

オンラインイベント

JavaScript ベースのローコードライブラリ **Webix** を用いて優れたユーザインタフェースを構築する方法

株式会社ライジングサン・システムコンサルティング
岩佐 和紀

セッション ID : T-04



限界のないローコード。
スキルを磨けば、ビジネスが輝く。

 Claris Engage Japan 2022

#ClarisEngageJP



株式会社ライジングサン・システムコンサルティング

代表取締役社長

岩佐 和紀

@kazukiwasa



ITエンジニアとしての経歴

01

スタート

敷島製パン株式会社
入社。情報システム
部門に配属。

富士通メインフレー
ムのCOBOLプログラ
マ・社内SEとして
キャリアをスタート

6年

02

SES

地方のソフトウェア
開発会社に転職。

某国営企業の民営化
PJに、多重下請けの
最下位レイヤープログ
ラマとして超巨大プ
ロジェクトに派遣(ド
ナドナ)される

1年

03

外資系

SGS Far East



BMWジャパン

ファイルナス



DOLCE&GABBANA
ジャパン

...と3社の外資系企
業を、社内SEとして
転々とする

5年

04

ベンチャー

株式会社ベストブライ
ダル（現ツカダ・グ
ローバルホールディン
グ）に社内SEとして
入社。

基幹システムのPJ
リーダーとしてマザー
ズ上場・東証1部指
定替えなどを経験

2年

05

独立

2006年、娘の誕生を
期に株式会社ライジ
ングサン・システム
コンサルティングを
設立

2010年から本格的に
FileMakerプラット
フォームを活用した
事業をスタート

16年

CodeZineに連載した技術系記事



<https://codezine.jp/author/2017>

01

DBエンジン性能

実際に「どこまでのシステム」が構築可能？
FileMaker Serverの強力なDBエンジン性能にフォーカス

02

外部システム連携

FileMakerプラットフォームと外部システムを連携する3つの手法

03

JavaScript連携

FileMakerとJavaScriptのマッシュアップでリッチなユーザインタフェースを実装しよう

04

ワンソース マルチデバイス

FileMakerがもつワンソース・マルチデバイスの強み

05

実行環境

FileMakerプラットフォームの豊富な実行環境

CodeZine人気記事ランキングで1位

ローコードはITプロフェッショナルを幸せにするのか？

FileMakerエンジニアに学ぶ、ローコード開発の本質



<https://codezine.jp/article/detail/14843>



事業内容

01

内製化支援

Claris FileMaker を用いた、業務システムの内製化を支援

2016年よりサービス開始

Zoomなどで、画面共有をしながらペアプログラミングで開発支援をするライトなサービスから、プロジェクト期間1年、基幹システムの構築と社内開発者の育成まで、様々な規模の事例がある。

02

スキル移行支援

Microsoft .net や Java、その他の一般汎用言語で開発してきたITエンジニアさんを、FileMakerプラットフォームの開発者としてスキルコンバートするプロセスを支援する

全国200店舗以上で運営するデイサービス事業の店舗運営システム等の構築を支援した実績がある

03

プロトタイプ開発

企画段階のWebサービスやパッケージシステムのプロトタイプを、FileMakerの高い生産性を活かして素早く開発することで、新規事業の立ち上げをサポートするサービス。

将来的にWeb系を始めとした、他の開発言語での焼き直しが前提だが、そのままFileMakerでサービスインするケースも有る。

事業内容

01

内製化支援

Claris FileMaker を用いた、業務システムの内製化を支援

2016年よりサービス開始

Zoomなどで、画面共有をしながらペアプログラミングで開発支援をするライトなサービスから、プロジェクト期間1年、基幹システムの構築と社内開発者の育成まで、様々な規模の事例がある。

02

スキル移行支援

Microsoft .net や Java、その他の一般汎用言語で開発してきたITエンジニアさんを、FileMakerプラットフォームの開発者としてスキルコンバートするプロセスを支援する

全国200店舗以上で運営するデイサービス事業の店舗運営システム等の構築を支援した実績がある

03

プロトタイプ開発

企画段階のWebサービスやパッケージシステムのプロトタイプを、FileMakerの高い生産性を活かして素早く開発することで、新規事業の立ち上げをサポートするサービス。

将来的にWeb系を始めとした、他の開発言語での焼き直しが前提だが、そのままFileMakerでサービスインするケースも有る。

事業内容

01

内製化支援

Clarix FileMaker を用いた、業務システムの内製化を支援

2016年よりサービス開始

Zoomなどで、画面共有をしながらペアプログラミングで開発支援をするライトなサービスから、プロジェクト期間1年、基幹システムの構築と社内開発者の育成まで、様々な規模の事例がある。

02

スキル移行支援

Microsoft .net や Java、その他の一般汎用言語で開発してきたITエンジニアさんを、FileMakerプラットフォームの開発者としてスキルコンバートするプロセスを支援する

全国200店舗以上で運営するサービス事業の店舗運営システム等の構築を支援した実績がある

03

プロトタイプ開発

企画段階のWebサービスやパッケージシステムのプロトタイプを、FileMakerの高い生産性を活かして素早く開発することで、新規事業の立ち上げをサポートするサービス。

将来的にWeb系を始めとした、他の開発言語での焼き直しが前提だが、そのままFileMakerでサービスインするケースも有る。

事業内容

01

内製化支援

Claris FileMaker を用いた、業務システムの内製化を支援

2016年よりサービス開始

Zoomなどで、画面共有をしながらペアプログラミングで開発支援をするライトなサービスから、プロジェクト期間1年、基幹システムの構築と社内開発者の育成まで、様々な規模の事例がある。

02

スキル移行支援

Microsoft .net や Java、その他の一般汎用言語で開発してきたITエンジニアさんを、FileMakerプラットフォームの開発者としてスキルコンバートするプロセスを支援する

全国200店舗以上で運営するデイサービス事業の店舗運営システム等の構築を支援した実績がある

03

プロトタイプ開発

企画段階のWebサービスやパッケージシステムのプロトタイプを、FileMakerの高い生産性を活かして素早く開発することで、新規事業の立ち上げをサポートするサービス。

将来的にWeb系を始めとした、他の開発言語での焼き直しが前提だが、そのままFileMakerでサービスインするケースも有る。

A close-up, slightly blurred image of a MacBook Pro keyboard and the bottom of the screen. The text "MacBook Pro" is visible on the bezel below the screen.

メインテーマ

JavaScript ベースの ローコードライブラリ **Webix** を用いて 優れたユーザインタフェースを 構築する方法

Webixを選択する 3つの理由

01

豊富なドキュメント とサンプルコード集

丁寧に作り込まれたドキュメント、そしてコードスニペット等を含めたサンプルコードが豊富。

特に Webix に特化した CodePenのようなコードスニペットは強力!

ドキュメント：

<https://docs.webix.com/>

コードスニペット：

<https://snippet.webix.com/>

02

業務システムに特化 したライブラリ

BootstrapやSemantic UI のような、一般Webサイト用の UIライブラリではなく、業務システムに特化したライブラリなので、「かゆいところに手が届く」UIパーツが盛りだくさん。

デザインも過度にモダンなものではなく、業務システムとしてふさわしいものなので、どんな業務システムにでも組み込みやすい。

03

ローコード!

極めて少ないコードを記述するだけで、高度なユーザインタフェースを直ぐに利用することが可能。

高度なユーザインタフェースを実装するために、FileMaker以外のコードを書く時間が増えては、せっかくFileMakerのようなローコードをチョイスした効果が相殺されてしまう。

Webixを選択する 3つの理由

01

豊富なドキュメント とサンプルコード集

丁寧に作り込まれたドキュメント、そしてコードスニペットを含めたサンプルコードが豊富。

特に Webix に特化した CodePenのようなコードスニペットは強力!

ドキュメント：

<https://docs.webix.com/>

コードスニペット：

<https://snippet.webix.com/>

02

業務システムに特化 したライブラリ

BootstrapやSemantic UI のような、一般Webサイト用の UIライブラリではなく、業務システムに特化したライブラリなので、「かゆいところに手が届く」UIパーツが盛りだくさん。

デザインも過度にモダンなものではなく、業務システムとしてふさわしいものなので、どんな業務システムにでも組み込みやすい。

03

ローコード!

極めて少ないコードを記述するだけで、高度なユーザインタフェースを直ぐに利用することが可能。

高度なユーザインタフェースを実装するために、FileMaker以外のコードを書く時間が増えては、せっかくFileMakerのようなローコードをチョイスした効果が相殺されてしまう。

Webixを選択する 3つの理由

01

豊富なドキュメント とサンプルコード集

丁寧に作り込まれたドキュメント、そしてコードスニペット等を含めたサンプルコードが豊富。

特に Webix に特化した CodePenのようなコードスニペットは強力!

ドキュメント：

<https://docs.webix.com/>

コードスニペット：

<https://snippet.webix.com/>

02

業務システムに特化 したライブラリ

BootstrapやSemantic UI のような、一般Webサイト用の UIライブラリではなく、業務システムに特化したライブラリなので、「かゆいところに手が届く」UIパーツが盛りだくさん。

デザインも過度にモダンなものではなく、業務システムとしてふさわしいものなので、どんな業務システムにでも組み込みやすい。

03

ローコード!

極めて少ないコードを記述するだけで、高度なユーザインタフェースを直ぐに利用することが可能。

高度なユーザインタフェースを実装するために、FileMaker以外のコードを書く時間が増えては、せっかくFileMakerのようなローコードをチョイスした効果が相殺されてしまう。

Webixを選択する 3つの理由

01

豊富なドキュメント とサンプルコード集

丁寧に作り込まれたドキュメント、そしてコードスニペット等を含めたサンプルコードが豊富。

特に Webix に特化した CodePen のようなコードスニペットは強力!

ドキュメント :

<https://docs.webix.com/>

コードスニペット :

<https://snippet.webix.com/>

02

業務システムに特化 したライブラリ

BootstrapやSemantic UI のような、一般Webサイト用の UIライブラリではなく、業務システムに特化したライブラリなので、「かゆいところに手が届く」UIパーツが盛りだくさん。

デザインも過度にモダンなものではなく、業務システムとしてふさわしいものなので、どんな業務システムにでも組み込みやすい。

03

ローコード!

極めて少ないコードを記述するだけで、高度なユーザインタフェースを直ぐに利用することが可能。

高度なユーザインタフェースを実装するために、FileMaker以外のコードを書く時間が増えては、せっかくFileMakerのようなローコードをチョイスした効果が相殺されてしまう。

まとめ

01

FileMaker × Webix ローコードの採用

いくら優れたユーザインタフェースを組み込めると言っても、FileMaker以外のコードを書く時間が増えては、ローコード開発プラットフォームを選択した意味が相殺されてしまう。

しかし、Webixローコードなライブラリを活用することで、この不利益を軽減することができる。

02

Webフロントエンド コードの自動生成

WebixControllerのような、Webフロントエンドを動かすコードを自動生成するツールを予め用意しておくことで、開発時間を圧倒的に短縮することができる。

コードを自動生成するので、コーディング時間を削減する効果はもちろん、バグの内在率は下がり、Webフロントエンドの技術を持たない開発者でもその恩恵を得ることができる。

03

FileMaker Data API

FileMaker Data API の素晴らしい点は、リクエストデータ、そしてレスポンスデータがJSONという標準的なデータ構造が採用されたこと。

Webixとの会話は FileMaker Data API の戻り値をベースに実行されている。これは API のデータ構造がJSONであるからできること。

まとめ

01

FileMaker × Webix ローコードの採用

いくら優れたユーザインタフェースを組み込めると言っても、FileMaker以外のコードを書く時間が増えては、ローコード開発プラットフォームを選択した意味が相殺されてしまう。

しかし、**Webix**のような**ローコードライブラリ**を活用することで、この不利益を軽減することができる。

02

Webフロントエンド コードの自動生成

WebixControllerのような、Webフロントエンドを動かすコードを自動生成するツールを予め用意しておくことで、開発時間を圧倒的に短縮することができる。

コードを自動生成するので、コーディング時間を削減する効果はもちろん、バグの内在率は下がり、Webフロントエンドの技術を持たない開発者でもその恩恵を得ることができる。

03

FileMaker Data API

FileMaker Data API の素晴らしい点は、リクエストデータ、そしてレスポンスデータがJSONという標準的なデータ構造が採用されたこと。

Webixとの会話は FileMaker Data API の戻り値をベースに実行されている。これは API のデータ構造がJSONであるからできること。

まとめ

01

FileMaker × Webix ローコードの採用

いくら優れたユーザインタフェースを組み込めると言っても、FileMaker以外のコードを書く時間が増えては、ローコード開発プラットフォームを選択した意味が相殺されてしまう。

しかし、Webixローコードなライブラリを活用することで、この不利益を軽減することができる。

02

Webフロントエンド コードの自動生成

WebixControllerのような、Webフロントエンドを動かす**コードを自動生成**するツールを予め用意しておくことで、開発時間を圧倒的に短縮することができる。

コードを自動生成するので、コーディング時間を削減する効果はもちろん、バグの内在率は下がるし、**Webフロントエンドの技術を持たない開発者でもその恩恵を得ることができる。**

03

FileMaker Data API

FileMaker Data API の素晴らしい点は、リクエストデータ、そしてレスポンスデータがJSONという標準的なデータ構造が採用されたこと。

Webixとの会話は FileMaker Data API の戻り値をベースに実行されている。これは API のデータ構造がJSONであるからできること。

まとめ

01

FileMaker × Webix ローコードの採用

いくら優れたユーザインタフェースを組み込めると言っても、FileMaker以外のコードを書く時間が増えては、ローコード開発プラットフォームを選択した意味が相殺されてしまう。

しかし、Webixローコードなライブラリを活用することで、この不利益を軽減することができる。

02

Webフロントエンド コードの自動生成

WebixControllerのような、Webフロントエンドを動かすコードを自動生成するツールを予め用意しておくことで、開発時間を圧倒的に短縮することができる。

コードを自動生成するので、コーディング時間を削減する効果はもちろん、バグの内在率は下がり、Webフロントエンドの技術を持たない開発者でもその恩恵を得ることができる。

03

FileMaker Data API

FileMaker Data API の素晴らしい点は、リクエストデータ、そしてレスポンスデータが**JSON**という標準的なデータ構造が採用されたこと。

Webixとの会話は FileMaker Data API の戻り値をベースに実行されている。これは API のデータ構造がJSONであるからできること。

 SpreadSheet

レイアウト/Webビューア名

```
fmLayoutName
webView01
```

T111_見積_webixDemo|FORM
webViewor01

Property

JavaScript Setting

CSS

T111_見積_webixDemo|FORM

webView01

1

0

☒ ボタンを設置する

cejWebixDataTable_clickDetailButton

[詳細](#)

__id

● テストデータ

サイズ

fmdapiResponse.json

83kb

レコード件数

100件



限界のないローコード。
スキルを磨けば、ビジネスが輝く。



Claris Engage Japan 2022

#ClarisEngageJP