant clusters. You can also engagewith a Cloudant specialist on cloudant.com through LiveChat.That brings us to the end of the fifth and last lesson in this course. This lesson provided you with theinformation you need to get started using IBM Cloudant