KARPAGAM
INSTITUTE OF TECHNOLOGY
Inspiring Innovation
Approved by AICTE and Affiliated to Anna University

KARPAGAM
INSTITUTE OF TECHNOLOGY
COIMBATORE -641 105

DEXAR

2019 - 2020

DEPARTMENT OF
COMPUTER SCIENCE
& ENGINEERING

EVERY PROGRAMMER
IS AN
" GENIUS "

VOL 4 ISSUE 1

DEXAR

Contents

■ Vol4 ■ Issue1

Mr.ROHITH KUMAR.S

A few years ago, Mark Reinhold, Chief Architect of the Java Platform Group at Oracle, wrote in a blog post titled "Moving Java ForwardFaster" that Java needed to advance more quickly in order to becompetitive.

A Java framework is specific to the Java programming language and serves as a platform for creating Java applications and web applications. The aim of frameworks is to provide a common structure so that developers don't have to redo it from scratch and can reuse the code provided. It also allows designers and developers to focus on creating unique features for their web- based projects rather than re-inventing the cycle through coding. Spring is one of the most widely-used Java frameworks primarily for the development of web applications. Spring supports suchthings as application events and listeners, externalized configuration, YAML, and type-safe configuration.Micro service frameworks can be used for deploying Java. SpringBootis probably the best Microservice in the Java framework thatworks on top languages for Inversion of Control, Aspect-OrientedProgramming, etc

.

JavaScript (js) is a light-weight object-oriented programming language JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by 98.0% of all websites. Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval. It has seen widespread adoptionfor fast analytical querying.The required skill for the upcoming years is to understand the basics on java frameworks, Java script and microservices can enrich the developing knowledge of each and everyone.

Editorial Student Member :ROHITH KUMAR.S

Introduction

The Data Cleaning with Python and Pandas series includes this article. Its goal is to help developers become proficient with data science tools and methods quickly. Most data cleaning procedures start with an analysis of the dataset to identify and address any missing data. Missing values or data commonly occur in datasets when no data value has been stored, usually when the value is calculated from other variables or is not required in a form. The display and interpretation of data sets can be significantly impacted by data missing from a data set, which happens rather frequently.

CORRECTING MISSING DATA IN PANDA

In this short article, you'll learn how to inspect a dataset using Pandas and identify any potential missing data, how to replace those data, and the functions Pandas afford for the many options of doing so

Finding the Missing Data Elements:

First, we have to find out how much data and what values are missing before deciding what we can do to interpret the missing values. Pandas Data Frames has some great methods for doing this, including:

- The isnull method checks to see if each field is null.
- The sum method adds together fields that are passed into it.

Pandas are a highly utilized data science library for the Python programming language. One of the many reasons Pandas has become the de facto data processing library is the ease with which it allows developers to find and replace missing values in datasets.Pandas is one of the defacto data science libraries for Python and for good reason. This library makes replacing missing values in one's data a breeze. The DataFrame class is host to several methods designed specifically for this use case. In this article, we'll cover three of the most common methods used to replace missing data in Pandas. We'll take a stepwise approach covering the following stages:

- Obtain data with missing values
- Check data for missing values
- Replace missing values

These are three basic concepts but we can find it important to have an explicit step-by-step approach to dealing with what is often a very messy situation. Fortunately, Pandas doesn't require any complicated syntax to move mountains of data.

First, let's start a new code block and enter the following:

PYTHON

```
customers.isnull()
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	state	postcode
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	True	True	True	False	False	True
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	True	True	False	False	True	True	False
...	...	...	...	...	...	...	...	...	...	...	...
995	False	False	False	False	False	False	False	False	False	False	False

customers.isnull()

This writes the table with a check on each field in the table to see if it's a null value.

Let's expand this a little to get a summary:

PYTHON

customers.isnull().sum()

```
In [42]: customers.isnull().sum()
Out[42]: id              0
         first_name      0
         last_name       0
         email         122
         gender         43
         street_num    262
         street_name    37
         street_suffix  37
         city           79
         state          80
         postcode      157
         dtype: int64
```

Dropping Missing Pandas Rows

Next, we're going to remove some of the rows that we can't interpret based on the fields, in particular city and state. If the customers haven't supplied this information, we'll try dropping those rows to see how much of our total data this affects.

Start a new code block and add the following:

PYTHON
customers.dropna(subset=['city'],inplace=True)
print(customers.shape)
print(customers.isnull().sum())

This drops all rows with null value in city column.

```
customers.dropna(subset = ['city'], inplace=True)

print(customers.shape)
print(customers.isnull().sum())
```

```
(921, 6)
id            0
first_name    0
last_name     0
gender       39
city          0
state        73
dtype: int64
```

Replacing Values

This just leaves gender fields with blank values, but we're going to use a different method for this field and replace these with the static value "Unspecified".

- The filling method fills columns or rows using the specified value

Start a new code block and add the following:

PYTHON
customers[“gender”].
fillna('Unspecified', inplace=True)
print(customers.shape)
print(customers.isnull().sum())

```
customers["gender"].fillna('Unspecified', inplace=True)
products["company"].fillna('Unspecified', inplace=True)

print(customers.shape)
print(customers.isnull().sum())
print(purchases.isnull().sum())
print(products.isnull().sum())
```

```
(848, 6)
id            0
first_name    0
last_name     0
gender        0
city          0
state         0
dtype: int64
id              0
purch_date      0
customer_num    0
product_num     0
amount          0
paid            0
dtype: int64
id         0
product    0
cost       0
company    0
dtype: int64
```

Conclusion:

If you look at the output now, all of our columns have values. Even though we haven't looked too closely at the products, we'll also add the value "Unspecified" to missing values in the company column in the products DataFrame.

Mr-Bharath.K (IV CSE)

Hayden AI Releases Safe Sense: A New App for Reporting Traffic Safety Violations

California's San Francisco, January 28, 2020 — Today, Hayden AI Technologies, Inc. ("Hayden AI") declared that the beta version of their Safe Sense App will be made available for public testing in March 2020.

Hayden AI's co-founder and CEO, Chris Carson, expressed his excitement over the release by saying, "We're incredibly happy to share our technology to the public. Our Safe Sense App automatically gathers real-time data to aid in the enforcement of traffic safety legislation using the camera on your smartphone and machine learning algorithms. Everyone is given the means to contribute to enhancing traffic safety, preventing traffic fatalities, and enabling efficient transportation through the use of this affordable and scalable technique.

The programme promotes civic engagement by making it simple for users to report traffic safety infractions like speeding, reckless driving, running stop signs and red lights, and illegal passing of school bus stop arms right from their phones. By signing up for the Safe Sense network, anyone can contribute to the elimination of traffic fatalities and the reduction of congestion through the real-time reporting of risky driving behaviour. All with the intention of returning control of municipal safety to its residents.

With the help of artificial intelligence (AI), the Safe Sense App transforms your smartphone into a dash cam that records road safety violations while also rewarding you.

Here are some key takeaways for the Nullish Coalescing Operator.

- The right argument of ??is evaluated only if needed ("short-circuiting").
- ?? has lower precedence than ||.
- ?? cannot immediately contain, or be contained within, an && or || operation.
- The right argument is selected if the left argument is null or undefined.

NULLISH-COALESCING OPERATOR

When performing property accesses, it is often desirable to provide a default value if the result of that property access is null or undefined. At present, a typical way to express this intent in JavaScript is by using the || operator:

```
const response = {
    settings: {
nullValue: null,
    height: 400,
animationDuration: 0,
headerText: '',
```

5 Javascript (ES6+) features that you should be using in 2019

NEW UPCOMING JAVASCRIPT FEATURE

JavaScript is evolving. Every day, new language ideas and features are discussed and proposed. The TC39 committee is responsible for the specification and development of the ECMAScript (JavaScript) language.

OPTIONAL CHAINING

When looking for a property value deep in a tree-like structure, one often has to check whether intermediate nodes exist:

const street = user.address && user.address.street;

Say we're iterating over an array:

const data = []

data.map(item => {item})

The solution is to check whether our data exists

and isn't empty by checking the array length:

if (data &&data.length> 0) {

data.map(item => {item})

}

/* or */

data &&data.length> 0 &&data.map(item =>

Javascript: ES6

A major release that took six years to finalize. Since then, Technical Committee 39 (TC39), the body in charge of developing the ECMAScript standard, has been releasing a new edition of the standard every year. This annual release cycle has streamlined the process and made new features rapidly available, which the JavaScript community has welcomed.

This year, ECMAScript 2019 (or ES2019 for short) will be released. The new features include Object.fromEntries(), trimStart(), trimEnd(), flat(), flatMap(), description property for symbol objects, optional catch binding, and more.

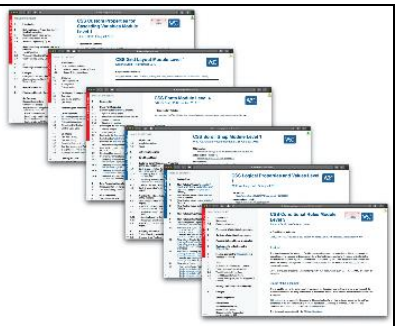
The good news is that these features have already been implemented in the latest versions of Firefox and Chrome, and they can also be transpiled so that older browsers are able to process them. In this post, we will take a good look at these features and see how they upgrade the language.

```
  showSplashScreen: false
   }
  };
  constundefinedValue =
response.settings.undefinedValue ||
'some other default'; // result: 'some
other default'
  constnullValue =
response.settings.nullValue || 'some
other default';
  // result: 'some other default'
```

{item})

It's hope that declaring static properties in the class body will be cleaner and do a better job of meeting programmer expectations of what classes should be for.

```
class CustomDate {
                static epoch = new
CustomDate(0);
}/* or */
                class CustomDate {
                                }
CustomDate.epoch = new CustomDate(0);
```



STATIC CLASS FEATURES

This proposal adds three features to JavaScript classes, building on the previous class fields and private methods proposals:

Static public fields -Static private methods -Static private fields

CONCLUSION:

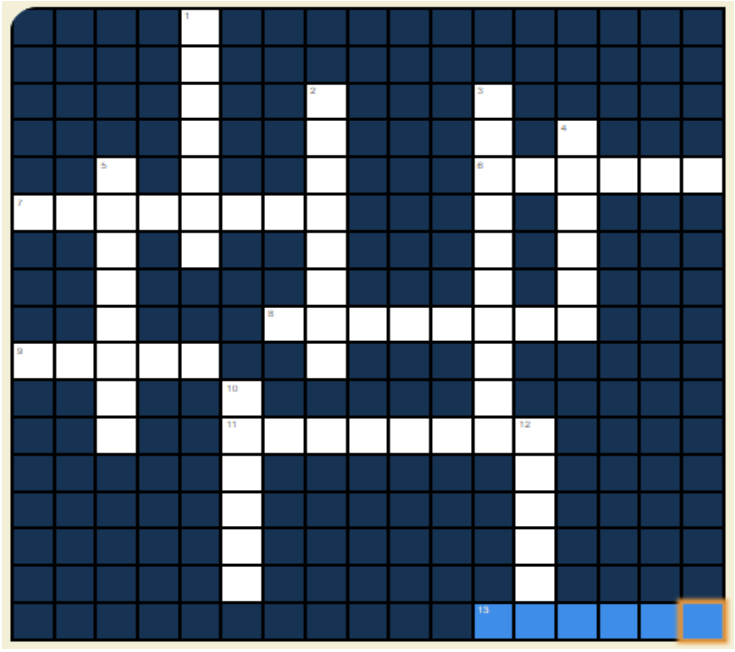
There are many more proposals and I urge you to take a look at them. Everyone has a say, and a vote, your voice matters for shaping the future of the JavaScript language.

Mr-Lokesh.v (IIICSE)

Test your knowledge

"Ask Questions to find out something about the world itself not to find out whether or not someone knows it"

1. LEARN TECHNOLOGY CROSSWORD



Down:

1. You do this to put software on your computer
2.This allows you to connect with the world
3. This helps you with your maths
4. To put things on the internet
5.Secret letter and numbers to access to your computer
10.you do this to put files on your computer
11.Remove a file

Across

6. Another name for a computer
7. Another word for a title
8. You use this to type on a computer
9. You use a camera or phone to take this.
11. Some people send this when they are on holiday.
13.A camera on your computer.

2. GOLD BAR PUZZLE

You've got someone working for you for seven days and a gold bar to pay him. The gold bar is segmented into seven connected pieces.

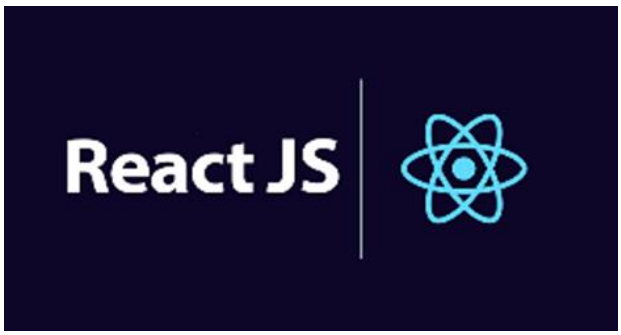
You must give them a piece of gold at the end of every day.

What and where are the fewest number of cuts to the bar of gold that will allow you to pay him $1/7^{th}$ each day?

3. 25HORSES AND 5TRACK PUZZZLE

Mr John have 25 horses, and he want to pick the fastest 3 horses out of those 25. He has only 5 track that means only 5 horse can run at a time, even he don't have a stop watch . What is the minimum number of races required to find the 3 fastest horses?

For answers refer Pg.No 17



You can't keep putting off learning React.js as a programmer or blockchain developer because it's one of the best technologies for building robust web applications and dApps.

LEARN TO CREATE REACT FILES

Go ahead and create an App.js file inside your source folder. Notice how the first letter is uppercase to indicate that the file is a React component, an element that you can use inside other components.

CODE:

```
import React from 'react'
import { render } from 'react-dom'
function App () {
  return (<div>The app has been setup</div>)
}
render(<App />, document.querySelector('#root'))
```

SETUP PACKAGE.JSON SCRIPTS

The package.json file is extremely important since it contains essential configuration about your overall project. There are all the dependencies used in the project so that others can install the same required tools to run the same code without issues.

Remove the "test" line inside your "scripts" section and instead write the following:

CODE:

```
"scripts": {
  "watch": "webpack -dw",
  "compile": "webpack -p",
  "start": "node server.js"
},
```

SETUP A STATIC EXPRESS SERVER IN NODE.JS

CODE:

```
const express = require('express')
const bodyParser = require('body-parser')
const { join } = require('path')
const app = express()
const port = 8000
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({extended: true}))
app.use((req, res, next) => {
  // Logger
  let time = new Date()
  console.log(`${req.method} to ${req.originalUrl} at ${time.getHours()}:${time.getMinutes()}}`)
  next()
})
app.use(express.static('dist'))
app.listen(port, '0.0.0.0', (req, res) => {
  console.log('The server has been started')
})
```

Finally we are using express.static('dist') to allow users to freely access all the files located inside the distribution folder.

UPDATING STATE, USING EFFECT AND IMPORTING FILES

Start by creating a components/ folder inside your src/ since we need a place to store all of our components except for that initial App.js. Then, inside components/ create a file called InputBox.js.

CODE:

```
import React, { useState, useEffect } from 'react'
export default ({ props }) => {
 const [value, setValue] = useState('')
 useEffect(() => {
  setValue('Empty')
 }, [])
 return (
<div className="input-box">
<input
      type="text"
      value={value}
      onChange={e => {
       setValue(e.target.value)
      }}
    />
<p>Your input is: {value}</p>
</div> )
}
```

CONCLUSION

This has been a short article that I created as an introduction to those interested in React development as a quick summary of the core things you need to know. The setup process is laborious but it makes perfect sense once you do it multiple times. React on itself is a completely different javacript than the one you may be used to.

Mr-Joel Johnson (III CSE)

THE EASY UPDATE LIBRARY FOR YOUR PROJECT

INTRODUCTION:

Applications that do not automatically update irritate me. Really,This library will perform every task for you automatically. All you need to save the updates is a web server that can execute PHP.

FEATURES:

• Full or differential updates
• Silent update
• Compression per item
• Detached databases

WINDOWS APPLICATION

• Include tu.hpp
•Instantiate a TU::TUobject:

C++

```
TU(const char* prjg,constwchar_t* host,constwchar_t* path,
bool SSL = false,unsigned short Port = 0,DWORD flg = 0,const wchar_t* un = 0,
constwchar_t* pwd = 0,const wchar_t* uploadpwd = 0)
```

PARAMETERS:

- The unique project GUID created by tu.php admin panel
- The hostname
- The path to the PHP script
- true/false for SSL
- Optional port (if 0, 80 or 443 are used)
- Flags for InternetConnect() function
- Optional username/password for your server
- The upload password if you plan to upload files

C++

```
HRESULT Upload
 (std::function<HRESULT(size_t sent, size_t total, void*)>func =nullptr,void* lp = 0);
HRESULT hr = tu.Upload();
```

C++

```
vector<tuple<wstring, string>> tux;
auto a = L"m.docx";
```

tux.emplace_back(make_tuple<wstring,

string>(forward<wstring>(a), string("A44BC1B3-D919-4835-A7D8-FC633EB7B7EC")));

auto b = L"m.pdf";

tux.emplace_back(make_tuple<wstring,

string>(forward<wstring>(b), string("A44BC1B3-D919-4835-A7D8-FC633EB7B7ED")));

tu.AddFiles(tux);

There is also AddSelf() to add your own executable automatically.

Uploading Updates to Your Server

The PHP script is communicated with using my lovely REST library, and the zip file containing the uploaded data is created using ZipUtils. Since this method requires the upload password and you don't want anyone to discover it by sniffing the connection, you normally won't be calling it from your app. Typically, an independent "uploader" will be used. All of the items are compressed into a ZIP file and sent to tu.php by the function. A sample "uploader" that uploads objects from an XML setup may be found in the github repository. The callback, which is optional, is called repeatedly while your files are uploading by the function. It must return E FAIL in order to halt the upload.

CHECKING FOR UPDATES

C++

HRESULT Check();

HRESULT hr = tu.Check();

Returns S_OK if all the checked files are up to date, or S_FALSE if any of the files need updating.

Updating

C++

HRESULT DownloadFull(std::function<HRESULT

(unsigned long long, unsigned long long, void*)>func = nullptr, void* lp = 0);

Automatically downloads all the files that need download and updates them. This function works also with files in use (such as your own application), by moving the current file into a .OLD one, creating a new target, then marking the OLD file for removal. This way, you can self-update easily with a few lines of code and your users won't ever notice it: The next time the application starts, the update will be there.

Hyperledger Fabric: Concepts, Configuration, and Deployment

Businesses and developers that wish to learn more about Hyperledger Fabric and its use cases will benefit from reading this article

INTRODUCTION

One of the most cutting-edge and distinctive blockchain platforms is Hyperledger Fabric. Using this platform, you can design private networks that are tailored to the requirements of your project or company.
We covered the network's capabilities, advantages, and disadvantages in this post. We set up and established a fundamental network for a practical purpose, and then we incorporated unique business logic. Then, we contrasted our solution with other, comparable implementations built on various platforms

Why is Hyperledger Fabric appropriate for commercial use?

Comparing Hyperledger Fabric to other blockchain networks like Bitcoin or Ethereum is challenging. Hyperledger is not designed for general use, unlike the majority of other networks.

In reality, the Hyperledger mainnet does not exist. Using the Hyperledger platform,

you can create networks that are unique to your requirements.
For a number of reasons, you might want to create a unique network for your project:

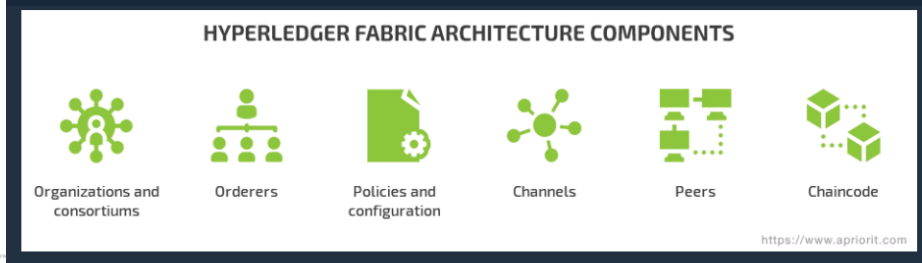
- You require a feature that isn't offered by any current network.
- Each of your jobs or clients is distinct and needs particular handling within the platform.
- You must protect your data if you want to keep some of it private.
- You desire access control for your network.

Possible applications of Hyperledger Fabric

What is Hyperledger Fabric?

A blockchain platform called Hyperledger Fabric is designed to build networks specifically for private businesses. Fabric is a Hyperledger project that was first created by Digital Asset and IBM and is now managed by the Linux Foundation.

Hyperledger Fabric: Network architecture components



Key features of the Hyperledger blockchain platform inc

- **Tamper-proof logs of every state transition**
- **Tools for using a Membership Service Provider to manage identities (MSP)**
- **Efficient transaction processing with no mining overhead**
- **Custom logic can be added using the chaincode functionality and called by particular transactions.**
- **Modular architecture to enable sophisticated network modification**

Basic network setup

```
# ---------------------------------------------------------------------------
# "OrdererOrgs" - Definition of organizations managing orderer nodes
# ---------------------------------------------------------------------------
OrdererOrgs:
  # ---------------------------------------------------------------------------
  # Orderer
  # ---------------------------------------------------------------------------
  - Name: Orderer
    Domain: example.com
    # ---------------------------------------------------------------------------
    # "Specs" - See PeerOrgs below for complete description
    # ---------------------------------------------------------------------------
    Specs:
      - Hostname: orderer
# ---------------------------------------------------------------------------
# "PeerOrgs" - Definition of organizations managing peer nodes
# ---------------------------------------------------------------------------
PeerOrgs:
  # ---------------------------------------------------------------------------
  # Org1: A simple organization
  # ---------------------------------------------------------------------------
  - Name: Org1
    Domain: org1.example.com
    EnableNodeOUs: true
    Template:
      Count: 2
    Users:
      Count: 1
  # ---------------------------------------------------------------------------
  # Org2: Another simple organization
  # ---------------------------------------------------------------------------
  - Name: Org2
    Domain: org2.example.com
    EnableNodeOUs: true
    Template:
      Count: 2
    Users:
      Count: 1
```

Advanced network configuration

- Let's revisit our example of network setting. There are two businesses that each operate an orderer. Everybody in a single channel has a peer. On this channel, just one chaincode is active.
- This set up is not at all representative of the real world. The number of entities involved in network governance, the complexity of access restrictions, and the likelihood of cross-channel conversations between businesses may all be significantly greater in real life.
- By including more businesses, creating new channels between them, and deploying different instances of the chaincode in the new channels, we can enhance the example network.
- Our network has also been made simpler by deploying Hyperledger Fabric on a single machine. Even while Docker is still used to deploy a real network, a

Chaincode implementation

```
// Init runs initialization for chaincode
func (t *DeliveryChaincode) Init(stub shim.ChaincodeStubInterface) peer.Response {
  caller, err := callerCN(stub)
  if err != nil {
    return shim.Error("Error getting caller cn")
  }
  ownerKey, err := getOwnerKey(stub)
  if err != nil {
    return shim.Error("Error getting database key")
  }
  err = stub.PutState(ownerKey, []byte(caller))
  if err != nil {
    return shim.Error("Error saving token data")
  }
  return shim.Success(nil)
}

type resolveDispute struct {
  DeliveryID uint64 `json:"delivery"`
  Refund     bool   `json:"refund"`
}
func (t *DeliveryChaincode) resolveDispute(stub shim.ChaincodeStubInterface, args []string)
peer.Response {
  params := resolveDispute{}
  caller, owner, err := getCallParams(stub, args, params)
  if err != nil {
    return shim.Error(err.Error())
  }
  delivery, _ := t.getDelivery(stub, params.DeliveryID)
  // Is delivery available
  if delivery.Status != dispute {
    return shim.Error("Delivery not available.")
  }
  // Check the customer
  if owner != caller {
    return shim.Error("Only owner can resolve a dispute.")
  }
  delivery.Status = complete
  if params.Refund {
    // Pay the customer
    keyCustomer, _ := stub.CreateCompositeKey(BalanceKey, []string{delivery.Customer})
    balance, _ := t.getValue(stub, keyCustomer)
    t.setValue(stub, keyCustomer, balance+delivery.Payment)
  } else {
    // Pay the courier
    keyCourier, _ := stub.CreateCompositeKey(BalanceKey, []string{delivery.Courier})
    balance, _ := t.getValue(stub, keyCourier)
    t.setValue(stub, keyCourier, balance+delivery.Payment)
  }
  t.setDelivery(stub, delivery)
  result, _ := json.Marshal(delivery)
  return shim.Success(result)
}
```

Implementation Comparision

Of course, Hyperledger Fabric isn't the only network that allows for the creation of custom on-chain business logic. Let's compare our sample implementation to similar solutions based on different networks.
While comparing, we'll mostly take into consideration the following factors:

- Effectiveness (based on performance and scalability)
- Development costs
- Ease of use
- Security

Conclusion

Compared to other well-known blockchains, Hyperledger offers distinctive capabilities. The best option for constructing a specialised blockchain network is definitely Hyperledger Fabric due to its permissioned nature, excellent access control, and scalability.

THE EASY UPDATE LIBRARY FOR YOUR PROJECT

INTRODUCTION:

Applications that do not automatically update irritate me. Really. This library will perform every task for you automatically. All you need to save the updates is a web server that can execute PHP.

FEATURES:

- Full or differential updates
- Silent update
- Compression per item
- Detached deatabases

WINDOWS APPLICATION

- Include tu.hpp
- Instantiate a TU::TU object:

C++

```
TU(const char* prjg,const wchar_t* host,const wchar_t* path,
bool SSL = false,unsigned short Port = 0,DWORD flg = 0,const wchar_t* un = 0,
const wchar_t* pwd = 0,const wchar_t* uploadpwd = 0)
```

PARAMETERS:

- The unique project GUID created by tu.php admin panel
- The hostname
- The path to the PHP script
- true/false for SSL
- Optional port (if 0, 80 or 443 are used)
- Flags for InternetConnect() function
- Optional username/password for your server
- The upload password if you plan to upload files

C++

```
HRESULT Upload
    (std::function<HRESULT(size_t sent, size_t total, void*)> func = nullptr,void* lp = 0);
HRESULT hr = tu.Upload();
```

C++

```
vector<tuple<wstring, string>> tux;
auto a = L"m.docx";
tux.emplace_back(make_tuple<wstring,
string>(forward<wstring>(a), string("A44BC1B3-D919-4835-A7D8-FC633EB7B7EC")));
auto b = L"m.pdf";
tux.emplace_back(make_tuple<wstring,
string>(forward<wstring>(b), string("A44BC1B3-D919-4835-A7D8-FC633EB7B7ED")));
tu.AddFiles(tux);
```

There is also AddSelf() to add your own executable automatically.

Uploading Updates to Your Server

The PHP script is communicated with using my lovely REST library, and the zip file containing the uploaded data is created using ZipUtils. Since this method requires the upload password and you don't want anyone to discover it by sniffing the connection, you normally won't be calling it from your app. Typically, an independent "uploader" will be used. All of the items are compressed into a ZIP file and sent to tu.php by the function.A sample "uploader" that uploads objects from an XML setup may be found in the github repository.

The callback, which is optional, is called repeatedly while your files are uploading by the function. It must return E FAIL in order to halt the upload.

CHECKING FOR UPDATES

C++

```
HRESULT Check();
HRESULT hr = tu.Check();
```

Returns S_OK if all the checked files are up to date, or S_FALSE if any of the files need updating.

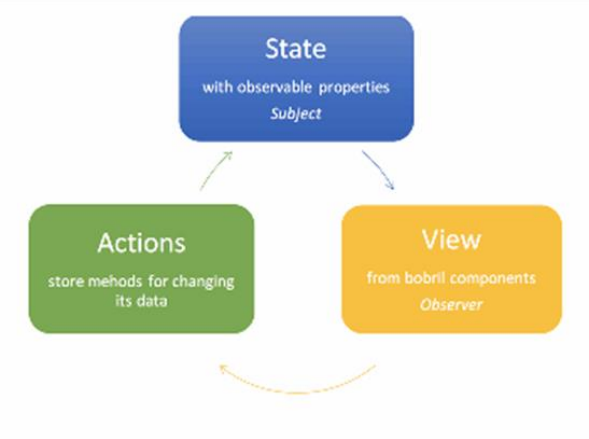
Updating

C++

```
HRESULT DownloadFull(std::function<HRESULT
(unsigned long long, unsigned long long, void*)> func = nullptr, void* lp = 0);
```

Automatically downloads all the files that need download and updates them. This function works also with files in use (such as your own application), by moving the current file into a .OLD one, creating a new target, then marking the OLD file for removal. This way, you can self-

Bobril - VI - BobX Application Store Management



INTRODUCTION:

In the second post, we used the bobflux framework to build a straightforward to-do application. Using the new framework BobX, we will learn how to develop such an application more quickly and easily in this post.

- Bobril - I - Getting Started
- Bobril - II - Bobflux Application Architecture
- Bobril - III - Localizations and Formatting
- Bobril - IV - Routing
- Bobril - V - Bobril-build
- Bobril - VI - BobX Application Store Management
- Bobril - VII - Components and TSX

A store management system executes store operations, utilizing a correspondence mechanical assembly In the store, associated with no less than one store terminal by method for a first line Inside the store and associated with a server by a second line outside the store.

Background

Boris Letocha designed BobX, a library for managing application stores that is similar to MobX. (Quadient). It is TypeScript-written and meets the requirements of the Bobril application. Stores act as observable subjects and bobril components function as observers in this observer architecture.

Store Optimalizations

There are more ways to define observable attributes besides the pure observable function. Let's look at some alternate approaches as you may not always want to keep track of every property an object has:

- observable.deep - The default observable way. It decorates all defined properties of given objects to be observable (tracked) recursively. The recursion stops when the property contains an object with defined prototype.
- observable.ref - Only the reference of object is tracked. No change of inner properties will trigger rendering.
- observable.shallow - This variant will track the reference of given object, its properties but nothing more. So for example, the array will be tracked for its reference, its content, but not for the content of its items.
- observable.map - You can use this function to create a dynamic keyed observable map.

Let's begin

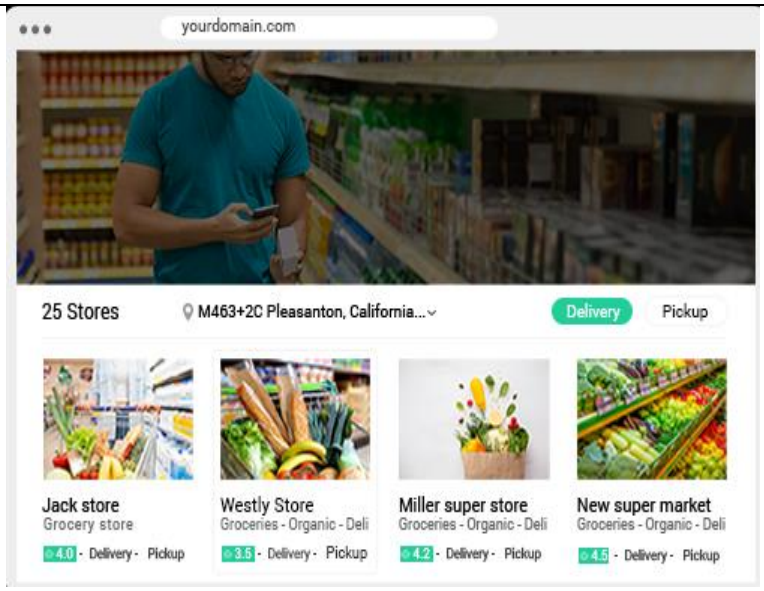
Once more, we'll make a straightforward TODO application. We must first have a computer with a prepared bobril-build. For installation of bobril-build, adhere to the instructions in the first article. You can now restart a new project or utilise a simpleApp skeleton that has already been created in the bobril-build github repository. It is used in the example that follows. Download the entire sample if you want the finished code with all the required parts.

BobX makes use of TypeScript's experimental decorators feature. Add the following parameter to the package's bobril/compilerOptions section to enable the use of decorators. Json:

Javascript

```
"bobril": {
    "compilerOptions": {
        "experimentalDecorators": true
    }
}
Add bobx to Application
BAT
npm i
npm i bobx --save
bb
Store
import { observable } from 'bobx';
class TodoStore {
    @observable todoName: string = '';
    @observable private _todos: string[] =
[];
    get todos(): string[] {
        return this._todos;
    }
    addTodo(): void {
        if (this.todoName.trim().length ===
0)
            return;

this._todos.push(this.todoName.trim());
        this.todoName = ''; }}
export const todoStore = new
TodoStore();
Composing the Page with BobX
import * as b from 'bobril';
import { button } from
'./components/button';
import { textbox } from
'./components/textbox';
import { p } from
'./components/paragraph';
import { h1 } from
'./components/header';
import { todoStore } from './store';
export const mainPage =
b.createComponent({
    render(_ctx: b.IBobrilCtx, me:
b.IBobrilNode): void {
```



Key Results of Store Management System

- Easy Identification of all material stored.
- Receipt of incoming goods.
- Inspection of all receipts.
- Storage and preservation.
- Easy to Handle Materials.
- Issue by FIFO.
- Maintenance of stock records.
- Stores accounting.
- Controlled Inventory.
- Easy for Stock-taking

CONCLUSION:

A Store Management System software is extremely critical to ensure smooth running of your retail business. Store management systems should be adaptable and secure to convey the right client service. These systems must be driven by access to real-time client data. At exactly that point can retailers convey customized offers and services that are up to the

```
        me.children = [
            h1({}, 'TODO'),
            p({}, [
                textbox({ value:
todoStore.todoName,
                          onChange: newValue =>
todoStore.todoName = newValue }),
                button({ title: 'ADD', onClick: ()
=> todoStore.addTodo() })
            ]),
            todoStore.todos.map(item =>
p({}, item)),
            p({}, `Count:
${todoStore.todos.length}`)
        ];
}});
```

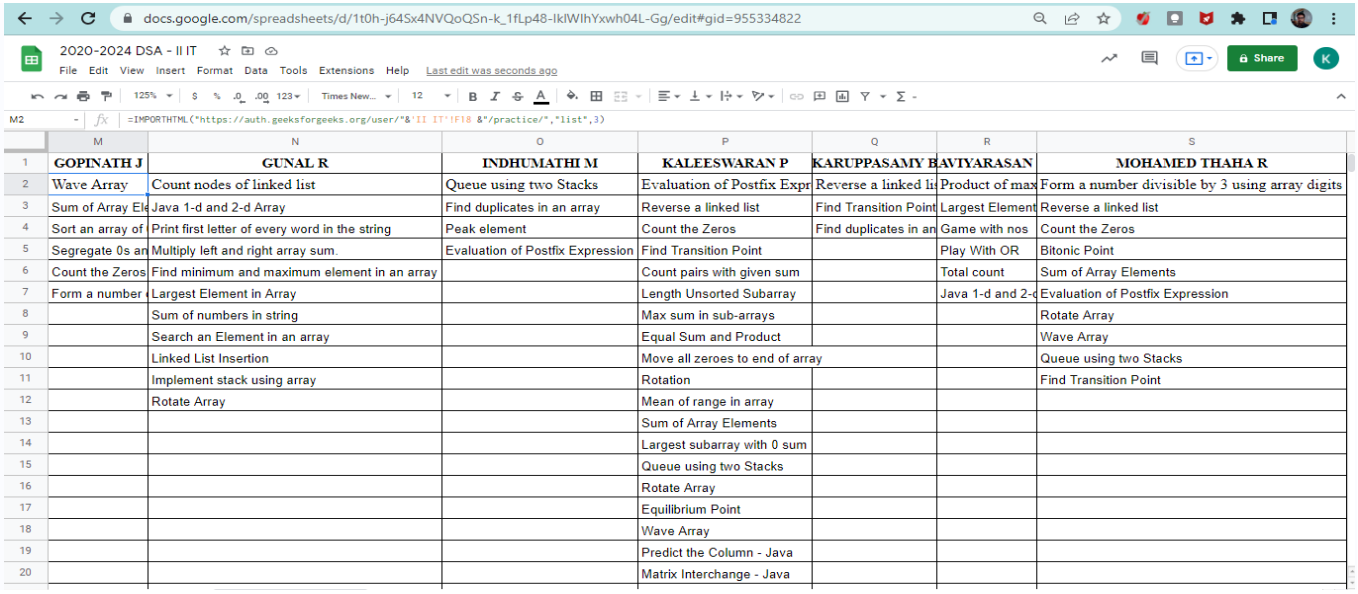
moment and targeted to each individual.

Mr-Sajeesh.K (II CSE)

PROJECT GLIMPSES

PROGRAMMERS DASHBOARD:

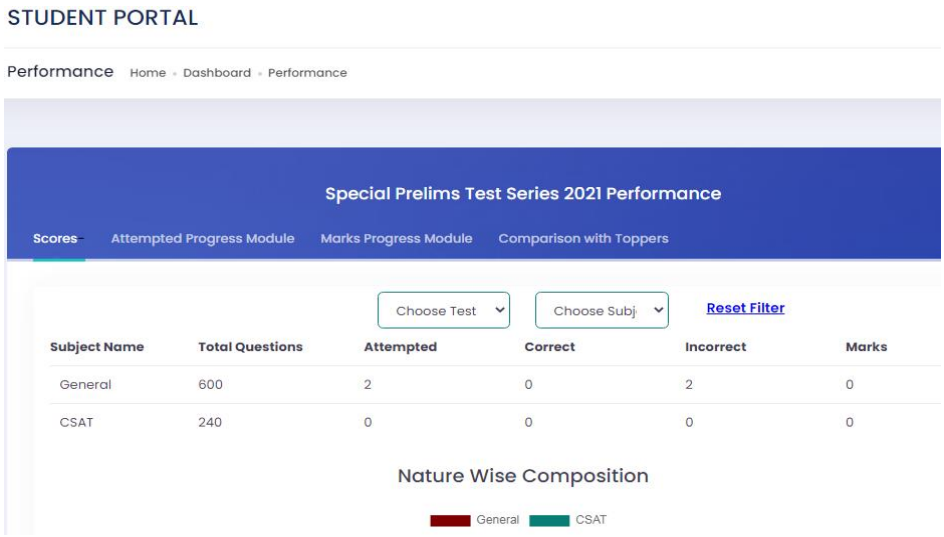
Client Profile Management System is mainly developed for a wholesale suppliers who is supplying the spareparts and accessories in the city. The system maintains the complete details of machine owned by the clients. Also it maintain the stock details of machine and locations of the clients for better location tracking.



	M	N	O	P	Q	R	S
1	GOPINATH J	GUNAL R	INDHUMATHI M	KALEESWARAN P	KARUPPASAMY B	AVIYARASAN	MOHAMED THAHA R
2	Wave Array	Count nodes of linked list	Queue using two Stacks	Evaluation of Postfix Expr	Reverse a linked li	Product of max	Form a number divisible by 3 using array digits
3	Sum of Array El	Java 1-d and 2-d Array	Find duplicates in an array	Reverse a linked list	Find Transition Point	Largest Element	Reverse a linked list
4	Sort an array of	Print first letter of every word in the string	Peak element	Count the Zeros	Find duplicates in an	Game with nos	Count the Zeros
5	Segregate 0s an	Multiply left and right array sum.	Evaluation of Postfix Expression	Find Transition Point		Play With OR	Bitonic Point
6	Count the Zeros	Find minimum and maximum element in an array		Count pairs with given sum		Total count	Sum of Array Elements
7	Form a number	Largest Element in Array		Length Unsorted Subarray		Java 1-d and 2-d	Evaluation of Postfix Expression
8		Sum of numbers in string		Max sum in sub-arrays			Rotate Array
9		Search an Element in an array		Equal Sum and Product			Wave Array
10		Linked List Insertion		Move all zeroes to end of array			Queue using two Stacks
11		Implement stack using array		Rotation			Find Transition Point
12		Rotate Array		Mean of range in array			
13				Sum of Array Elements			
14				Largest subarray with 0 sum			
15				Queue using two Stacks			
16				Rotate Array			
17				Equilibrium Point			
18				Wave Array			
19				Predict the Column - Java			
20				Matrix Interchange - Java			

PROGRAMMERS DASHBOARD

Programmers dashboard is intended to display studetns programming ability in “GEEKWEEK” platform. It works well based on the web scraping. It also provides additional Inoformation about the number of programs solved in different levels.



Test your Knowledge solutions

LEARN TECHNOLOGY CROSSWORD

Here are the word from crossword:

- calculator
- delete
- download
- keyboard
- headline
- install
- internet
- laptop
- password
- photograph
- postcard
- upload
- webcam

GOLD BAR PUZZLE

Gold Bar Puzzle

Just 2cuts

First one between 1 and 2 and second between 3 and 4

Let’s number them from 1 to 7

So you have 1 2-3 4-5-6-7 in this information and connection

Day 1 – Give 1

Day 2 – Give 2-3 take back 1

Day 3 – Give 1 also

Day 4 – Take all back and give 4-5-6-7

Day 5- Give 1 with the rest

Day 6 – Take back 1 and give 2-3

Day – Give 1

So we need only two cuts.

25HORSES AND 5TRACK PUZZLE:

Total 7races are required

By using Work through process of elimination

This means we only have 5 horses left! Now we race those horses one more time – in the seventh (7th) race – and we can take out the top 2 horses and that would mean we have the 2nd and 3rd place horses! So, we have found our answer! It takes 7 races to find the top 3 horses in this problem.

EDITORIAL MEMBERS

Editorial Faculty Member

Mr.P.N.Ramesh

Editorial Student Members

Mr.Rohithkumar.S (IV-CSE)
Ms.Pooja Shree.P.K (IV-CSE)
Mr.Joel Johnson.S (III-CSE)
Ms.Gayathre.B (III-CSE)
Mr.Vikash.V (II-CSE)