Web アプリケーションの作成基準

2009/08/24

この文書では、各種 Web ベースのアプリケーション（Perl 等を利用した CGI プログラムだけでなく、PHP や Servlet 等も全て含む）を作成するにあたっての指針を示す。

本基準に示す内容は、セキュリティ・アクセシビリティ・ユーザビリティの確保を基本的な目的として、個別の要件を具体化するものである。本基準に準拠することによって上記の基本的な目的に反することになると思われる場合は、本来の目的を考慮し適宜判断すること。

基本的動作とセキュリティ

アプリケーションを運用するサーバ、利用者の環境にセキュリティホールを作らず、様々なブラウザで正常な動作を行うため、以下の内容を守ること。

1. 作成を検討するアプリケーションについて、もし同機能または近い機能を備えたアプリケーションが既にある場合は、その既存アプリケーションを共同利用するよう努め、個別開発を極力避けること。特に、申し込みや意見募集等の簡易な送信画面については、外部向けには簡易申請システム、内部向けには共通アンケートフォームプログラムが既設で存在するため、それらの利用について最初に検討すること。
2. HTTP、SMTP 等のプロトコルその他そのアプリケーションが使用する技術について、その仕様に則り適切に処理すること。特に HTTP ステータスコードを適切に使い分けるよう注意を払うこと。
3. 出力する HTML は、原則として WEB ページ作成基準に準拠すること。ただし、庁内専用 Web サーバで運用する場合は、使用するページテンプレートについては YCAN Web ページテンプレートを使用すること。アプリケーションの動作上、HTML 4.01 Strict に準拠した構成とするとユーザビリティを損なうことになってしまう場合は、HTML 4.01 Transitional や HTML 4.01 Frameset を使用して良いものとするが、視覚障碍者等の利用に支障が出る可能性があることについて、想定する利用者にとって問題とならないことを確認すること。

4. FORM における action 属性として mailto: スキームでの送信先指定は用いてはならない。
5. Web サーバにファイルをアップロードする機能を備える際、日本語等のマルチバイト文字が含まれるファイルがアップロードされた場合に、サーバ上では英数字のみで構成される別の物理ファイル名で格納するか、DBMS 上に格納し、生のマルチバイト文字がサーバ上で物理ファイル名として保存されないようにすること。また、そのファイルをユーザがダウンロードする際には、できる限り元のマルチバイト文字のファイル名でダウンロードされるように配慮すること。
6. プログラムの中で意図的に出力する場合や、本当にエラーが発生した場合を除き、通常の正常動作時に Web サーバのエラーログに情報が出力されないようにすること。
7. 排他制御を適切に行い、同時に複数のアクセスがあっても矛盾のない処理を行うこと。
8. ユーザが URL を直接入力することによって、非公開のデータ等をプログラムを通さず直接覗けてしまうようなことがない作りとすること。
9. ユーザがプログラムへパラメータを直接渡すことによって、できてはならない操作ができてしまうようなことがない作りとすること。
10. その他、ユーザが受信した HTML、JavaScript 等のソースや HTTP ヘッダ等の通信内容を解析しても不正アクセスを行うための手がかりとなる情報を得られない作りとすること。
11. パスワード等の重要な情報は暗号化して保存すること。暗号化の手法は、特段の理由がない限り不可逆のハッシュとすること。
12. ブラウザやデータベース等から取り出すパラメータに想定外の文字列が入っていることを前提とした設計とし、HTML やファイルパス、シェルコマンド、SQL 等で特別な機能を持つ文字は、全て適切にエスケープすること。このエスケープは、入力時ではなく、HTML や SQL 等への埋め込む時等、アプリケーションの外部に渡す時に行うことを原則とすること。
13. 利用者が電子メールアドレスを入力する欄を設ける場合は、原則として管理者やその本人等、限定されたユーザ以外がそのアドレスを見ることができないようにすること。もしインターネット向けに不特定多数の利用者にも見せる必要がある場合は、アドレスの @ を数値文字参照 @ に変換するか、全角文字の＠やその他全く異なる文字列等に変換して出力する等、迷惑メール対策に十分に気を配ること。
14. 個人情報や ID・パスワード、その他機密性の高い情報を送信する必要があるような FORM を設置する際は、SSL を利用すること。SSL を利用した FORM を作成する際は、SSL を利用したページと、SSL を利用しないページの両方を用意し、利用者が選択できるようにすること。Web サーバ証明書については、信頼済みルート証明機関として一般的なブラウザに登録済みの認証局か、LGPKI Application CA から発行を受けた証明書を利用すること。庁内の場合は、横浜市庁内認証局から発行を受けた証明書を利用すること。
15. サーバ証明書に LGPKI を利用する場合は、問題なく利用できるブラウザが Windows XP または Mac OS X 10.4 以降の OS 標準ブラウザ（Internet Explorer および Safari）

に限られてしまう点に注意し、それ以外のブラウザ（Firefox や携帯電話等）でも利用される可能性がある場合は、ルート証明書の組み込み手順の解説や SSL 非対応の画面を用意する等の配慮を行うこと。

16. SSL を利用したページでは、HTML 本文だけでなく、そのページ内で使用される画像・CSS・JavaScript 等の部品の読み込みについても SSL でアクセスさせること。特に市共通ヘッダパーツ、CSS 等について HTTP の絶対 URL 表記で読み込む指定を行わないよう注意すること。
17. その他本文書に明記のない項目についても、情報処理推進機構（IPA）が提供する「安全なウェブサイトの作り方」および「セキュア・プログラミング講座」また各時点での最新の情勢を踏まえ、セキュリティ対策に十分に気を配って作成したうえで、セキュリティテストを行い、脆弱性が無いことを確認すること。
18. 初期構築時に存在しなかった、または想定していなかったブラウザでの利用において、サービス稼働後、サービス利用そのものの可否に関わるような大きな問題点が発見された場合は、判明次第プログラムを適切に修正すること。

アクセシビリティとユーザビリティ（基本事項）

多様な環境で適切に利用可能なものとするため、以下の内容を守ること。

1. アドレスバー、ツールバー、ステータスバー等、ブラウザの基本的操作や情報提供に使用される領域を非表示にしたり、標準と異なる表示・動作をさせないこと。また、マウスボタン右クリック等によるコンテキストメニューの利用を阻害しないこと。ただし、ヘルプ表示を目的としたサブウィンドウ等、開かれた後は読んで閉じさせるだけの利用を想定した小さなポップアップウィンドウについてはこの限りではない。
2. 著しくユーザが不便を被る等の特殊な事情がない限り、ブラウザの既存ウィンドウサイズや位置をアプリケーション側で操作し変更したり、ユーザによる任意のサイズ変更やスクロールを行えない状態にしないこと。
3. 著しくユーザが不便を被る等の特殊な事情がない限り、新規ウィンドウのポップアップを行わないこと。やむを得ず行う場合は、メインウィンドウより小さいウィンドウサイズを指定し、既存ウィンドウに覆い被さるようなことがないようにすること。また、モーダルダイアログボックスとして表示することが望ましい。
4. 日付を選択するためのカレンダーをポップアップさせる場合は、別ウィンドウではなく、同一ウィンドウ内にレイヤーとして表示すること。
5. FORM における SUBMIT の種類（METHOD）は GET メソッドを原則とし、サーバ側のデータの変更を伴うものについてのみ POST メソッドとすること。
6. ユーザの情報入力を伴わない単なる画面遷移等、A 要素による通常のリンクで実現できる内容は原則として通常の A 要素でリンクし、無闇に INPUT 要素や BUTTON 要素、

JavaScript 等を使わないこと。ボタンのような見た目を表現したい場合は CSS で実現すること。

7. INPUT 要素や BUTTON 要素以外のリンクを CSS によってボタンのように見せかける時は、文字部分だけでなくボタンとして見せる矩形領域全体をクリッカブルにすること。またマウスオーバー時とクリック時の見た目に変化を与え、それがボタンであることや、押したことが解るようフィードバックすること。
8. 画像によるボタンはできる限り使用を控えること。もし使用する場合は、マウスオーバー時とクリック時の見た目に変化を与え、それがボタンであることや、押したことが解るようフィードバックすること。
9. サーバ側に持つセッション情報はログインユーザ情報等セキュリティ上必要最低限のもののみとし、画面遷移にかかわる情報は原則としてサーバ側に保持しないこと。また、Referer によるアクセス制限等も行わないこと。これにより、ユーザが任意に複数のウィンドウを開き、それぞれ並行して別々の操作を行うことを妨げないこと。
10. ユーザから入力されたデータに URL やメールアドレスと推測される文字列が含まれていた場合、特段の支障がない限り、表示時にハイパーリンクとして加工し出力すること。
11. 全市の情報をデータベース化し、まとめて一覧・検索等を行えるプログラムを作成する場合は、全市単位だけでなく、区局単位で一覧・検索できる画面を用意すること。それは、ログインユーザにしか見せないようなシステムの場合を除き、特段のセッション Cookie 等を持たないブラウザからでも GET メソッドによる HTTP リクエストで閲覧できる状態にすること。また、その URL には ? を含まないこと。

アクセシビリティとユーザビリティ（FORM 入力）

1. 画面設計にあたっては、初めて利用するユーザでも、画面の指示に従って入力していくだけで重大な間違いのない結果が得られるように工夫すること。
2. 何か補足説明がなければ利用が困難な場合、それは別のマニュアルではなく、入力画面そのものに説明を併記するか、ポップアップするか、補足説明へのリンクを設けること。この補足説明は、後日プログラム本体に手を加えることなく修正できるような設計としておくこと。
3. 入力不足、不適切な入力、その他利用者が犯すミスで、その誤りを機械的に検出可能な場合は、プログラムが適切な警告を表示し、再入力を促すこと。再入力・修正を促すメッセージは、できる限り早いタイミングでユーザに知らせるように努めること。
4. ユーザによって半角カナ（JIS X 0201 片仮名）が入力された場合、原則としてプログラムが全角カナ（JIS X 0208 片仮名）に修正して処理すること。
5. ユーザによって「保土ヶ谷区」「都築区」が入力された場合は、原則としてプログラムが「保土ケ谷区」「都筑区」に修正して処理すること。
6. ユーザによって機種依存文字（JIS X 0208 で未定義とされている 9～15 区、85～92 区の文字、および JIS X 0213 で新規に定義された 2 面の文字）が入力された場合、理由を提示し入力を拒否（修正を要求）するか、代替となる機種依存でない文字の組み合わ

せに自動的に変換して処理すること。なお、修正を要求する場合は、具体的にどの文字が機種依存文字なのかを明示すること。

7. 全角文字、半角文字の区別をユーザに行わせないこと。例えば、メールアドレス等、半角であることが必須であるフィールドに全角アルファベットを入力された場合、「半角文字で入力して下さい」等の表示を行うのではなく、プログラムが自動的に半角に変換すること。
8. 上記の例以外についても、ユーザに再入力を要求するまでもなくプログラムが自動的に修正することができる問題については、プログラムが自動的に修正すること。(修正したという事実を参考としてユーザに提供する必要がある場合は参考として表示しても構わない)
9. テキストボックスでユーザから入力を受けたデータの先頭や末尾に空白文字がついていた場合は、全角・半角問わず自動的に削除すること。ただし、検索以外の用途に用いる複数行入力欄で、各行頭で 1 つだけ使用されている全角空白文字に限り、削除しなくても良いものとする。
10. その入力結果が最終的に PRE 要素でフォーマット済みテキストとして表示されることを目的としたものである場合を除き、連続した空白文字は全角半角を問わず 1 つの半角空白にまとめること。ただし、検索以外の用途に用いる複数行入力欄で、各行頭で 1 つだけ使用されている全角空白文字に限り、半角に変換しなくても良いものとする。
11. ふりがな入力欄を設ける場合は、原則としてひらがなで入力させることとし、不必要にカタカナ入力を求めないこと。
12. 電話番号にハイフンを入れるかどうか等、複数の入力の仕方が想定されるテキストボックスには、どのように入力すべきかユーザが迷わずに済むよう、入力例を添えること。入力例については、読み上げを考慮し、テキストボックスより先に読み上げられる配置とすること。また、ハイフンを含まない数字の羅列を期待する入力欄においてハイフンが含まれていた場合、エラーとするのではなくプログラムが自動的に削除することが望ましい。
13. 必須項目の入力漏れ等、ユーザに再入力を求めるような場合、「ブラウザの戻るボタンを押して下さい」という旨の画面を表示するのではなく、入力済みの項目があらかじめ埋まった画面を再表示したうえで、どのフィールドにどのような問題があるかを分かり易く表示すること。JavaScript の history.back() や、それに類する機能も使用しないこと。
14. ユーザの誤りを指摘する・修正するだけでなく、その前段として「どのような画面設計にすればユーザが間違いにくいか」を考慮して設計すること。
15. FORM の入力に原則として時間制限を設けないこと。何らかの事情により入力時間に制限を設けざるを得ない場合は、あらかじめその旨を明記すること。
16. FORM の各入力要素、特にチェックボックスとラジオボタンには、LABEL 要素を使い、テキストとの関連を明示すること。

17. ユーザが情報を POST するための FORM を設置する際は、入力 FORM だけでなく、原則として内容確定前の確認画面、確認結果を訂正する画面、送信が完了したことを通知する画面も用意すること。
18. 複数画面に渡って流れが進むシステムでは、各画面において、そのフローが全部で何画面あり、今何画面目にいるのかを表示するよう努めること。
19. テキストボックスを持った FORM を設置する際、それがその画面の主たる機能である場合は、JavaScript を利用し、画面表示時に最初のテキストボックスに自動的にフォーカスさせること。
20. FORM での入力内容のチェックは、サーバ側でのチェックに加え、SUBMIT 時に JavaScript を用いて適宜エラーダイアログを表示し、実際にサーバに送信する前にユーザが問題点に気づけるようにするよう努めること。ただし、このような作りとした場合も、サーバ側でのチェックを省いてはならないことに注意すること。
21. SUBMIT ボタンは、原則としてその FORM の末尾に設置すること。末尾に設置したうえで、中間や上部にも設置することは構わない。
22. 明確な必要性が認められない限り、RESET ボタンを FORM に配置しないこと。

アクセシビリティとユーザビリティ（検索）

1. 検索機能は、全角と半角の違いや、大文字と小文字の違いを利用者が意識せず利用できるよう、検索語、検索対象双方を正規化して処理すること。
2. 検索機能を実装する場合において、空白文字が検索キーに含まれていた場合は、全角半角問わずそれを区切り文字として単語を分割し、AND 検索とすることを原則とすること。また、オプションで OR 検索、フレーズ検索等を選択できるようにすることが望ましい。

アクセシビリティとユーザビリティ（拡張機能）

1. 不特定多数のユーザがアクセスするアプリケーション（特にインターネット向けのもの）については、原則として Web 標準技術（HTML、CSS、JavaScript）のみで利用可能なものとし、プラグイン、ActiveX コントロール、Java アプレット等、クライアントがバイナリプログラムをダウンロードし実行する仕組みは使用しないこと。標準技術以外のものを使用しなければ機能要件をどうしても実現できない場合や、使用しなければ著しくユーザビリティを損なう場合は、利用目的を明確にし、使用するという旨とその目的をあらかじめ明示すること。また、利用するプログラム等は、ブラウザが警告を表示しない物を使用すること。
2. JavaScript（その他のクライアントサイドスクリプトも含む）の利用は、ユーザの利便性向上やその他付加機能追加等にのみ利用し、それが機能しない、またはその機能を無効化されたブラウザでアクセスした場合、多少使い勝手が悪くなっても基本機能は問題なく使えるよう考慮した作りにすること。

3. 不特定多数の利用者向けサービスでも、高度なユーザビリティを実現するために JavaScript やプラグイン等の拡張機能の利用を必須とすることについて利用者の理解を得られると考えられる場合は上記の限りではないが、利用者向けの説明の中でその旨を明示するとともに、適切な代替手段を残す等の配慮を行うよう努めること。また、各拡張機能が使えない、または無効化されているブラウザでアクセスされている場合については、アプリケーション側でそれを判定し、有効化するための適切なアドバイスを表示すること。
4. 内部職員向け等、ユーザが特定される場合の場合は、高いユーザビリティを重視するために、想定するユーザのアクセシビリティを損なわない範囲で Ajax 等の RIA 技術を積極的に活用すること。
5. Cookie の利用は最小限に止めること。もし利用する場合は、利用者向けの説明の中でその旨と目的を明示すること。
6. 各種プラグイン、JavaScript、Cookie 等を利用する場合は、Web ページの作成基準 6-(2) に示す各ブラウザに加え、各ブラウザの 1 世代前のリリース版での動作確認を行い、正常動作することを確認すること。ただし、使用するユーザ環境が限定される場合においては、その想定されるユーザ環境のみで良いこととする。

検索エンジン対応

1. 市の全文検索システムや庁内の全文検索システム、その他インターネット上の Web ページ検索サービスでの検索でヒットした方が望ましいと思われるページを生成するプログラムについては、以下のように取り扱うこと。
    - プログラムへのパラメータ受け渡しには、サーバ内部での URL 書き換え機能（mod_rewrite）を活用し、URL の外見上に ？ が現れないようにすること。mod_rewrite の利用が困難な場合は、パラメータの受け渡しに PATH_INFO を使用し、QUERY_STRING は使用しないこと。逆に、検索で見つけてもらう意味のないページについては、このような対応を取らず、META 要素や robots.txt 等を用いてロボットによる収集を拒否すること。
    - レスポンスヘッダで Last-Modified 情報を返すこと。また、If-Modified-Since ヘッダにも対応し、適切に処理することが望ましい。
    - TITLE 要素に適切な内容が挿入されるよう特に気を遣うこと。
2. 利用者にとって意味のある情報が存在しないページにロボットがアクセスし続けることのないように、リンクが無限・または半永久的にループするようなことがないようにすること。カレンダー形式で過去・未来に辿ることができる画面については特に注意すること。

3. 情報が存在しない、削除された、有効期間が過ぎた情報を指す URL へのアクセス要求があった場合は、単に本文中でその旨を表示するだけでなく、HTTP レスポンスコード 404 や 410 を用いてそれが無効な URL であることを示すこと。
4. 同一の内容が異なる URL で表示されることをできる限り避けること。利用者の利便性向上その他特別な事情があり、複数の URL で同一コンテンツに案内することを可能としたい場合は、HTTP ステータスコード 301 を用いて 1 つの URL に転送するか、重複したコンテンツのページで META タグ等による検索エンジン避け設定および canonical 属性設定を活用し、最終的には 1 つのアドレスにアクセスが集約されるようにすること。

効率的な実装

サーバの負荷やユーザの快適性を保つため、以下の内容を守ること。

1. 小規模で簡易な物を除き、データ数が増えた場合に、それに比例してサーバ負荷が高くなるようなことがないよう考慮したデータ構造とアルゴリズムを採用すること。RDB を用いる場合は、各列に適切な型を選択、必要十分かつ最小限な範囲でインデックスを付加し、またインデックスが適切に使用されるようなクエリを用いるようにすること。
2. プログラムのロジックを記述するファイルと画面表示用テンプレートファイルとをできる限り分離した設計とし、軽易な表示内容の変更は、HTML を理解している人間ならプログラムを理解していなくても行えるようにすること。
3. 頻繁にアクセスされるページは、ユーザからのアクセス要求がある度に動的にページを生成するのではなく、あらかじめ静的な HTML ファイルとして吐き出しておくか、生成済みの HTML をキャッシュしておく等の仕組みとし、サーバ負荷軽減に努めること。
4. 動的にページを生成する場合も、Last-Modified や If-Modified-Since ヘッダを活用し、無駄なトラフィックを避けるよう努めること。

プログラム開発を外部委託する場合の注意点

1. 発注者は、委託先の候補として選定する業者に見積書の提出を依頼する際には、仕様書の付則要件として本作成基準を添付し、契約後はこれを遵守させること。
2. Web アプリケーションの作成を受託した者は、本作成基準に準拠した成果物を収めること。なお、Web ページを自動生成する Web アプリケーション（イベントカレンダー・掲示板・ブログ等広義の CMS 全般）については、Web アプリケーション自体が本作成基準に準じるだけでなく、それを用いて作成される Web ページが必然的に本作成基準に準拠したものになるよう、UI やテンプレートを設計しなければならないことに注意すること。

3. 基準の一部を満たすことが不可能または困難な場合、契約前（見積書提出前）に前提条件を発注主に書面で提示し、発注者の了解を得ること。
4. 契約後、業務進行のうえで、本作成基準を満たすことによって逆にアクセシビリティ、ユーザビリティを損なうと思われる状況や、極めて高コストになる、作成基準そのものに疑義がある、その他業務目的を達成するうえで本作成基準を満たすことが好ましくないと思われる状況が発生した場合は、受託者はその理由・根拠を明確にしたうえで発注主に報告し、その是非について、双方に記録の残る通信手段を利用して了承を得ること。なお、「開発が後期に差し掛かっており出戻りが大きいため」というような趣旨のものは高コストになる理由として認められない。
5. また、発注主からの指示内容が作成基準に反する内容であった場合も、受託者はその旨を説明し、発注主の意思を確認すること。
6. 発注主は、契約前・契約後にかかわらず、上記の報告を受けた場合は、利用者にとってのアクセシビリティ・ユーザビリティを最優先に考え、その是非について判断すること。判断が困難と思われるものは、区局の Web ページ担当課や、IT 活用推進課担当に適宜相談しながら進めること。
7. 上記の手順を踏まずに作成基準に反した成果物を収めたことが判明した場合、その影響が僅かなものであっても、契約時に設定した瑕疵担保期間において発注主は委託先に対して全ての修正を求めることができるものとする。なお、修正にあたり、納品後発注主によって設定やコンテンツ等の更新が行われていた場合、その更新後の内容を維持したまま修正版を作成すること。

※ここでの書面とは、双方に記録が残る媒体であれば、電子メールで構わないものとする。口頭でのやりとりがあった場合は、原則として受託者が議事録案を作成・提出し、発注主の了承を得ること。