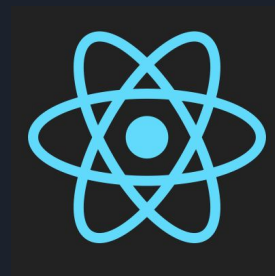
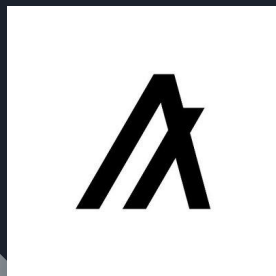


Pipeline UI

Algorand + React



From Headline INC

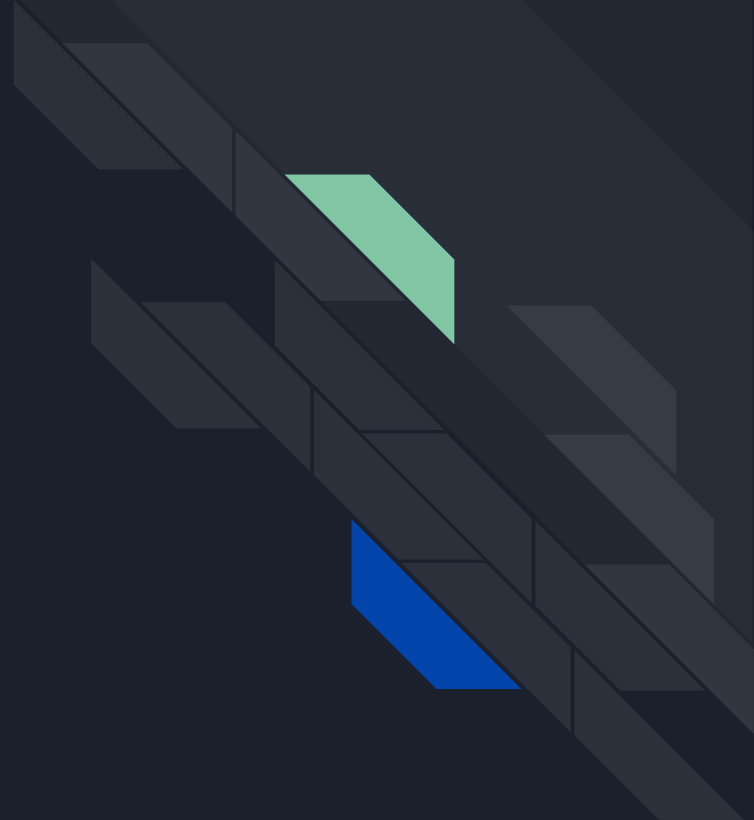


Obstacle:

Steep on-ramp to use
blockchain technology in
web apps

A Solution:

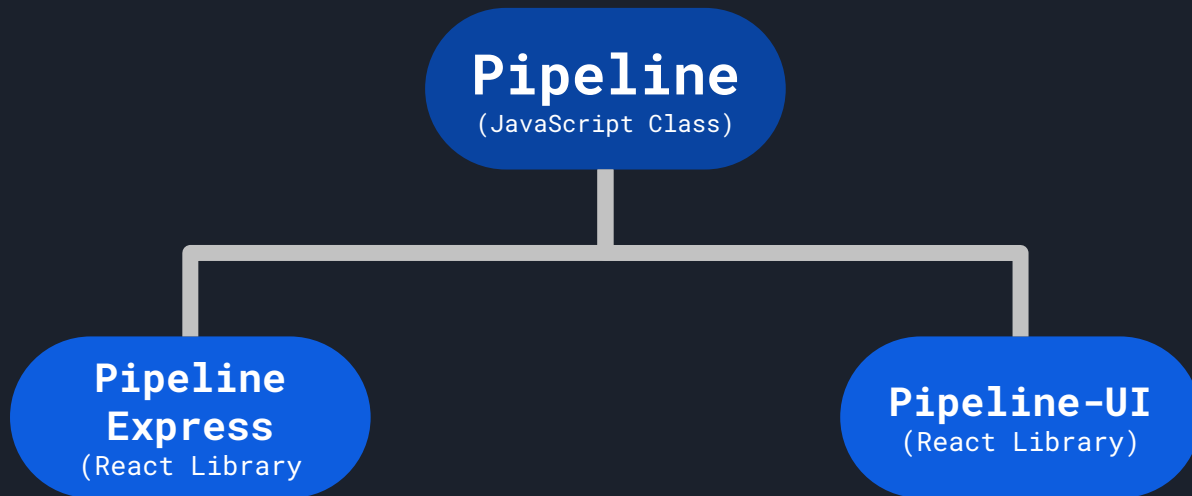
Hard-code blockchain functions in
React components!





What is Pipeline UI?

- React libraries derived from a common JavaScript class
- Simplifies the process of enabling websites to incorporate Algorand transaction capabilities.



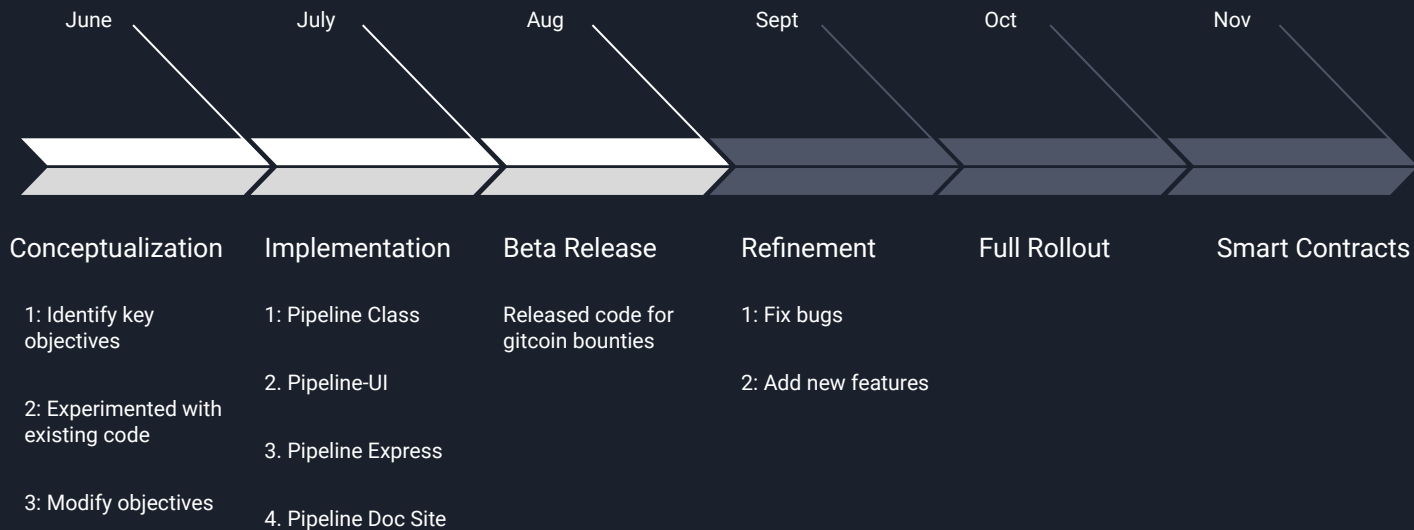
Target Users

- Small businesses who may be unfamiliar with React, JavaScript and/or Algorand
- Experienced developers who wish to rapidly build new web applications with Algorand functionality
- Enterprise innovators who wish to rapidly deploy demos






Project Timeline: Milestone 1





Completed Objectives:

- Created a JavaScript class to simplify Algorand payments, ASA transfers, balance checks and transaction fetching
 - Published two libraries of standard UI and Algorand-specific React components
 - Compiled an interactive website with live code examples to document library features
 - Created and deployed fully-functional demos that utilize Pipeline UI libraries
- 



Pipeline Class: *A Micro Algorand SDK*

- Foundational building block of all Pipeline UI projects
- Functions largely hidden in UI & Express, but still accessible
- Standalone usability without UI or Express (for JavaScript or React apps).



Pipeline Class: Motivations

- Dependency reduction
- Code size reduction
 - No need for signing functions, as MyAlgo Connect is used for signing
- Increased compatibility and ease of use



Pipeline Class: Overview

- 1 dependency (@randlabs/myalgo-connect)
- Vanilla JavaScript
- A micro SDK (does not use algosdk)
- Supports Algorand functions:
 - TestNet & MainNet
 - Connect MyAlgo Wallet
 - Check balances
 - Send payments
 - Send ASA transfers
 - Get transaction ID's

```
13 static async balance(address) {  
14  
15     let indexerURL = 'https://'  
16  
17     if (this.main == true) {  
18         indexerURL = indexerURL + 'algoexplorerapi.io/idx2/v2/accounts/'  
19     }  
20     else {  
21         indexerURL = indexerURL + "testnet.algoexplorerapi.io/idx2/v2/accounts/"  
22     }  
23  
24     let url2 = indexerURL + address  
25     try {  
26         let data = await fetch(url2)  
27         let data2 = await data.json()  
28         let data3 = JSON.stringify(data2.account.amount / 1000000) + ' Algos'  
29         this.myBalance = data3;  
30         return data3  
31     } catch (error) {  
32         console.log(error)  
33     }  
34 }  
--
```



Pipeline Class: Primary Methods

balance

Sends request to
AlgoExplorer (returns
balance)

connect

Connects MyAlgo wallet
(returns first address)

send

Sends an Algorand or ASA
transaction (returns txID)

```
34     fetchBalance = () => {  
35         Pipeline.balance(this.state.myAddress).then(  
36             data => {  
37                 this.setState({balance: data});  
38             }  
39         )  
40     }
```



Pipeline Class: Switching Nets

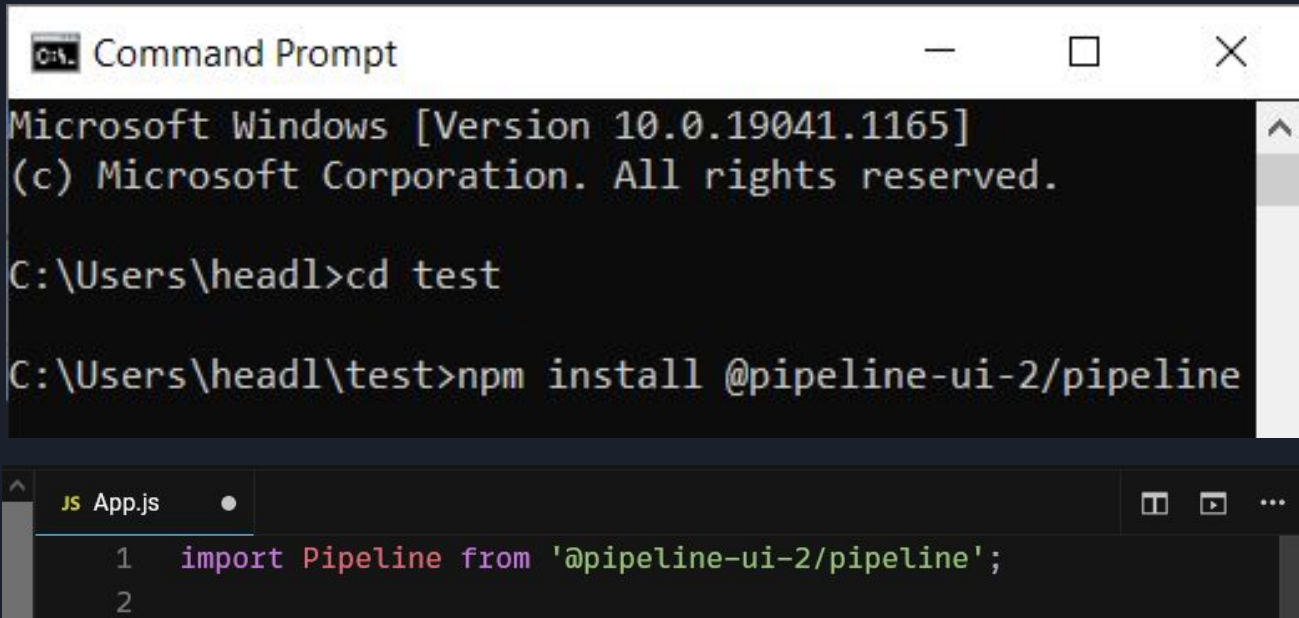
`Pipeline.main = false;`

Switch to TestNet

`Pipeline.main = true;`

Switch to MainNet

Pipeline Class: Installing and Importing



The image shows a Windows Command Prompt window and a VS Code editor. The Command Prompt window is titled "Command Prompt" and shows the following text:

```
Microsoft Windows [Version 10.0.19041.1165]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\headl>cd test  
  
C:\Users\headl\test>npm install @pipeline-ui-2/pipeline
```

The VS Code editor shows a file named "App.js" with the following code:

```
1 import Pipeline from '@pipeline-ui-2/pipeline';  
2
```



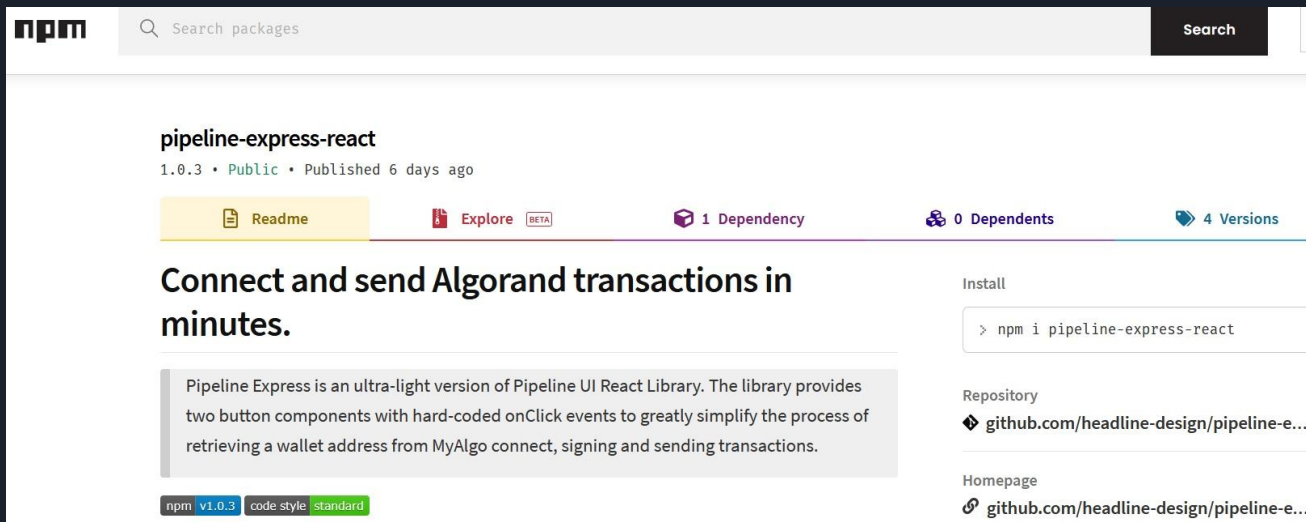
Pipeline Class: Usage Example

React *onClick*
handler for a
“send” button

```
50     handleSend = () => {  
51         Pipeline.send(  
52             recipient,  
53             amount,  
54             note,  
55             undefined,  
56             myAlgoWallet,  
57             index)  
58         .then(  
59             data => {  
60                 this.setState({ txID: data });  
61             }  
62         );  
63     }
```

Pipeline Express: Overview

- Ultra-light (sole dependency @randlabs/myalgo-connect)
- 2 React components with built-in onClick handlers
- Access to Pipeline Class



The screenshot shows the npm package page for **pipeline-express-react**. The package is version 1.0.3, is public, and was published 6 days ago. It has 1 dependency, 0 dependents, and 4 versions. The main description states: "Connect and send Algorand transactions in minutes." A detailed description explains that Pipeline Express is an ultra-light version of the Pipeline UI React Library, providing two button components with hard-coded onClick events to simplify retrieving a wallet address from MyAlgo connect, signing, and sending transactions. The install command is shown as `> npm i pipeline-express-react`. The repository is listed as github.com/headline-design/pipeline-express-react. The homepage is also listed as github.com/headline-design/pipeline-express-react. The package is licensed under the MIT license, and the code style is standard.

pipeline-express-react
1.0.3 • Public • Published 6 days ago

[Readme](#) [Explore](#) [BETA](#) [1 Dependency](#) [0 Dependents](#) [4 Versions](#)

Connect and send Algorand transactions in minutes.

Pipeline Express is an ultra-light version of Pipeline UI React Library. The library provides two button components with hard-coded onClick events to greatly simplify the process of retrieving a wallet address from MyAlgo connect, signing and sending transactions.

`npm` `v1.0.3` `code style` `standard`

Install

```
> npm i pipeline-express-react
```

Repository

github.com/headline-design/pipeline-express-react

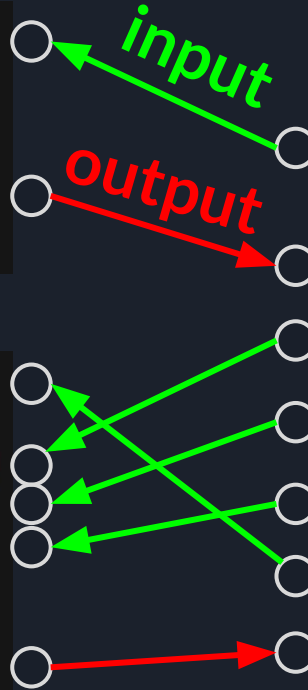
Homepage

github.com/headline-design/pipeline-express-react

Pipeline Express

```
40      <AlgoButton
41        wallet={myAlgoWallet}
42        context={this}
43        returnTo={"myAddress"}
44      />
```

```
68      <AlgoSendButton
69        index={0}
70        recipient={this.state.recipient}
71        amount={this.state.amount}
72        note={this.state.note}
73        wallet={myAlgoWallet}
74        context={this}
75        returnTo="txID"
76      />
```



The image shows a web form titled "Pipeline EXPRESS" with a rocket icon. The form contains the following elements:

- A button labeled "Connect to MyAlgo".
- A checkbox labeled "MainNet:" which is checked.
- A text input field labeled "My Address:".
- A text input field labeled "Recipient:".
- A text input field labeled "Amount:" with a dropdown arrow on the right.
- A text input field labeled "Note:".
- A large blue button labeled "Send".
- A text input field labeled "Transaction ID:".

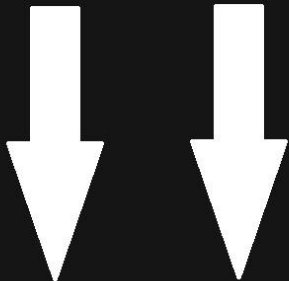
context and *returnTo* props

```
12 handleChange = (value) => {  
13   this.setState(  
14     { myAddress: value }  
15   )  
16 }
```



- No need for callback functions
- Immediate re-rendering when used with class-based React apps
- *onChange* prop also provided for functional components

```
1  import React, { Component } from 'react'  
2  import { AlgoButton, Pipeline } from 'pipeline-express-react'  
3  
4  const myAlgoWallet = Pipeline.init();  
5  
6  class App extends Component {  
7  
8    constructor(props) {  
9      super(props);  
10     this.state = {  
11       myAddress: "",  
12     }  
13   }  
14  
15   render() {  
16     return(  
17       <div>  
18         <AlgoButton wallet={myAlgoWallet} context={this} returnTo={"myAddress"} />  
19         <h3>{"My Address: " + this.state.myAddress}</h3>  
20       </div>  
21     )  
22   }  
23 }  
24  
25 export default App;
```





Pipeline UI: Overview

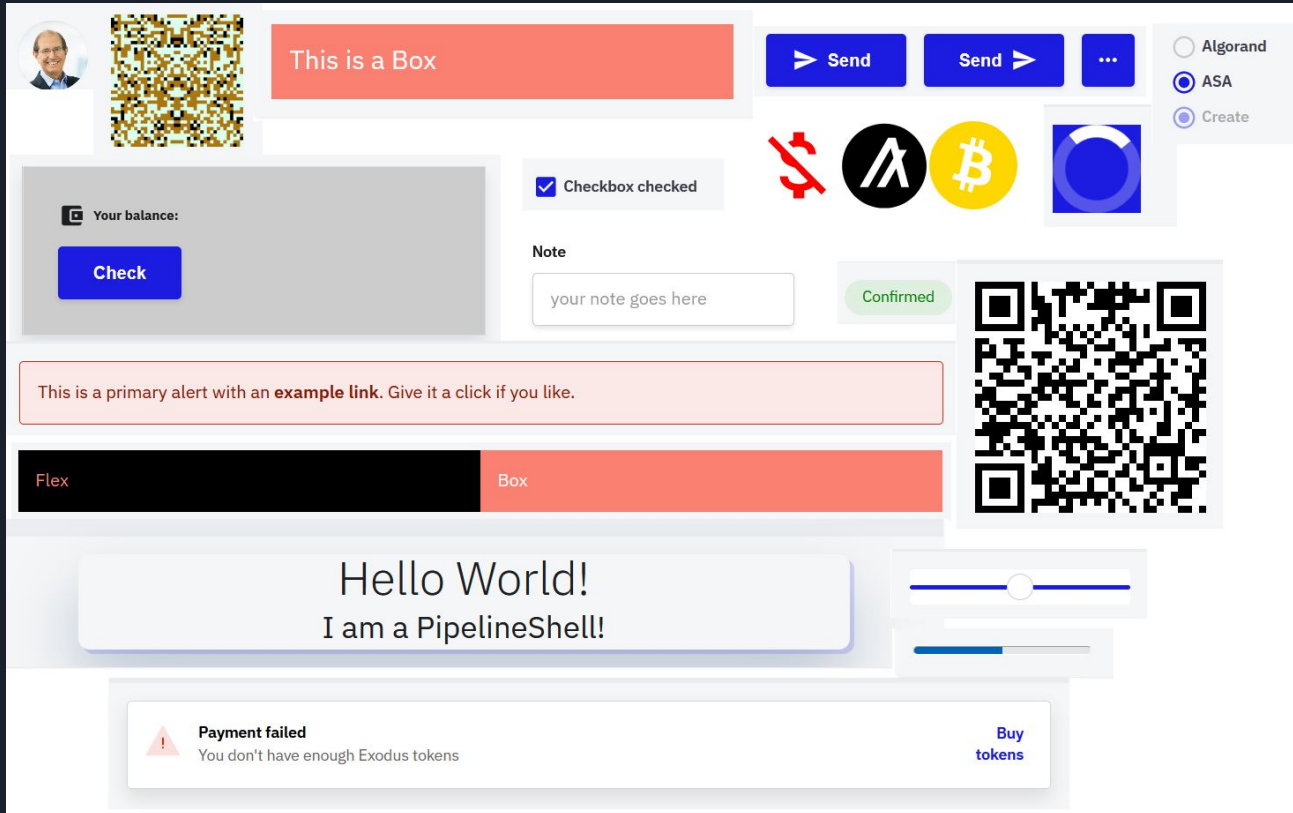
- 28 styled standard UI components
- Includes Pipeline class
- 6 additional Algorand components
 - **AlgoAddress** (QR code with copy to clipboard)
 - **AlgoButton** (connects to MyAlgo wallet)
 - **AlgoFetch** (refreshable table of 5 latest txID's & links)
 - **AlgoSendButton** (sends payments and ASA's)
 - **AsaList** (array of verified ASA's and index numbers)
 - **SwitchNet** (one-click, single JSX tag switch between test and main nets)



Standard UI Components

- Avatar
- Blockie
- Box
- Button
- Card
- Checkbox
- Field
- Flash
- Flex
- Heading
- Icon
- Image
- Input
- Link
- Loader
- Modal
- Pill
- PipelineShell
- Progress
- QR
- Radio
- Select
- Slider
- Table
- Text
- Textarea
- ToastMessage
- BaseStyles
- ThemeProvider
- theme


Standard UI Components: Preview



Pipeline UI: Algorand Components

Algorand Address

To send funds to this Algorand address, scan this code using your mobile wallet app



LMKFQIPL3VQCZGGFK4WZ7FPCQWLNBTJQ3UWSTA7D7QZS ...

Copy



☒ MainNet

MY LATEST TRANSACTIONS

YKGSUNCQZX7UBF2FUKGCQVCGJUHVGZG5ZEG2QLEUYU66UDMKRVNA

OJGVBSL6I3L322HURFPBSV4XTWIJMRIQTJ7HP7JZTIAY5MHG5RLQ

FCBFVZ7RTEAVSNOS5GGUAKZUOICVKPSUS6KYRWURX6WC6AJOJJLA

QABVHMOSFVGEWLKPGNMC63GZLIBSMWIJWBGLRXZARH2L5YXVNPFFQ

YHSZN2BAOZOP4CWYQQVTDEQXROGG4IIQ2XTP2ADKTOC57ZB5MD5A

Refresh

Select a Verified ASA:

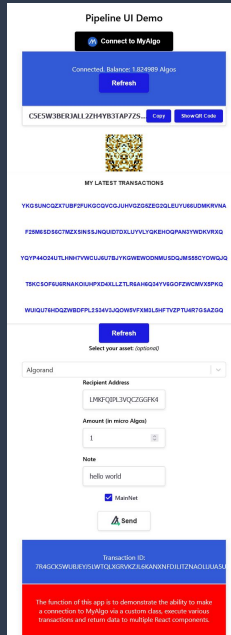
Select...

ARCC

OBA

USDt

Demos by headline-design

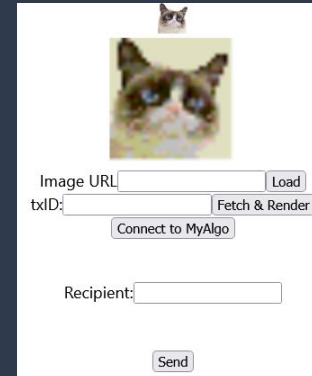


Test

Original example
used to create and
test all Algorand
components

AlgoPost


Store tiny images in
Algorand "note" field







Demos by other developers



Restricted Algorand Checkout

By John Grimm

 **Algorand Demo Store**



Algorand test item 1
Some description of the item that is being sold on this App
5  [Buy Now!](#)


Algorand test item 2
Some description of the item that is being sold on this App
11  [Buy Now!](#)


Algorand test item 3
Some description of the item that is being sold on this App
30  [Buy Now!](#)


Algorand Donation App

By DragonDev1906

Donate 

Additional Message (optional)

Amount
☒ 1 ALGO
☐ 5 ALGO
☐ 10 ALGO
☐ 50 ALGO

 Send

Thank you for your Donation. [Show in Explorer](#)



Pipeline-UI Building Blocks

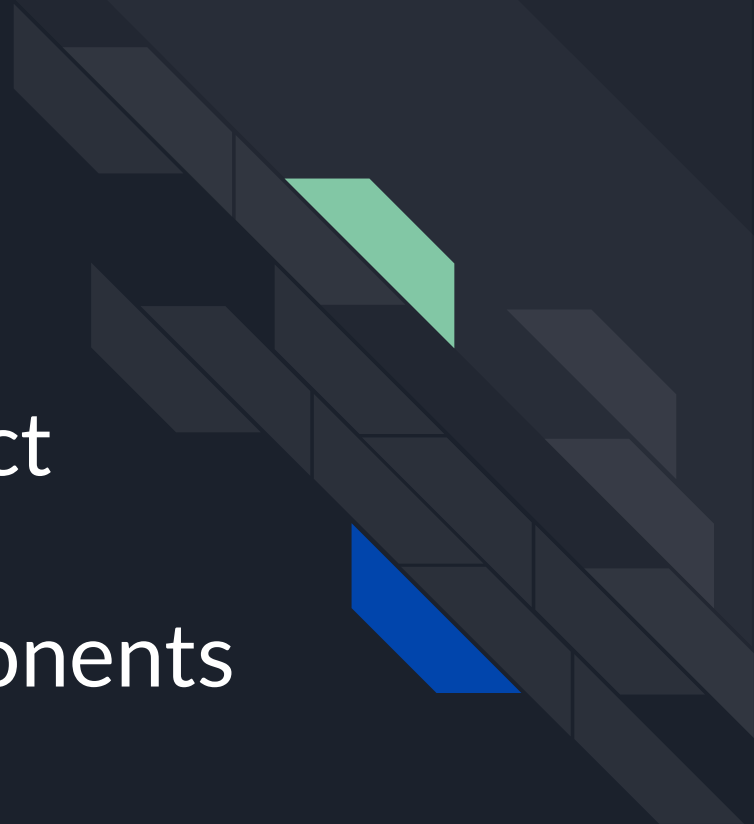
- Styled-Components
- React-Select
- Bootstrap
- Material Icons
- Material Design
- Carbon Design
- Rimple-UI
- Rollup
- MDX
- JSX
- Docusaurus
- React-Live
- Prisma



Looking Ahead:

Milestone 2 Integration

- TEAL
- Reach
- AlgoSigner
- WalletConnect
- Apex Charts
- Voting Components





Links

[Headline INC](#) (Headline company website)

[Pipeline UI](#) (Documentation site for all Pipeline projects)

[Test](#) (Pipeline-UI functional Algorand Demo)

[AlgoPost](#) (Pipeline Class demo to store tiny images on Algorand blockchain)

[Restricted Algorand Checkout](#) (demo by John Grimm)

[Algorand Donation App](#) (demo by DragonDev1906)