

intra-mart WebPlatform/AppFramework Ver.7.2

アクセスブロック機能仕様書

2011/01/31 第2版

<< 変更履歴 >>

変更年月日	変更内容
2010/04/01	初版
2011/01/31	第 2 版 「4.2.4 遷移先ページ」にクライアントタイプに関する説明を追記しました。

<< 目次 >>

1	はじめに.....	2
1.1	目的.....	2
1.2	構成.....	2
2	リクエスト遮断定義.....	3
2.1	リクエスト遮断定義とは.....	3
2.2	リクエスト遮断定義の設定.....	3
2.2.1	リクエスト遮断定義.....	3
3	リクエスト検証定義.....	5
3.1	リクエスト検証定義とは.....	5
3.2	リクエスト検証定義の設定.....	5
3.2.1	リクエスト検証定義.....	5
4	リクエスト遮断ルール.....	8
4.1	リクエスト遮断ルールとは.....	8
4.2	リクエスト遮断ルールの設定.....	8
4.2.1	リクエスト遮断ルール.....	9
4.2.2	リクエスト遮断のマッピング.....	9
4.2.3	リクエスト検証のマッピング.....	9
4.2.4	遷移先ページ.....	9
4.3	アクセスブロックコンフィグファイルの分割.....	10
4.4	フィルタの設定.....	11
4.4.1	AccessBlockerFilterの定義.....	11
4.4.2	フィルタマッピング.....	11

1 はじめに

1.1 目的

本書は、intra-mart において、im-JavaEE Framework やマスカット、Struts 等のフレームワークを利用して開発されたアプリケーションへのリクエストの遮断やリクエストの検証を行う方法について説明する。

1.2 構成

本書は以下の構成でアクセスブロックを行う方法を述べる。

「2 リクエスト遮断定義」

「3 リクエスト検証定義」

「4 リクエスト遮断ルール」

2 リクエスト遮断定義

2.1 リクエスト遮断定義とは

リクエスト遮断定義とは、クライアントからのリクエストを判別し、遮断すべきリクエストであるかを決定するものである。リクエスト遮断定義は `jp.co.intra_mart.foundation.access_block.block.BlockPageDef` の実装クラスであり、デフォルトコンストラクタが存在しなければならない。

2.2 リクエスト遮断定義の設定

リクエスト遮断定義を使用するにはアクセスブロックコンフィグファイルに `BlockPageDef` インタフェースの実装クラスを記述する必要がある。アクセスブロックコンフィグファイルは `ServerManager` インストールディレクトリの以下のパスに配置されている。

- `/conf/access_block/access-block-config.xml`

リクエスト遮断定義は以下のように `<block-class>` に `BlockPageDef` インタフェースの実装クラス名と `<block-type>` に固有の ID を設定することで利用可能となる。

```
<?xml version="1.0" encoding="UTF-8"?>
<access-block-config>

  <block-def>
    <block-type>url</block-type>
    <block-class>jp.co.intra_mart.foundation.access_block.block.URLBlockPage</block-class>
  </block-def>

</access-block-config>
```

2.2.1 リクエスト遮断定義

`intra-mart` には標準で幾つかのリクエスト遮断定義が含まれている。ここではそのリクエスト遮断定義について説明する。

2.2.1.1 URLBlockPage

`jp.co.intra_mart.foundation.access_block.block.URLBlockPage` は URL パスを条件にしてリクエスト遮断を決定する。JSP や Struts を利用して作成されたアプリケーションに利用可能である。URLBlockPage には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
url	URL パスとファイル名を指定する。または URL パスとアスタリスクを指定。 <ul style="list-style-type: none"> ■ <code>aaa/bbb/sample.jsp</code> ■ <code>aaa/bbb/*</code> 	必須

ファイル名を指定する場合はそのファイルへの要求のみを遮断する。パスとアスタリスクを指定する場合はディレクトリとそのサブディレクトリ全てのファイルへの要求を遮断する。

URLBlockPage はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<block-def>
  <block-type>url</block-type>
  <block-class>jp.co.intra_mart.foundation.access_block.block.URLBlockPage</block-class>
</block-def>
```

2.2.1.2 JSSPBlockPage

jp.co.intra_mart.foundation.access_block.block.JSSPBlockPage は jssp を利用して作成されたアプリケーションへのリクエスト遮断を決定する。JSSPBlockPage には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
page	プログラムのパスとファイル名を拡張子無しで指定する。またはパスとアスタリスクを指定。 <ul style="list-style-type: none"> ■ aaa/bbb/sample ■ aaa/bbb/* 	必須

ファイル名を指定する場合はそのファイルへの要求のみを遮断する。パスとアスタリスクを指定する場合はディレクトリとそのサブディレクトリ全てのファイルへの要求を遮断する。

JSSPBlockPage はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<block-def>
  <block-type>jssp</block-type>
  <block-class>jp.co.intra_mart.foundation.access_block.block.JSSPBlockPage</block-class>
</block-def>
```

2.2.1.3 ServiceBlockPage

jp.co.intra_mart.foundation.access_block.block.ServiceBlockPage は im-JavaEE Framework を利用して作成されたアプリケーションへのリクエスト遮断を決定します。ServiceBlockPage には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
application	アプリケーション ID を指定。アスタリスク可	必須
service	サービス ID を指定。アスタリスク可	必須

アスタリスクをを指定した場合、各 ID の内容にかかわらず一致とみなされる。

ServiceBlockPage はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<block-def>
  <block-type>service</block-type>
  <block-class>jp.co.intra_mart.foundation.access_block.block.ServiceBlockPage</block-class>
</block-def>
```

2.2.1.4 MaskatBlockPage

jp.co.intra_mart.foundation.access_block.block.MaskatBlockPage はマスカットからのリクエスト遮断を決定します。

MaskatBlockPage には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
layout	レイアウト ID を指定。アスタリスク可	必須
component	コンポーネント ID を指定。アスタリスク可	必須
event	イベント ID を指定。アスタリスク可	必須

アスタリスクをを指定した場合、各 ID の内容にかかわらず一致とみなされる。

MaskatBlockPage はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<block-def>
  <block-type>maskat</block-type>
  <block-class>jp.co.intra_mart.foundation.access_block.block.MaskatBlockPage</block-class>
</block-def>
```


3 リクエスト検証定義

3.1 リクエスト検証定義とは

リクエスト検証定義とは、クライアントからのリクエストを判別し、関連付けられたリクエスト遮断の状態を操作するものである。リクエスト検証定義は `jp.co.intra_mart.foundation.access_block.matcher.MatcherDef` の実装クラスであり、デフォルトコンストラクタが存在しなければならない。

3.2 リクエスト検証定義の設定

リクエスト検証定義もリクエスト遮断定義と同様、アクセスブロックコンフィグファイルに記述する必要がある。リクエスト検証定義を以下のように `<matcher-class>` に `MatcherDef` インタフェースの実装クラス名と `<matcher-type>` に固有の ID を設定することで利用可能となる。

```
<?xml version="1.0" encoding="UTF-8"?>
<access-block-config>

  <matcher-def>
    <matcher-type>url</matcher-type>
    <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.URLMatcher</matcher-class>
  </matcher-def>

</access-block-config>
```

3.2.1 リクエスト検証定義

intra-mart には標準で幾つかのリクエスト検証定義が含まれている。ここではそのリクエスト検証定義について説明する。

3.2.1.1 MenuMatcher

`jp.co.intra_mart.foundation.access_block.matcher.MenuMatcher` は intra-mart イントラマートのメニューの選択時に発生するリクエストを検証する。ユーザが対象のメニューにアクセスする権限があれば、リクエスト遮断の状態を無効にする。一度無効にされたリクエスト遮断は、ログインセッションが続く限り有効になることはない。

MenuMatcher には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
menuId	intra-mart のメニューメンテナンスで登録されたメニューのメニューID。このメニューID のメニュー選択時に発生したリクエストを検証する。	必須
urlSuffix	メニュー選択時に発生するリクエスト URL の拡張子。デフォルトは <code>.menu</code> である。通常このパラメータを設定する必要はない。	

MenuMatcher はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<matcher-def>
  <matcher-type>menu</matcher-type>
  <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.MenuMatcher</matcher-class>
</matcher-def>
```

3.2.1.2 URLMatcher

jp.co.intra_mart.foundation.access_block.matcher.URLMatcher は URL のパスを指定してリクエストを検証する。JSP や Struts を利用して作成されたアプリケーションに利用可能である。リクエストされた URL パスとパラメータで設定された URL が一致する場合リクエスト遮断の状態を無効にする。一度無効にされたリクエスト遮断は、ログインセッションが続く限り有効になることはない。URLMatcher には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
url	URL パスとファイル名を指定する。または URL パスとアスタリスクを指定。 <ul style="list-style-type: none"> ■ aaa/bbb/sample.jsp ■ aaa/bbb/* 	必須

ファイル名を指定する場合はそのファイルへの要求のみを検証する。パスとアスタリスクを指定する場合はディレクトリとそのサブディレクトリ全てのファイルへの要求を検証する。

URLMatcher はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<matcher-def>
  <matcher-type>url</matcher-type>
  <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.URLMatcher</matcher-class>
</matcher-def>
```

3.2.1.3 JSSPMatcher

jp.co.intra_mart.foundation.access_block.matcher.JSSPMatcher は jssp を利用して作成されたアプリケーションへのリクエストを検証する。リクエストされた jssp のページパスとパラメータで設定されたページパスが一致する場合リクエスト遮断の状態を無効にする。一度無効にされたリクエスト遮断は、ログインセッションが続く限り有効になることはない。JSSPMatcher には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
page	プログラムのパスとファイル名を拡張子無しで指定する。またはパスとアスタリスクを指定。 <ul style="list-style-type: none"> ■ aaa/bbb/sample ■ aaa/bbb/* 	必須

ファイル名を指定する場合はそのファイルへの要求のみを検証する。パスとアスタリスクを指定する場合はディレクトリとそのサブディレクトリ全てのファイルへの要求を検証する。

JSSPMatcher はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<matcher-def>
  <matcher-type>jssp</matcher-type>
  <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.JSSPMatcher</matcher-class>
</matcher-def>
```

3.2.1.4 ServiceMatcher

jp.co.intra_mart.foundation.access_block.matcher.ServiceMatcher は im-JavaEE Framework を利用して作成されたアプリケーションへのリクエストを検証する。リクエストに含まれるアプリケーション ID とサービス ID がパラメータで設定された ID と一致する場合リクエスト遮断の状態を無効にする。一度無効にされたリクエスト遮断は、ログインセッションが続く限り有効になることはない。ServiceMatcher には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
application	アプリケーション ID を指定。アスタリスク可	必須
service	サービス ID を指定。アスタリスク可	必須

アスタリスクをを指定した場合、各 ID の内容にかかわらず一致とみなされる。

ServiceMatcher はデフォルトでアクセスブロックコンフィグファイルに設定されており、利用可能な状態である。

```
<matcher-def>
  <matcher-type>service</matcher-type>
  <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.ServiceMatcher</matcher-class>
</matcher-def>
```

3.2.1.5 MaskatMatcher

jp.co.intra_mart.foundation.access_block.matcher.MaskatMatcher はマスカットからのリクエストを検証する。リクエストに含まれるレイアウト ID、コンポーネント ID、イベント ID がパラメータで設定された ID と一致する場合リクエスト遮断の状態を無効にする。一度無効にされたリクエスト遮断は、ログインセッションが続く限り有効になることはない。MaskatMatcher には以下のパラメータが存在する。

パラメータ名	パラメータ値	必須
layout	レイアウト ID を指定。アスタリスク可	必須
component	コンポーネント ID を指定。アスタリスク可	必須
event	イベント ID を指定。アスタリスク可	必須

アスタリスクをを指定した場合、各 ID の内容にかかわらず一致とみなされる。

```
<matcher-def>
  <matcher-type>maskat</matcher-type>
  <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.MaskatMatcher</matcher-class>
</matcher-def>
```

4 リクエスト遮断ルール

4.1 リクエスト遮断ルールとは

リクエスト遮断ルールとはアクセスブロックコンフィグファイルに設定された、リクエスト遮断定義やリクエスト検証定義をマッピングし、遮断するリクエストのルール、遮断の有効、無効のルールを設定することである。

4.2 リクエスト遮断ルールの設定

リクエスト遮断ルールの設定はリクエスト遮断定義やリクエスト検証定義と同様にアクセスブロックコンフィグファイルに設定する。以下がその設定例である。

```
<?xml version="1.0" encoding="UTF-8"?>
<access-block-config>

  <block-def>
    <block-type>url</block-type>
    <block-class>jp.co.intra_mart.foundation.access_block.block.URLBlockPage</block-class>
  </block-def>

  <matcher-def>
    <matcher-type>menu</matcher-type>
    <matcher-class>jp.co.intra_mart.foundation.access_block.matcher.MenuMatcher</matcher-class>
  </matcher-def>

  <block-rule>
    <block-rule-id>rule1</block-rule-id>

    <block-mapping>
      <block-type>url</block-type>
      <param name="url" value="/aaa/bbb/ccc.jsp" />
    </block-mapping>

    <matcher-mapping>
      <matcher-type>menu</matcher-type>
      <param name="menuId" value="testMenu" />
    </matcher-mapping>

  </block-rule>

  <forward-page>system/security/error/access_block.jsp</forward-page>
</access-block-config>
```

4.2.1 リクエスト遮断ルール

リクエスト遮断ルールは<block-rule>を追加することで有効となる。

<block-rule>はひとつのアクセスブロックコンフィグに複数記述することが可能であり、<block-rule-id>には固有のIDを記述する必要がある。

4.2.2 リクエスト遮断のマッピング

<block-def>記述されたリクエスト遮断定義をリクエスト遮断ルールにマッピングする。<block-mapping>を追加し、利用したいリクエスト遮断定義に記述されている<block-type>を記述する。また、リクエスト遮断定義毎に必要なパラメータを<param>に記述する。マッピングされたリクエスト遮断定義が、クライアントからのリクエストやパラメータ値などの情報からアクセス遮断を決定した場合、リクエストは遮断され強制的に特定のページに遷移する。

ひとつのリクエスト遮断ルールに複数の<block-mapping>が設定可能である。

4.2.3 リクエスト検証のマッピング

<matcher-def>記述されたリクエスト検証定義をリクエスト遮断ルールにマッピングする。<matcher-mapping>を追加し、利用したいリクエスト検証定義に記述されている<matcher-type>を記述する。また、リクエスト検証定義毎に必要なパラメータを<param>に記述する。マッピングされたリクエスト検証定義は、クライアントからのリクエストやパラメータ値などの情報からアクセス遮断の状態を有効、無効に切り替える。

アクセス遮断の状態が無効に変更された場合、<block-rule>内に設定されている全ての<block-mapping>が機能しなくなり、クエスト遮断されることはない。

ひとつのリクエスト遮断ルールに複数の<matcher-mapping>が設定可能である。

4.2.4 遷移先ページ

リクエストが遮断された場合の遷移先ページを設定する。遷移先ページは<forward-page>に記述する。<forward-page>はひとつのアクセスブロックコンフィグにひとつだけ設定可能であるが、アクセス遮断ルール毎に遷移先ページを設定することが可能である。以下はその設定例である。また client-type 属性を記述することで、PC ブラウザや携帯電話等のクライアントの種類によって遷移先ページを切り替えることが可能である。

クライアントの種類に一致する client-type 属性が記述されていない場合は client-type 属性が記述されていないデフォルトの<forward-page>に遷移する。

```
<block-rule>
  <block-rule-id>rule2</block-rule-id>

  <block-mapping>
    <block-type>url</block-type>
    <param name="url" value="/aaa/bbb/ccc.jsp" />
  </block-mapping>

  <matcher-mapping>
    <matcher-type>menu</matcher-type>
    <param name="menuId" value="testMenu" />
  </matcher-mapping>

  <forward-page>test/access_block.jsp</forward-page>
  <forward-page client-type="mobile">test/access_block_mobile.jsp</forward-page>
</block-rule>
```

クライアントタイプ識別子	クライアントの種類
pc	PC ブラウザ
mobile	携帯電話

4.3 アクセスブロックコンフィグファイルの分割

リクエスト遮断ルールは複数のアクセスブロックコンフィグファイルに分割して記述することが可能である。以下の条件でファイルを配置することでアクセスブロックコンフィグファイルとして認識される。

- ServerManager がインストールされたディレクトリ配下の「/conf/access_block/」ディレクトリに存在するファイルである。
- 「access-block-config-」で始まるファイル名である。
- 拡張子が「.xml」のファイル名である。

例として以下のようなファイルを ServerManager のインストールされたディレクトリに配置することで、アクセスブロックコンフィグファイルとして認識される。

- /conf/access_block/access-block-config-test.xml

この条件で独自にアクセスブロックコンフィグファイルを作成した場合、access-block-config.xml に記述されているリクエスト遮断定義、リクエスト検証定義、遷移先ページは共通設定として認識され、独自に作成したアクセスブロックコンフィグファイルからも利用可能である。以下は独自に作成したアクセスブロックコンフィグファイルの設定例である。

```
<?xml version="1.0" encoding="UTF-8"?>
<access-block-config>

  <block-rule>
    <block-rule-id>rule3</block-rule-id>

    <block-mapping>
      <block-type>url</block-type>
      <param name="url" value="/aaa/bbb/ccc.jsp" />
    </block-mapping>

    <matcher-mapping>
      <matcher-type>menu</matcher-type>
      <param name="menuId" value="testMenu" />
    </matcher-mapping>

  </block-rule>

</access-block-config>
```

このアクセスブロックコンフィグファイルにはリクエスト遮断定義、リクエスト検証定義は設定されていないが、access-block-config.xml に設定されているものであるため利用可能である。また、遷移先ページも設定されていないため access-block-config.xml の設定されている遷移先ページが利用される。

4.4 フィルタの設定

4.4.1 AccessBlockerFilterの定義

リクエストの遮断、リクエストの検証を行うためには AccessBlockerFilter を有効にする必要がある。
intra-mart には標準で web.xml に AccessBlockerFilter が有効になっている。

```
<filter>
  <filter-name>AccessBlockerFilter</filter-name>
  <filter-class>jp.co.intra_mart.foundation.access_block.filter.AccessBlockerFilter</filter-class>
</filter>
```

4.4.2 フィルタマッピング

リクエストの遮断、リクエストの検証を行うためにサーブレットや URL に AccessBlockerFilter をマッピングしなければならぬ。以下はデフォルトのフィルタマッピングの一覧である。

```
<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <url-pattern>*.jsp</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>JSSPServlet</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>SecureJSSPServlet</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>JsspRpcServlet</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>ServiceServlet</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>action</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>MKJSServlet</servlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>AccessBlockerFilter</filter-name>
  <servlet-name>MKServiceServlet</servlet-name>
</filter-mapping>
```

これらのフィルタマッピングは intra-mart に標準で含まれているリクエスト遮断定義、リクエスト検証定義のために必要なフィルタマッピングである。独自にリクエスト遮断定義やリクエスト検証定義を作成する場合や、別のサーブレットでアクセスブロック機能を利用する場合は利用者がフィルタマッピングを追加する必要がある。

しかし、html や css、jpg などの Web サーバに配置される静的ファイルにはアクセスブロック機能は利用できない。以下に標準で設定されているフィルタマッピングとリクエスト遮断定義、リクエスト検証定義の関係を記す。

4.4.2.1 JSPファイルへのフィルタマッピング

JSP ファイルへのフィルタマッピングは以下のリクエスト遮断定義、リクエスト検証定義のために必要である。

- リクエスト遮断定義 - URLBlockPage
- リクエスト検証定義 - URLMatcher

4.4.2.2 Jsspへのフィルタマッピング

jssp が関係するサーブレットは以下のものである。

- JSSPServlet
- SecureJSSPServlet
- JsspRpcServlet

Jssp へのフィルタマッピングは以下のリクエスト遮断定義、リクエスト検証定義のために必要である。

- リクエスト遮断定義 - JSSPBlockPage
- リクエスト検証定義 - JSSPMatcher

4.4.2.3 im-JavaEE Frameworkへのフィルタマッピング

im-JavaEE Framework が関係するサーブレットは以下のものである。

- ServiceServlet

im-JavaEE Framework へのフィルタマッピングは以下のリクエスト遮断定義、リクエスト検証定義のために必要である。

- リクエスト遮断定義 - ServiceBlockPage
- リクエスト検証定義 - ServiceMatcher

4.4.2.4 メニュー選択時に発生するリクエストへのフィルタマッピング

メニュー選択時に発生するリクエストに関するサーブレットは以下のものである。

- MenuServlet

メニュー選択時に発生するリクエストへのフィルタマッピングは以下のリクエスト検証定義のために必要である。

- リクエスト検証定義 - MenuMatcher

4.4.2.5 Strutsへのフィルタマッピング

Struts が関係するサーブレットは以下のものである。

- ActionServlet

Struts へのフィルタマッピングは以下のリクエスト遮断定義、リクエスト検証定義のために必要である。

- リクエスト遮断定義 - URLBlockPage
- リクエスト検証定義 - URLMatcher

4.4.2.6 マスカットへのフィルタマッピング

マスカットが関係するサーブレットは以下のものである。

- MKJSServlet
- MKServiceServlet

マスカットへのフィルタマッピングは以下のリクエスト遮断定義、リクエスト検証定義のために必要である。

- リクエスト遮断定義 - MaskatBlockPage
- リクエスト検証定義 - MaskatMatcher

**intra-mart WebPlatform/AppFramework Ver. 7.2
アクセスブロック機能仕様書**

2011/01/31 第2版

**Copyright 2000-2011 株式会社NTTデータ イントラマート
All rights Reserved.**

TEL: 03-5549-2821

FAX: 03-5549-2816

E-MAIL: info@intra-mart.jp

URL: <http://www.intra-mart.jp/>