

# Webシステム開発の最新トレンド

## — フロントエンドMVCフレームワークの活用事例 —

Webシステムの開発に、新たな手法が導入されつつある。最新のフレームワークを活用することで、ビジネスが要求するスピードに合わせ、Webシステムの構築や改修が可能になってきている。本稿では、Webシステム開発の最新トレンドであるフロントエンドMVCフレームワークについて、導入事例を交え、その効果について解説する。

NRIネットコム Web ネット事業本部  
Web ブランドクリエイション部 主任システムエンジニア

ふじさわ すぐる  
藤澤 優

専門はWebアプリケーション設計、ID認証・認可



### フロントエンドMVCフレームワークとは

フロントエンドMVCとは、主にWebブラウザ上で動的なアプリケーションを実装するために利用されるプログラミング言語であるJavaScriptに、サーバーアプリケーションでよく利用されるデザインパターンのModel - View - Controller (MVC) を組み合わせたものである。このフロントエンドMVCフレームワーク（以下、フロントエンドMVC）をWebシステム開発に採用する動きが活発になってきている。

新しいフレームワークを取り入れ、従来のWebアプリケーションの構造そのものを見直すことで、開発・運用をより効率よく行うことができる。本稿では、フロントエンドMVCについて、従来の開発手法との違いを踏まえ、その導入効果について説明する。

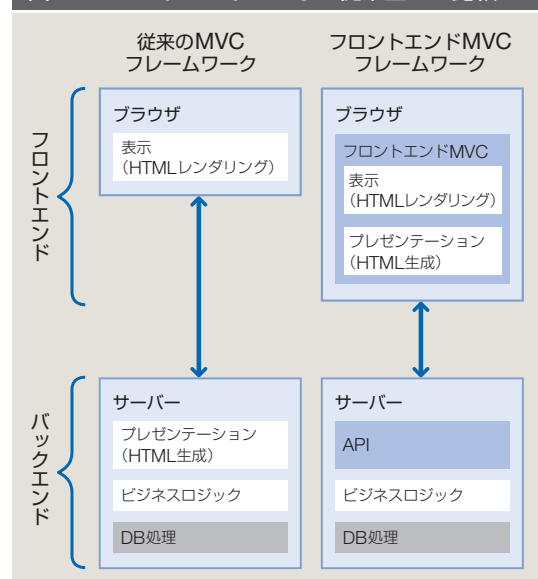
MVCは、従来、サーバーアプリケーションの開発効率向上のために用いられてきた設計思想であったが、この思想がフロントエンドへと移植されている。図1に従来型の

MVCフレームワークとフロントエンドMVCの違いを示す。

フロントエンドMVCを適用する最大の理由は、「表示変更を伴う改修に、タイムリーに対応する」ことである。システム運用において表示の変更は要望が多く、またビジネスに合わせてタイムリーな対応が求められる。

料金計算やユーザー管理などの仕組みは、一度構築すれば頻繁に変更することはない。一方でユーザーに情報を提供するインター

図1 フロントエンドMVCと従来型との比較



フェース部分は、Webサイトにおける商品購入の成約率向上やサービスの利用促進のために、さまざまな変更改修のニーズが頻繁に発生する。このように変更点の多い箇所と少ない箇所をシステムの構造上明確に分割し、ビジネスロジックに影響を与えず、一方でインターフェースを効率よく開発できるようにするフレームワークがフロントエンドMVCである。

フロントエンドMVCのフレームワークとして主要なものを表1に示す。Webサービスをけん引する世界的な企業がフレームワークを開発・提供している点に注目してほしい。提供開始時期を見ると分かるが、ここ数年で開発された比較的新しいフレームワークではあるものの、開発元のWebサービスでは既に自社サイトなどで利用しており、実績のあるフレームワークである。

## フロントエンドMVCの導入事例

NRI ネットコムが、フロントエンドMVCの主要なフレームワークのうち、Google社のAngularを利用し、Webサイトの一部リニューアルを行った事例を紹介する。

リニューアルを行ったWebサイトは、多くの商品をエンドユーザーに提示し、契約へとつなげるサイトである。顧客の契約情報や商品の情報そのものは、これまで大きな変更は必要としていなかった。商品については、ビジュアルが顧客の契約意欲の喚起に直結していたため、Webサイトのデザインリニューアルはこれまでに数回実施されてきた。しかし、例えば、契約成立数の向上を狙った商品

表1 MVCフレームワーク一覧

フレームワーク名	開発母体	提供開始時期
Angular	Google社	2016年
React	Facebook社	2013年
Vue.js	コミュニティ (Evan You 他)	2015年

のマーケティング施策をサイト上で実施したいといった要望があった場合、1つの変更を画面に反映させるために数カ月かかるなど、ビジネスの期待するスピードに対応できないことが問題となっていた。

主な原因は、アプリケーションの構造にある。画面の表示と商品の管理機能が分割されておらず、画面の一部を変更するとシステム全体への影響があり得るため、前述のような画面の変更要件だけに対応することが難しくなっていた。また新たなサービスの展開時なども、その対応には画面表示と管理機能を合わせて修正する必要があるため、大きなコストがかかっていた。

スピードとコストの問題に対応するため、特に商品をエンドユーザーに提示する部分について容易に改修が行えることを目的に、「フロントエンドMVCの導入」と「サーバー側処理のAPI化」という2つの施策を実施した。

「フロントエンドMVCの導入」については、前節で述べた通り、開発の効率化が狙いである。特に当該サイトは、ビジュアル的な訴求が求められており、画面上でのアニメーションや動画再生などの改善を今後も行う予定があったため、その開発を見越した仕組みが求められていた。

「サーバー側処理のAPI化」については、フ

ロントエンドMVCを生かしつつ、業務ロジックへの影響を完全に分離することが狙いである。フロントエンドMVCの採用で開発効率が上がったとしても、業務ロジックへ影響がある修正は開発現場としては受け入れがたい。一方で、修正内容が業務ロジックに対して全く影響がないことをテストで担保しようとすると、コスト増は明らかである。そこで、システムの構造自体をフロントエンドMVCとサーバー側のAPI処理として明確に分け、変更が頻繁に行われる処理はフロントエンドMVC側へ寄せ、業務として変更が少ない処理は接続用の公開APIへと再配置した。図2が現行システムと新システムの比較である。

## 課題と対応策

フロントエンドMVCの導入に当たり、いくつかの課題も上がった。対応策とともに紹介する。

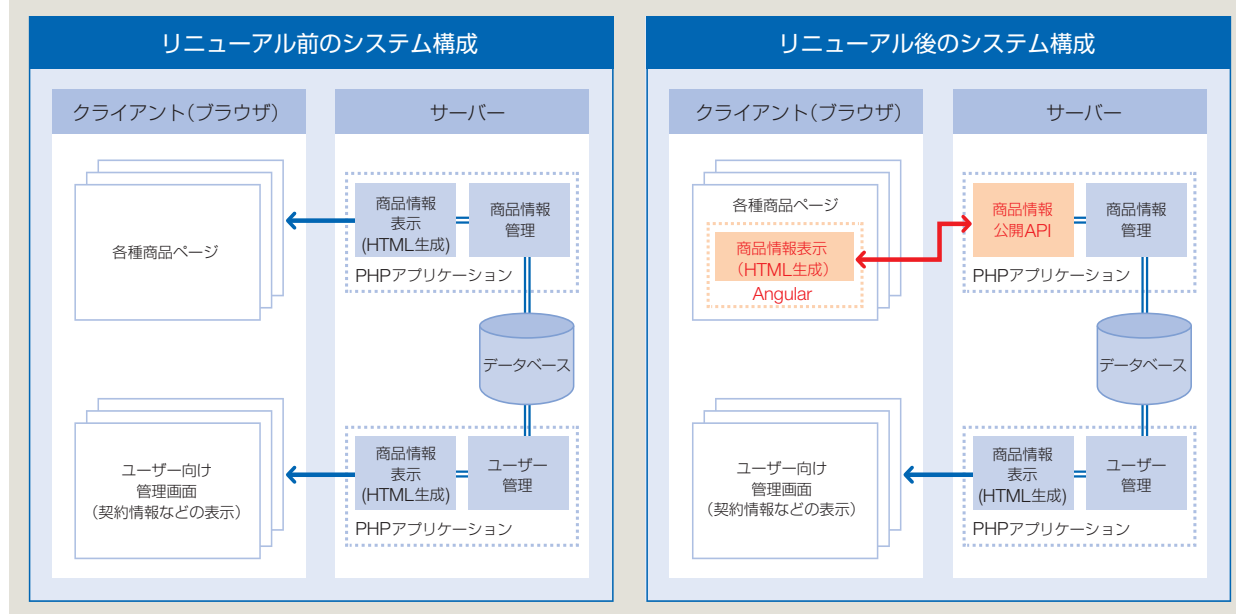
### (1) クローラ対策

GoogleやTwitterでは、ユーザーに代わってWebサイトを巡回し、検索に必要な情報を収集するプログラム「クローラ」を利用している。クローラはJavaScriptを実行できないため、フロントエンドMVCで生成しているHTML領域については情報収集ができず、結果として検索結果に反映されないことがあった。これについては、当該サイトのWebページをキャッシュし、クローラへ応答できるサービスを利用することで、GoogleやTwitterの検索結果に反映されるように対応した。

### (2) パフォーマンス対策

従来と比べて、ユーザーの待ち時間が増加することがあった。ユーザーのブラウザおよび利用端末のスペックに左右されるところもあるが、フレームワークを追加したことにより、それが読み込まれるまでの時間として、約500ミリ秒程度、初期表示の時間が増加した。

図2 フロントエンドMVC導入事例のシステム構成



これについては、表示全体をつかさどるフレームワークの読み込みを他の処理より優先的に行い、初期表示までの時間を300ミリ秒まで短縮した。さらに読み込み中を示すインジケータをアニメーション表示することで、300ミリ秒の待ち時間によるユーザーが感じるストレスを低減させた。

### (3)旧ブラウザ対策

一般的に最新のフレームワークは、古いWebブラウザをサポートしていない。IE9以前のバージョンやAndroid 4.2以前に搭載されているWebブラウザがこれに当たる。これらのユーザーはサイトを訪れるユーザー数の割合からしても少数であったため、未対応のブラウザを利用しているユーザーについては、推奨のブラウザへの移行を促すこととした。

これらのリニューアル施策の結果、商品やサービスの追加・変更時のプロモーションなどを行う際には、フロント側の変更のみで対応でき、柔軟な運用が可能になった。具体的には、商品の表示方法を追加（サムネイル表示および商品説明も含めたリスト表示を追加）する対応を2週間程度で実施する、また顧客が特定の画面にアクセスした場合、顧客の商品嗜好しこうに合わせて契約につながる訴求コンテンツを表示する機能を追加（外部サービスと連動させて2カ月程度で実施）するなど、ビジネスの要求スピードに合った対応が行えるようになった。

また、リニューアル時のフロントエンドMVCの適用箇所を必要な箇所のみ限定することで、変更時の影響を最小化するだけでなく、リニューアル対応の必要コストを抑えることもできた。

## フロントエンドMVCの展望

フレームワークの開発元であるGoogle社やFacebook社では、自社サイトでフロントエンドMVCを利用し、効率的な運営を行っている。これらの会社が提供するサービスでは数カ月ごとにサイトの改善や機能の追加が行われるが、それらを支える技術としてフロントエンドMVCの存在が大きい。

国内のWebシステム全体を見渡すと、最新技術であるフロントエンドMVCを採用しているサイトはまだ多くない。しかしながら、例えばベンチャー企業などが最近サービスを立ち上げたサイトでは、最初からフロントエンドMVCを採用していることが多い。

既存のシステムにあまり影響を与えず、Google社やFacebook社、あるいはベンチャー企業のように短いサイクルでWebサイトの改善やサービス追加を行って、Webサイトを中心としたビジネスを展開したいと考えている企業は、筆者の経験からも数多く存在している。近年のWebシステムの重要性の高まりとビジネスの要求するスピード感を踏まえると、今後、MVCフレームワークは、大手サイトから中・小規模サイトまで広まっていくと予想される。

本稿で示した事例のように、フロントエンドMVCの導入は、既存サイトに部分的に採用し、その効果を確かめつつ適用範囲を広げていくといった導入も可能であり、導入の敷居は高くない。長期的なWebサイト運用効率向上と、ますます加速するビジネスの要求に備えて、早い段階でフロントエンドMVCの導入を検討いただきたい。 ■