

共有化を狙った Web アプリケーション仕様の提案とサンプルの開発の試み

中野 裕司^{1,4,2,3}, 永井 孝幸^{1,3,2,4}, 松葉 龍一^{2,4,1}, 喜多 敏博^{2,4,1}, 杉谷 賢一^{1,3,4,2}, 宇佐川 毅^{3,4,1}

¹熊本大学 総合情報基盤センター, ²熊本大学 eラーニング推進機構, ³熊本大学 大学院自然科学研究科,

⁴熊本大学 大学院社会文化科学研究科 教授システム学専攻

nakano@cc.kumamoto-u.ac.jp

概要: 大学における学習支援や各種手続き等の ICT 化は年々進みつつあり、それらの殆どは Web アプリケーションで実現されている。シングルサインオンによる認証の連携は進みつつあるが、各種機能の連携はなかなか難しい。また、大学特有のものや小規模なもの等は、パッケージではなく独自開発や外部発注等を行うことも多い。このような場合、個々の Web アプリケーションの仕様や動作・開発環境が異なるため、後からの修正はなかなか難しいのが現状である。そこで、Web アプリケーションの大まかな仕様を定め、サンプルプログラムとともに提供することを考えた。具体的には、ブラウザとのデータのやり取りを Web サービス(Rest, JSON)に定め、ユーザインターフェースの国際化を含め、表示はブラウザ上で行うことを原則とした。これにより、サーバ側は、Rest に対応し、シングルサインオンと権限管理を一箇所でを行い、全てのサービスを一括管理することとした。このような仕様の提案とサンプルプログラムに関して紹介する。

1 はじめに

大学における学習支援や各種手続き等の ICT 化は年々進みつつあり、それらの殆どは Web アプリケーションで実現されている。シングルサインオンによる認証の連携は進みつつあるが、各種機能の連携はなかなか難しい。また、大学特有のものや小規模なもの等は、パッケージではなく独自開発や外部発注等を行うことも多い。このような場合、個々の Web アプリケーションの仕様や動作・開発環境が異なるため、後からの修正はなかなか難しいのが現状である。

また、Web アプリケーション自体も近年大きく変わりつつある。旧来のページ遷移型、すなわち、サーバ側で HTML ページを構成してから Web ブラウザへ送る形式から、Web ブラウザはサーバにデータのみを要求し非同期通信で得たデータによってページ遷移を伴わずに自分自身を書き換える形式へと大きく変化しつつある。現状では、Web アプリケーションを内製や依頼等によって新たに開発する場合、旧来の手法と新しい手法のどちらか、又は、混合した状態になることが多いと思われる。また、モバイルデバイスへの対応を考えると旧来のページ遷移型は基本的には全てのページを対応させる必要があって難しい。

大学における各種サービスは認証はもとよりその扱うデータや機能の関連性が強く、例えば、学務システム、シラバスシステム、eラーニング関係システム等は、科目情報、受講情報、学生や教職員の情報等を共有したり、お互いのデータを参照・表示する必要性がある。また、教育制度の改革や様々な取り組みによって、

各機能の修正も頻繁に起きる。

しかし、最初のほうで述べたように、Web アプリケーションの仕様や作・開発環境が異なったり、新旧の大きく異なる手法が入り乱れた状況では、なかなか連携が難しい。また、新たに開発する部分もガイドライン等なしに進めていくと、このような状況がさらに長くつづくことが懸念される。

そこで、本取り組みでは、その Web アプリケーションの大まかな仕様を定め、ページ遷移型ではなく Web API を使った非同期データ通信型を採用し、カスタマイズを用意するために出来る限りクライアントリッチな構成とした。さらに、サンプルプログラムを用意することで、内製であろうが外部に開発を委託する場合にも、その構成をある程度維持することを考えた。

2 開発仕様

2.1 基本的な開発方針と構造

作成した本人または関係者でなくとも、後述の一般的スキルをもった者であれば、後から修正や追加開発が可能なことを目指し、基本的なシステムの開発方針や構造を、以下のように全体、サーバ、クライアント各々に設定した。

- 全体
 - 出来る限りクライアントリッチな構成とする。サーバ側は基本的に認証と WebAPI を用意するだけで、HTML の生成等は Web ブラウザ側で行う。
- サーバ側

- Web サービス(API)を REST 形式で提供する。送受信データは JSON 形式とする。
- シングルサインオンにフィルタ形式で対応する。また、ログインユーザ等の情報が取得できること。(同一サービス内の個々のプログラムで対応するのではなく、全体又は一部にシングルサインオンのは以下に置けること)
- シングルサインオンとは別に、一元管理された関数やメソッド等で、ユーザ属性によって個別に利用制限が掛けられること。(個々のプログラム内で記述せず、一元管理する。)
- プログラミング言語は、Java, PHP, Ruby on Rails 等汎用かつオープンソースを用いる。フレームワークは可能な限り使用を控え、利用する場合は汎用かつオープンソースなものを用いる。
- データベースへのアクセスは、可能な限り SQL 言語の範囲内で行う。
- クライアント側
 - 出来る限りの処理をクライアント(Web ブラウザ)側で行うことで、ユーザインタフェース等の変更、別の Web サービスの利用、マッシュアップ等を容易にする。また、修正の多くをクライアント側のプログラミングで行えるようにする。ページ遷移を安易に行わない。
 - サーバとのデータの授受は、REST 型の Web サービスを用いて JSON 形式のデータで行う。
 - 国際化は原則としてクライアント側で行う。ただし、データベースを伴うものは、Web サービスと併用する。
 - モバイルデバイスへの対応は原則としてクライアント側で行う。デバイス単位に別のページを用意するのではなく、一括して行えるような仕組みを用いる。

2.2 REST 開発に必要なスキルの設定

一般的なスキルを持っていれば、誰でも開発や修正が行えることを目指し、以下のように必要なスキルをサーバとクライアント各々に設定した。

- サーバ側の開発スキル
 - REST による Web サービスが構築できること。
 - JSON 形式のデータが扱えること。
 - データベースの構築と SQL 言語が使えること。
 - Java, PHP, Ruby on Rails 等の汎用かつオー

ンソースなプログラミング言語の中から開発言語を選んで開発が行えること。

- クライアント側の開発スキル
 - JavaScript, HTML, CSS が使えること。
 - jQuery 等による同期・非同期通信、HTML の書き換え等が行えること。
 - jQuery のプラグイン等を利用して、クライアント側で国際化やモバイルデバイスへの対応が行えること。

以上のスキルは特別なものはなく、関連する解説書も数多く出ている。

2.3 サーバと Web ブラウザのデータ授受

ここは重要なところで、サーバとクライアントである Web ブラウザ間のデータのやり取りの形式(Web API)をしっかり規定しておけば、既に Web API が存在するものは、サーバ側のプログラミンを全く伴わずに開発が可能である。

- サーバ側
 - Web サービス(API)を REST 形式で提供し、その仕様をパラメータ等についても明記する。REST のサポートのためのライブラリ等が必要になる可能性がある。
 - 送受信データは JSON 形式とする。クライアント側は JavaScript で直接扱えるが、サーバ側は何らかのパーサが必要になる可能性がある。
- クライアント側
 - 必要ない限りページ遷移は行わず、サーバとのデータのやり取りでページの一部を書き換える。
 - jQuery 等による同期・非同期通信により、サーバと REST 型の WebAPI を用いて JSON 形式のデータを送受信する。
 - 原則として、jQuery の利用を前提としている。

2.4 シングルサインオンと認証

Web アプリケーション間の連携にはシングルサインオンや認証は必須かつ重要な要素であるが、その実現方法がまちまちだと深刻なセキュリティホールとなり得る。

- サーバ側
 - シングルサインオンにフィルタ形式で対応する。同一サービス内の個々のプログラムで対応するのではなく、ディレクトリ単位等のグループで制

限が掛けられること。例えば、Web サービスも含めてシングルサインオンにできること。

- シングルサインオンとは別に、一元管理された関数やメソッド等で、ユーザ属性によって個別に利用制限が掛けられること。例えば、特定のユーザないしグループのみアクセス可能な Web サービスが定義できることが必要。ただし、このような処理を個別に行なっているとミスや漏れになりやすいので、一元管理すべきである。
- クライアント側
 - 認証が通ったユーザとそうでないユーザで画面表示を変更するといった場合、JavaScript のソースコード等にその内容を書きってしまうと、セキュリティ上問題であるため、この場合はサーバ上で最初のページを生成したほうが安全かもしれない。シングルサインオン配下の Web サービスで表示データを取得することも可能であろうが、処理が複雑化する可能性がある。

2.5 国際化

大学のグローバル化と留学生対応のため、大学が提供するサービスの多くで国際化が必須となりつつある。まず、Web アプリケーションに国際化が可能な仕組みを導入してあることが当然必要であるが、その手法がまちまちだと後で修正や言語の追加等が大変なので、統一することとした。

- サーバ側
 - 原則として、ユーザインターフェース等の国際化はサーバ側では行わない。ただし、データベース等にあるコンテンツに関しては、Web サービスに locale を渡すことで切り替える。
- クライアント側
 - 原則として、ユーザインターフェース等の国際化は全てクライアント側で行う。これにより、クライアント側のプログラミングだけで国際化を行うことが可能になる。
 - 原則として、jQuery のプラグインである jquery-i18n-properties [1] の利用を前提としている。この利用で、java の ResourceBundle クラスと同様にプロパティファイルで国際化に対応でき、旧来のサーバ側での実装と同様の手法が使える。

2.6 モバイルデバイスへの対応

従来のページ遷移型の Web アプリケーションはモバイル対応を行うために、多くの場合、全てのページの修正が必要になる。この意味でも、ページ遷移型からの移行は意味があると思う。また、個別のモバイルデバイス各々に対応するページを作成するのもコストが大きい。そこで、原則として、全体を通してモバイル対応が可能な仕組みを導入することが望ましいと判断した。なお、原則としてクライアント側で対応する。

- クライアント側
 - クライアントの Web ブラウザがどんなデバイスにあっても同一システムで対応するのが理想と考える。
 - 原則として、jQuery Mobile [2] の利用を前提としている。jQuery Mobile は PC から各種モバイルデバイス上の Web ブラウザに対応しており、モバイルデバイスでの利用に対応する必要がある場合は、最初から jQuery Mobile のユーザインターフェースを用いて構築すべきと思う。

3 サンプルプログラム

以上のような仕様を満足するためのサンプルプログラムをモバイルデバイスへの対応を除き作成した。

サンプルプログラムは、サーバ側を、JSP、サーブレット、JSONIC ライブラリ [3] を利用し、クライアント側を、HTML、CSS、jQuery とそのプラグインで作成し、ソースコード、日本語と英語併記の解説 Web ページを含んだ war ファイルとして、以下のものを作成した。

また、全てを同時に含むのではなく、わかりやすくするため、主に単一の機能を追加した形で作成した。

- restSample00.war (※) : REST サービスのみ
- restSample00cas.war : ※に加えて、シングルサインオンの cas のサポート
- restSample00i18n.war : ※に加えて、国際化のサポート
- restSample02.war : ※に加えて、jdbc による MySQL のサポート

以上のサンプルプログラムの実行例を図 1 に、その解説を図 2 に示す。

4 まとめと今後の課題

大学における Web アプリケーションの連携等を考え、Web アプリケーションの大まかな仕様を定め、サン

プルプログラムとともに提供した。具体的には、ブラウザとのデータのやり取りを Web サービス (Rest, JSON) に定め、ユーザインターフェースの国際化を含め、表示はブラウザ上で行うことを原則とした。これにより、サーバ側は、Rest に対応し、シングルサインオンと権限管理を一箇所で行い、全てのサービスを一括管理することとした。このような仕様の提案とサンプルプログラムに関して紹介した。

しかし、モバイルデバイスへの対応のサンプルがまだできておらず、また、既に作成した部分も改善すべき点も多いと思われる。今後、実際に使いながら改善を行っていきたい。

参考文献

- [1] i18n-properties (2013 年 11 月 1 日確認):
<http://code.google.com/p/jquery-i18n-properties/>
- [2] jQuery Mobile (2013 年 11 月 1 日確認):
<http://jquerymobile.com/>
- [3] JSONIC ホーム (2013 年 11 月 1 日確認):
<http://jsonic.sourceforge.jp/>
 (Java 用の JSON エンコーダー/デコーダーライブラリで、RPC/REST Web サービス可能。)

English Japanese

Rest Sample 00 (国際化版)

[このサンプルの解説](#)

科目を選択してください:

科目名	計算機活用教育論
曜日	木
時限	第5限
内容	情報コミュニケーション技術 (ICT) を活用した教育システム (eラーニング等) に関しては、その仕組みを授業の中で体験・活用しながら講義形式で学習する。 Java言語に関しては、PC教室でLinux上のEclipse環境を利用して、オブジェクト指向に従って、単純な「もの(プログラム)」に、グラフィックス、動き、GUI、Applet、可能ならばSP等の機能を徐々に追加しながら、実習形式で学習する。こちらも、課題提出、相互評価等にICTを活用する。

図 1. サンプルプログラムの例 (国際化)

English Japanese

Document for restSample00i18n / restSample00i18nの解説

2013-05-21 (since 2013-04-22/restSample00), by nakano@cc.kumamoto-u.ac.jp

Dynamic web project on Eclipse (Directory Structure / ディレクトリ構成)

- restSample00i18n
 - java Resources
 - src
 - rest sample
 - QueryDummy.java (Preparing dummy data instead of DB / DBの代わりにダミーデータ作成)
 - SampleService.java (REST service / RESTサービス)
 - WebContent
 - js
 - jquery-1.9.1.min.js (jQuery library / jQueryライブラリ(JavaScript))
 - excanvas.js (Support HTML5's canvas for old IE / 古いIEのためのHTML5のcanvasタグサポート(JavaScript))
 - jquery.i18n.properties-min-1.0.9.js (jQuery internationalization (i18n) library / jQuery国際化(i18n)ライブラリ(JavaScript))
 - bundle
 - syllabus.properties (default messages / デフォルトメッセージ)
 - syllabus_ja.properties (messages in Japanese (locale: ja) / 日本語メッセージ (ローカル: ja))
 - WEB-INF
 - lib
 - jsonic-1.3.0.jar (Support REST and JSON / REST及びJSONのサポートライブラリ(java))
 - web.xml (configuration / 初期設定)
 - restSample00i18n.jsp (Client program / クライアントプログラム (JavaScript))

図 2. サンプルプログラムの解説の例 (国際化)