

フロントエンド技術を中心とした軽量な Web フレームワークの開発と評価

山口 湧太[†] 古谷 文弥[†] 田胡 和哉[‡]

東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻[†]

東京工科大学コンピュータサイエンス学部[‡]

1 はじめに

Web サービスの第一印象は、利用者が最初に訪れるポータルサイトによって決まる。そのため以下の3点が要求される。

- 利用者を惹きつける豊かな表現能力と快適な操作感
- 機能変更の容易さと高い拡張性
- 上記の2点を満たしつつ、サーバやネットワークの負荷を軽減できる機構

本稿ではフロントエンド技術を中心としたポータルサイト向けの軽量な Web フレームワークの実装について述べる。また、実利用することで評価を行う。

2 フレームワークの仕様

本稿で開発するフレームワークを「Vimana」と命名した。Vimana を用いたポータルサイトを構成する要素は、ナビゲーションメニュー、コンテンツページ、カード型プラグインである。ナビゲーションのメニューとコンテンツページは1対1対応となっている。

Vimana では、前述の3点を満たす為に、次のアプローチを取ることにした。まず、利用者を惹きつける豊かな表現能力を実現するために JavaScript による UI・UX ライブラリと CSS によるデザインテンプレートを提供する。機能変更の容易さと高い拡張性を実現する手法としては2種類の拡張機構を用意する。ひとつはトップページとなるホーム画面にカード型の小機能を追加するプラグイン機構。もうひとつは、ホーム画面とは別のコンテンツページを増やす機構である。サーバやネットワークの負荷を軽減する機構としては、シングルページアプリケーション[1]を採用する。

Development and evaluation of lightweight Web framework that the front-end technology with a focus.

[†] Yuta YAMAGUCHI, Fumiya FURUYA

Bionics, Computer and media science, Entrepreneurship program, Tokyo University of Technology Graduate school.

[‡] Kazuya TAGO

School of Computer Science, Tokyo University of Technology.

3 実装

前述の仕様をもとに設計し、Vimana フレームワークの実装を行った。また、ソースコードはオープンソースとして公開した。Vimana はフロントサイドを JavaScript ライブラリと CSS ライブラリで実装される。JavaScript ライブラリでは、UI・UX だけでなくカード型プラグイン用の API も提供している。また、サーバサイドは Java ライブラリとして実装される。

Vimana フレームワークの全体像を図1に示す。

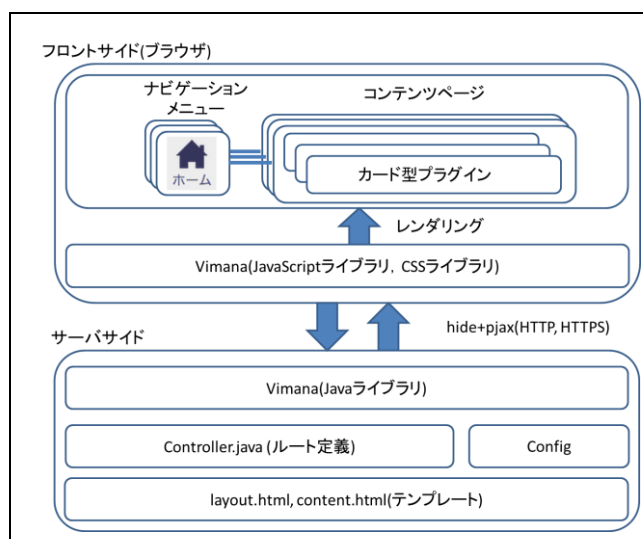


図1: Vimana フレームワークの全体像

Vimana を利用するポータルサイト開発では、開発者は Controller によるルート定義と各コンテンツページの HTML ファイル、Config ファイルの記述が最低限必要となる。また、一般的なテンプレートエンジン機構を持つ為、全体を構成するレイアウト部分とコンテンツ部分で HTML ファイルを分ける必要がある。

3.1 サーバ負荷軽減技術 hide+ajax

サーバ負荷軽減技術として、既存技術である pjax にレンダリングエンジンによる非表示機能を組み合わせた「hide+ajax」を提案した。

hide+ajax は、シングルページアプリケーションという技術概念とその実現技術である pjax をもとに実装を行っている。pjax はシングルページ

アプリケーションを実現する技術として注目されており、URL の履歴を書き換える History API を用いて現在の WEB ページの URL を書き換えつつ、Ajax を用いてコンテンツを書き換えるものである。pjax によってコンテンツに一意の URL を振り分けることができるため、ユーザビリティを損なうことなく、シングルページアプリケーションを実現させることができる。しかしながら、pjax ではページ遷移ごとにリクエストが発生している。ポータルサイトのような WEB ページでは、即時的な反映が必要であるコンテンツは少ない。そのため一度表示したものに対してはキャッシュを行うことが有効である。pjax に加えブラウザ上でコンテンツの非表示化によるキャッシュを行うことで、余分なリクエスト回数を削減することができると考え、それを hide+pjax と命名した。

4 利用例

東京工科大学では、学生に対してクラウドサービスの提供を行うプロジェクトが実施されている[3]。その一環として独自のポータルサイトを開発し、2014 年 4 月から全学生・教員、約 8000 人に向けて公開している。このポータルサイトでは、本稿で開発した Vimana フレームワークを使用している。

東京工科大学のポータルサイトを図 2 に示す。

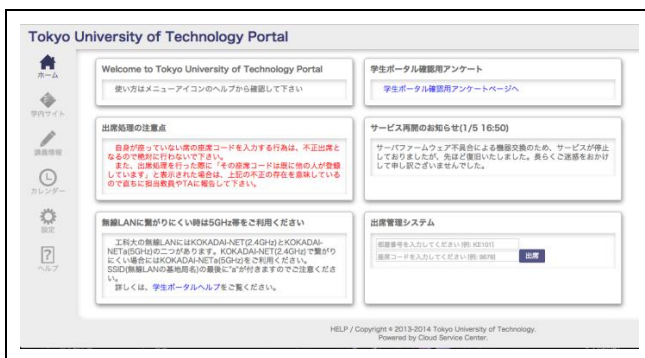


図 2：東京工科大学のポータルサイト

東京工科大学のポータルサイトではカード型プラグインとして以下の 2 つが実装・運用されている。

- お知らせシステム
hide+pjax 機能により、即時的なメッセージの反映が困難となっている。そこで、お知らせシステムとしてプラグインを実装することによって、緊急を要するサービスのメッセージ通知等を行うことができる。
- 出席管理システム
東京工科大学では、クラウドを用いた出席管理システムの提供も行っており、ユーザインタ

フェースはポータルサイトのプラグインとして実装されている。この出席管理システムでは学生個人の履修状況によって、必要な時間に自動で出席フォームが立ち上がるように設計されている。

5 評価

東京工科大学での運用データをもとに評価を行った。フレームワークを使用しない場合のレスポンスデータ量は一日平均で約 5,772MB であった。

フレームワーク適用後の平日一週間のレスポンスデータ量を表 1 に示す。

表 1：平日一週間のレスポンスデータ量

	アクセス数	データ量[byte]
2014/12/15	12,392	144,623,903
2014/12/16	8,584	95,955,847
2014/12/17	16,085	189,088,700
2014/12/18	10,929	120,143,603
2014/12/19	11,118	133,806,448
平均	11,821	136,723,700

上記表の通り、一日平均で約 130MB のデータ量に抑えることができている。これは hide+pjax によってリクエスト回数を大きく削減できたことが要因であると考えられる。

6 おわりに

本稿により、フロントエンド技術を中心とした軽量な Web フレームワークの開発を行うことができた。特に hide+pjax は負荷軽減に大きな効果をもたらしている。

しかしながら、hide+pjax ではコンテンツ量の増加に対するメモリ使用量が問題であり、将来的には一定の使用量によりリフレッシュを行う機構が必要である。

また、オープンソースとして公開したことで、より多くの開発者によって改修されると期待している。

参考文献

[1]Michael S. Mikowski, Josh C. Powell (2014) 『シングルページ Web アプリケーション—Node.js, MongoDB を活用した JavaScript SPA』オライリー・ジャパン
 [2] PJAX : "PJAX 0.5 seconds Faster Website" <http://falsandtru.github.io/jquery-pjax/> (2015/01/07)
 [3]東京工科大学 : "クラウドサービスセンター" <http://www.teu.ac.jp/cloud> (2015/01/07)