웹 서비스 시나리오
포함된 도움말 시스템 및 전자적으로 배포된 매체를 포함하는 이 문서(이하 "문서")는 정보 제공의 목적으로만 제공되며 CA 에 의해 언제든 변경 또는 취소될 수 있습니다.

CA 의 사전 서면 동의 없이 본문 문서의 전체 혹은 일부를 복사, 전송, 재생, 공개, 수정 또는 복제할 수 없습니다. 이 문서는 CA 의 기밀 및 독점 정보이며, 귀하는 이 문서를 공개하거나 다음에 의해 허용된 경우를 제외한 다른 용도로 사용할 수 없습니다: (i) 귀하가 이 문서와 관련된 CA 소프트웨어를 사용함에 있어 귀하와 CA 사이에 별도 동의가 있는 경우, 또는 (ii) 귀하가 CA 사이에 별도 기밀 유지 동의가 있는 경우.

상기 사항에도 불구하고, 본문 문서에 기술된 라이선스가 있는 사용자는 귀하 및 귀하 직원들의 해당 소프트웨어와 관련된 내부적인 사용을 위해 합당한 수의 문서 복사본을 인쇄 또는 재작할 수 있습니다. 단, 이 경우 각 복사본에는 전체 CA 저작권 정보와 범례가 첨부되어야 합니다.

본문 문서의 사본 인쇄 또는 재작 권한은 해당 소프트웨어의 라이선스가 전체 효력을 가지고 유효한 상태를 유지하는 기간으로 제한됩니다. 어떤 사유로 인해 라이선스가 종료되는 경우, 귀하는 사전으로 문서의 전체 또는 일부 복사본이 CA 에 반환되거나 파기되었음을 입증할 책임이 있습니다.

CA 는 관련법의 허용 범위 내에서, 상품성에 대한 묵시적 보증, 특정 목적에 대한 적합성 또는 권리 위반 보호를 비롯하여(이에 제한되지 않음) 어떤 종류의 보증 없이 본 문서를 "있는 그대로" 제공합니다. CA 는 본 시스템의 사용으로 인해 발생하는 직, 간접 손실이나 손해(수익의 손실, 사업 중단, 영업권 또는 데이터 손실 포함)에 대해서는(상기 손실이나 손해에 대해 사전에 명시적으로 동지를 받은 경우라도) 귀하가 제 3 자에게 책임을 지지 않습니다.

본문 문서에 언급된 모든 소프트웨어 제품의 사용 조건은 해당 라이선스 계약을 따르며 어떠한 경우에도 이 문서에서 언급된 조건에 의해 라이선스 계약이 수정되지 않습니다.

본문 문서는 CA 에서 제작되었습니다.

본 시스템은 “제한적 권리”와 함께 제공됩니다. 미합중국 정부에 의한 사용, 복제 또는 공개는 연방조달규정 (FAR) 제 12.212 조, 제 52.227-14 조, 제 52.227-19(c)(1)호 - 제(2)호 및 국방연방구매규정 (DFARS) 제 252.227-7014(b)(3)호 또는 해당하는 경우 후속 조항에 명시된 제한 사항을 따름니다.

Copyright © 2013 CA. All rights reserved. 본 시스템에서 언급된 모든 상표, 상호, 서비스 표시 및 로고는 각 해당 회사의 소유입니다.
CA Technologies 제품 참조

이 문서는 다음 CA Technologies 제품을 참조합니다:

- SiteMinder®
- CA SiteMinder for Secure Proxy Server

CA 에 문의

기술 지원팀에 문의

온라인 기술 지원 및 지사 목록, 기본 서비스 시간, 전화 번호에 대해서는 http://www.ca.com/worldwide에서 기술 지원팀에 문의하십시오.
설명서 변경 사항

이 설명서가 마지막으로 릴리스된 이후에 다음과 같이 업데이트되었습니다.

- 클라이언트 프로그램 생성 (페이지 12) - WADL 경로가 수정되었습니다. 이 변경 사항은 CQ171651 및 STAR Issue # 21406410:01을 해결합니다.
목차

제 1 장: SiteMinder 인증 및 권한 부여 웹 서비스

인증 및 권한 부여 웹 서비스로 작업하는 방법 ................................................................. 7
인증 및 권한 부여 웹 서비스 개요 ................................................................................ 8
웹 서비스 구성 .................................................................................................................. 9
웹 서비스에 대한 ACO 생성 .......................................................................................... 10
웹 서비스 보호 ................................................................................................................ 11
웹 서비스가 사용되도록 설정 ....................................................................................... 11
웹 서비스 로그 구성 ........................................................................................................ 12
클라이언트 프로그램 생성 ............................................................................................ 12
인증 SOAP 인터페이스 ............................................................................................... 13
인증 REST 인터페이스 ............................................................................................... 16
권한 부여 SOAP 서비스 ............................................................................................... 18
권한 부여 REST 인터페이스 ....................................................................................... 19
제 1 장: SiteMinder 인증 및 권한 부여 웹 서비스

이 섹션은 다음 항목을 포함하고 있습니다.
인증 및 권한 부여 웹 서비스로 작동하는 방법 (페이지 7)
인증 및 권한 부여 웹 서비스 개요 (페이지 8)
웹 서비스 구성 (페이지 9)
클라이언트 프로그램 생성 (페이지 12)

인증 및 권한 부여 웹 서비스로 작동하는 방법

SiteMinder 에서는 현재 인증 웹 서비스와 권한 부여 웹 서비스를 제공하고 있습니다. SiteMinder 인증 및 권한 부여 웹 서비스를 사용하는 프로세스에는 다음 다이어그램의 절차가 포함됩니다.

인증 및 권한 부여 웹 서비스에 대한 작업 방법
인증 및 권한 부여 웹 서비스 개요

인증 및 권한 부여 웹 서비스에 대한 작업을 하려면 다음 단계를 수행하십시오.

1. **ACO 를 생성합니다** (페이지 10).
2. **웹 서비스를 보호합니다** (페이지 11).
3. **웹 서비스가 사용되도록 설정합니다** (페이지 11).
4. **웹 서비스 로그를 구성합니다** (페이지 12).
5. **클라이언트 프로그램을 생성합니다** (페이지 12).

SiteMinder 인증 및 권한 부여 웹 서비스는 SPS(보안 프록시 서버) 설치에 포함됩니다. 각 구성 요소의 사용 여부를 개별적으로 설정할 수 있습니다.

웹 서비스 구성 프로세스에서는 다음과 같은 SiteMinder 개체가 구성되어 있다고 가정합니다.
- 호출자가 인증하는 대상 응용 프로그램을 보호하기 위한 에이전트 하나 이상
- 인증 및 권한 부여에 필요한 영역, 사용자 디렉터리, 정책 및 응답

인증 및 권한 부여 웹 서비스를 사용하면 다른 방법으로 보호되지 않는 응용 프로그램을 지원할 수 있습니다. 예를 들어 휴대폰의 독립형 응용 프로그램은 적절한 SiteMinder 개체가 있으면 사용자를 인증할 수 있습니다.

이러한 웹 서비스는 SOAP 1.2 프로토콜과 HTTP 기반 RESTful 아키텍처를 지원합니다. 인증 및 권한 부여 웹 서비스에서는 다음과 같은 기능을 제공합니다.
- 로그인(login) - 인증을 시도하고 인증이 성공하면 세션 토큰을 반환합니다.
  
  참고: 사용자 추적 사용 옵션이 사용되도록 설정되면 응답에 아이덴티티 토큰도 포함됩니다.
- 부울 로그인(blogin) - 인증을 시도하고 로그인 성공 여부를 확인합니다. 세션 토큰은 반환하지 않습니다.
- 로그아웃(logout) - 사용자 또는 사용자 그룹을 로그아웃합니다.
- 권한 부여(authorize) - 권한 부여 상태 메시지와 세로 고친 세션 토큰을 반환합니다.
웹 서비스 구성

작업 요청에 대한 응답은 SiteMinder 에서 생성한 헤더에 따라 다릅니다. 
리소스가 응답 인증 체계로 보호되는 경우 응답에 세션 토큰이 포함되지 
않고 아이덴티티 토큰이 포함됩니다. 이 아이덴티티 토큰을 이후 권한 부여 
요청에 세션 토큰 대신 사용할 수 있습니다.

인증 요청에는 다음 매개 변수가 포함됩니다.
- 응용 프로그램 ID(appId)
- 리소스 문자열(resource)
- 작업(action)
- 사용자 자격 증명

응용 프로그램 ID 는 SiteMinder 응용 프로그램 개체가 아니라 리소스 계층의 
위치에 대해 사용자가 정의한 논리적 이름을 나타냅니다. 응용 프로그램 
ID 는 내부적으로 에이전트에 매핑됩니다. SiteMinder 는 에이전트 이름을 
사용하여 영역을 확인합니다. 영역, 리소스 문자열 및 사용자 자격 증명으로 
사용자를 인증할 수 있습니다.

권한 부여 요청은 인증 요청보다 단순합니다. 권한 부여 요청에는 로그인 
응답에서 가져온 응용 프로그램 ID(appId), 리소스 경로, 작업 및 세션 토큰이 
포함됩니다. 웹 서비스는 토큰의 유효성을 검사한 후 지정된 리소스에 대한 
액세스 권한을 부여할지 여부를 결정합니다.

웹 서비스 구성

기본적으로 SPS 12.51 을 설치하거나 SPS 12.51 로 업그레이드하면 웹 서비스 
기능이 설치됩니다.

웹 서비스를 구성하려면 다음 단계를 수행하십시오.
1. WAMUI 를 통해 웹 서비스에 대한 ACO 를 생성합니다.
2. 웹 서비스를 보호합니다.
3. SPS 관리 UI 를 통해 웹 서비스가 사용되도록 설정합니다.
4. (선택 사항) 웹 서비스 로그를 구성합니다.
웹 서비스에 대한 ACO 생성

ACO 를 통해 웹 서비스를 관리할 수 있습니다. 웹 서비스를 사용하려면 enableauth 매개 변수나 enableaz 매개 변수, 또는 둘 모두를 활성화해야 합니다.

다음 단계를 수행합니다.

1. WAMUI 에서 AuthAzServiceDefaultSettings 템플릿을 기반으로 하는 ACO 를 생성합니다.
2. 다음 매개 변수를 구성합니다.

**AgentName**

리소스를 보호하는 웹 에이전트의 이름을 정의합니다. 각 웹 에이전트가 응용 프로그램을 보호하는 하나 이상의 웹 에이전트를 정의할 수 있습니다. 값은 다음 형식으로 입력하시십시오.

```
agent_name1,agent_appID1,agent_name2,agent_appID2,...,agent_nameN,agent_appIDN
```

**agent_name**

리소스를 보호하는 웹 에이전트의 이름을 정의합니다.

**appID**

`agent_name`에 지정된 웹 에이전트의 참조 이름 또는 웹 에이전트에 의해 보호되는 응용 프로그램의 참조 이름을 정의합니다. SiteMinder는 웹 서비스 요청에서 이 값을 사용하므로 사용자로부터 에이전트 이름을 보호합니다.

**enableauth**

인증 웹 서비스의 상태를 지정합니다. 인증 웹 서비스를 사용하려면 이 값을 yes 로 설정하십시오.

**enableaz**

권한 부여 웹 서비스의 상태를 지정합니다. 권한 부여 웹 서비스를 사용하려면 이 값을 yes 로 설정하십시오.

**RequireAgentEnforcement**

SiteMinder 에이전트가 웹 서비스를 보호해야 하는지 여부를 지정합니다. 프로덕션 환경에서는 SiteMinder 에이전트가 웹 서비스를 보호하도록 이 값을 yes 로 설정할 것을 강력히 권장합니다. 이 값을 yes 로 설정한 경우 웹 서비스가 보호되지 않으면 웹 서비스에 대한 요청이 실패합니다.
웹서비스 보호

프로덕션 환경에서는 웹 서비스를 보호할 것을 권장합니다. 웹 서비스의 웹 에이전트를 보호하면 사용자 요청이 처리되기 전에 SiteMinder 가 웹 서비스 클라이언트를 인증 및 권한 부여할 수 있습니다. 프로덕션 환경에서 웹 서비스를 보호하면 CA SiteMinder for Secure Proxy Server 가 사용자 요청에 SMSESSION 쿠키를 포함합니다. RequestSmSessionCookie ACO 매개 변수가 활성화된 경우 SiteMinder 는 웹 서비스가 사용자 요청을 처리하기 전에 SMSESSION 쿠키에 대한 사용자 요청을 확인하도록 합니다.

웹 서비스를 보호하기 위해 X.509 클라이언트 인증서 인증 체계를 사용하여 웹 서비스 루트 URL 을 보호하도록 CA SiteMinder for Secure Proxy Server 를 구성할 것을 권장합니다.

웹서비스가 사용되도록 설정

SPS 관리 UI 를 통해 이전 절차에서 생성한 ACO 를 사용하여 웹 서비스가 사용되도록 설정하십시오.

참고: enableauth 및 enableaz 의 값이 no 로 설정되어 있으면 SPS 관리 UI 를 통해 해당 지원 기능이 사용되도록 설정해도 웹 서비스가 작동하지 않습니다.

다음 단계를 수행하십시오.
1. "프록시 구성", "인증 및 권한 부여 웹 서비스"로 이동합니다.
2. "호스트 이름"에 웹 서비스가상 호스트의 고유 호스트 이름을 입력합니다.
3. "에이전트 구성 개체"에 웹 서비스에 대해 생성한 ACO 의 이름을 입력합니다.
4. "저장"을 클릭합니다.
 웹 서비스가 사용되도록 설정됩니다.
웹 서비스 로그 구성

웹 서비스가 사용되도록 설정하면 CA SiteMinder for Secure Proxy Server 에서 해당 웹 서비스의 로그를 server.log 파일에 저장합니다. 로그 위치를 server.log 에서 authazws.log 로 변경할 수도 있습니다.

로그 위치를 변경하려면 다음 단계를 수행하십시오.
1. sps_home/proxy-engine/conf/webservicesagent 로 이동합니다.
2. authaz-log4j.xml 파일을 백업합니다.
3. 원본 authaz-log4j.xml 파일을 열고 다음 단계를 수행합니다.
   a. 다음 AuthAZ.Rolling appenders 태그의 주석 처리를 제거합니다.

```
<appender name="AuthAZ.ROLLING"
   class="org.apache.log4j.DailyRollingFileAppender">
   <param name="File" value="logs/authazws.log"/>
   <layout class="org.apache.log4j.PatternLayout">
     <param name="ConversionPattern" value="%d [%5p] [%c] - [%m%n]"/>
   </layout>
</appender>
```

   b. AuthAZ.Rolling 태그에 대한 다음 appender-ref 를 모두 찾아 해당 주석 처리를 제거합니다.

```
<appender-ref ref="AuthAZ.ROLLING"/>
```

4. 변경 내용을 저장하고 CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

로그 위치가 sps_home/proxy-engine/logs/에 있는 authazws.log 파일로 변경됩니다.

클라이언트 프로그램 생성

클라이언트 프로그램은 다른 응용 프로그램 대신 웹 서비스에 대한 입증 및 권한 부여 요청을 발행하는 역할을 합니다. 클라이언트 프로그램에는 클라이언트 스탑에 대한 코드가 필요합니다. 스탑은 웹 서비스와의 통신을 위해 메시지를 관리하고 서로 주고받습니다. 웹 서비스는 WSDL 파일(SOAP 프로토콜의 경우)과 WADL 파일(REST 아키텍처의 경우)을 지원합니다. 웹 브라우저를 사용하여 WSDL 또는 WADL 파일에 액세스한 후 이를 XML 파일로 저장할 수 있습니다.
다음 단계를 수행하십시오.

1. 필요한 자격 증명을 수집하는 응용 프로그램용 비즈니스 논리를 작성합니다.

2. 클라이언트 스텝을 생성합니다. 필요한 경우 타사 도구와 함께 WSDL 또는 WADL 파일을 사용하여 클라이언트 스텝을 생성할 수 있습니다.
   - WSDL을 로드하려면 다음 URL을 사용하십시오:
     ```
     http://hostname/authazws/auth?wsdl
     ```
   - WADL을 로드하려면 다음 URL을 사용하십시오:
     ```
     http://hostname/authazws/AuthRestService/application.wadl
     ```

3. 클라이언트 스텝을 가져오고, 웹 서비스를 호출하는 스텝 개체를 인스턴스화합니다.

다음 단원에서는 참조를 위해 단순화된 샘플 SOAP 및 REST 메시지를 보여 줍니다.

인증 SOAP 인터페이스

이 단순화된 샘플에서는 SOAP 프로토콜을 사용한 인증 작업을 보여 줍니다. username, password 및 binaryCredentials라는 세 개의 필드만으로 구성된 IdentityContext로 대부분의 인증 체계를 지원할 수 있습니다. 다른 필드가 더 필요한 체계의 경우 자격 증명 유형에 맞게 입력이 조정된 추가 작업에서 지원됩니다.

다음 예제는 인증 웹 서비스의 일반적인 사용자 로그인 요청입니다.

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
  <s:Header/>
  <s:Body>
    <aut:login>
      <identityContext>
        <binaryCreds>
          <password>user1</password>
          <username>user1</username>
        </binaryCreds>
      </identityContext>
      <appId>app1</appId>
      <action>GET</action>
      <resource>*</resource>
    </aut:login>
  </s:Body>
</s:Envelope>
```
부울 로그인(blogin) 작업은 로그인(login) 작업과 유사하지만 다음 예제와 같이 blogin은 응답에 SMSESSION 값을 반환하지 않는다는 차이점이 있습니다.

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
            xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
    <s:Header/>
    <s:Body>
        <aut:blogin>
            <identityContext>
                <binaryCreds>
                </binaryCreds>
                <password>user1</password>
                <username>user1</username>
            </identityContext>
            <appId>app1</appId>
            <action>GET</action>
            <resource>/*</resource>
        </aut:blogin>
    </s:Body>
</s:Envelope>
```

다음 예제에서는 성공적인 로그인 응답을 나타냅니다.

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
    <s:Header/>
    <s:Body>
        <aut:loginResponse
            xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
            <return>
                <message>Authentication successful.</message>
                <resultCode>LOGIN_SUCCESS</resultCode>
                <sessionToken>session</sessionToken>
            </return>
        </aut:loginResponse>
    </s:Body>
</s:Envelope>
```
다음 예제에서는 실패한 로그인 시도를 나타냅니다.

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header/>
  <s:Body>
    <ns2:loginResponse xmlns:ns2="http://webservice.sm.services.soa.ca.com/">
      <return>
        <message>Authentication failed</message>
        <resultCode>LOGIN_FAILED</resultCode>
        <smSessionCookieValue/>
      </return>
    </ns2:loginResponse>
  </s:Body>
</s:Envelope>
```

다음 예제에서는 인증 웹 서비스 사용자 로그아웃 요청을 나타냅니다.

참고: 사용자가 성공적으로 로그아웃했더라도 해당 SessionToken 은 유효한 사용자 자격 증명으로 간주되므로 에이전트가 권한 부여에 이 SessionToken 을 계속 사용할 수 있습니다.

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header/>
  <s:Body>
    <aut:logout>
      <smSessionCookieValue>session</smSessionCookieValue>
    </aut:logout>
  </s:Body>
</s:Envelope>
```

다음 예제에서는 성공적인 인증 웹 서비스 로그아웃 응답을 나타냅니다.

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header/>
  <s:Body>
    <ns2:logoutResponse
        xmlns:ns2="http://ca.com/2010/04/15/authentication.xsd">
      <return>
        <message>Logout successful.</message>
        <resultCode>SUCCESS</resultCode>
      </return>
    </ns2:logoutResponse>
  </s:Body>
</s:Envelope>
```
인증 REST 인터페이스

REST는 REpresentational State Transfer를 의미합니다. REST에서는 서비스 요청이 개체의 상태를 URI를 통해 액세스할 수 있도록 변환합니다. HTTP의 경우 생성, 읽기, 업데이트 및 삭제 같은 작업을 통해 상태가 변경됩니다.

인증 및 권한 부여를 위한 URI 매핑은 appId와 resourcePath로 구성됩니다. 리소스 상태와 인증 또는 권한 부여된 사용자와 리소스를 연결한 것을 총칭합니다. 인증에 사용되는 서비스 이름은 login, blogin 및 logout입니다.

https://host:port#/login/appId/resourcePath/ 형식의 URI는 다음 요청을 포스트합니다.

```xml
<loginRequest>
  <binaryCreds/>
  <password>user1</password>
  <username>user1</username>
  <action>GET</action>
</loginRequest>
```

로그인 응답:

HTTP 반환 코드 200

```xml
<loginResponse>
  <message>Authentication successful</message>
  <resultCode>LOGIN_SUCCESS</resultCode>
  <smSessionToken>session</smSessionCookieValue>
  <authenticationResponses>
    <response>
      <name>SM_SESSIONDRIFT</name>
      <value>0</value>
    </response>
  </authenticationResponses>
</loginResponse>
```

HTTP 반환 코드 400

```xml
<loginResponse>
  <message>Bad Request</message>
  <resultCode>LOGIN_ERROR</resultCode>
</loginResponse>
```
HTTP 반환 코드 200

   <loginResponse>
   <message>Authentication Failed</message>
   <resultCode>LOGIN_FAILED</resultCode>
   <authenticationResponses>
      <response><name>SM_AUTHREASON</name><value>0</value>
      </response>
   </authenticationResponses>
   </loginResponse>

HTTP 반환 코드 500

   <loginResponse>
   <message>System</message>
   <resultCode>Server Error</resultCode>
   </loginResponse>

부울 로그인(blogin) 작업은 로그인(login)과 유사합니다. http://host:port#/blogin/appId/resourcePath 형식의 URI는 로그인 요청에 표시된 것과 같이 포스트하며, 응답 메시지에서 yes 또는 no를 반환합니다.

http://host:port#/logout/appId/resourcePath/ 형식의 URI는 다음과 같은 로그아웃 요청을 포스트합니다.

   <logoutRequest>
   <smSessionCookieValue>session</smSessionCookieValue>
   </logoutRequest>

인증 웹 서비스 로그아웃 응답:

   <logoutResponse>
   <message>logout successful(failed).</message>
   <resultCode>LOGOUT_SUCCESS</resultCode>
   <smSessionCookieValue>session</smSessionCookieValue>
   </logoutResponse>
권한 부여 SOAP 서비스

다음 XML은 웹 서비스에 대한 권한 부여 요청을 대략적으로 나타낸 것입니다.

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:aut="http://ca.com/2010/04/15/authorization.xsd">
    <soapenv:Header/>
    <soapenv:Body>
        <aut:authorize>
            <sessionToken>session</sessionToken>
            <appId></appId>
            <action>GET,POST</action>
            <resource>/domainAdmin/a.jsp</resource>
        </aut:authorize>
    </soapenv:Body>
</soapenv:Envelope>
```

다음 예제에서는 권한 부여 웹 서비스 AUTHORIZED 응답을 나타냅니다.

```xml
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Header/>
    <env:Body>
        <ns2:authorizeResponse
            xmlns:ns2="http://ca.com/2010/04/15/authorization.xsd">
            <return>
                <message>Authorization Successful</message>
                <resultCode>AUTHORIZED</resultCode>
                <sessionToken>aklaks</sessionToken>
                <authorizationResponses>
                    <response/>
                </authorizationResponses>
            </return>
        </ns2:authorizeResponse>
    </env:Body>
</env:Envelope>
```

다음 예제에서는 권한 부여 웹 서비스 UN AUTHORIZED 응답을 나타냅니다.

```xml
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Header/>
    <env:Body>
        <ns2:authorizeResponse
            xmlns:ns2="http://ca.com/2010/04/15/authorization.xsd">
            <return>
                <message>Authorization Failed</message>
                <resultCode>NOTAUTHORIZED</resultCode>
            </return>
        </ns2:authorizeResponse>
    </env:Body>
</env:Envelope>
```
참고: 유효한 세션 토큰을 포함한 권한 부여 웹 서비스 요청의 경우 NOTAUTHORIZED 권한 부여 응답에는 다음과 같은 제약 조건이 있습니다.

1. WAMUI 에서 다음 특성으로만 응답을 구성할 수 있습니다.
   ■ SM_ONREJECTTEXT
   ■ SMREDIRECTURL 또는 SM_REDIRECTURL
   ■ SMERROR
2. 응답에는 세션 토큰이 포함되지 않습니다.

권한 부여 REST 인터페이스
권한 부여를 위한 REST 인터페이스는
https://fqservername:port/AuthZ/appId/resourcePath/ 입니다.
<authorizationRequest>
  <action>POST</action>
  <resource>RealmA/index.html</resource>
  <sessionToken>affl;;alkf;l;fd</sessionToken>
</authorizationRequest>

HTTP 반환 코드 200:
< authorizationResult >
  <message>The user is authorized.</message>
  <resultCode>AUTHORIZED</resultCode>
</authorizationResult>