

16. AUD SSO ?

1. Application Access Token

AUD i-PORTAL

AUD Rest API

2. Application Access Token

1) AUD Repository DB MTX_AUTH_AP_INFO ()

Create Table(PostgreSQL)

```
-- [Admin > Repository ] Repository Script      Database
CREATE TABLE MTX_AUTH_AP_INFO (
  APP_ID VARCHAR(50) NOT NULL,
  APP_NAME VARCHAR(50) NULL,
  APP_SSH_PUB_KEY VARCHAR(2000) NULL,
  APP_SECRET_KEY VARCHAR(255) NOT NULL,
  CREATE_DATE TIMESTAMP NOT NULL,
  EXPIRE_DATE TIMESTAMP NULL,
  CONSTRAINT MTX_AUTH_AP_INFO_UNIQUE UNIQUE (APP_ID)
);
CREATE INDEX MTX_AUTH_AP_INFO_APP_ID_IDX ON MTX_AUTH_AP_INFO USING BTREE (APP_ID);
```

2) AUD Domain name (Application Access Token .)

3. Application Access Token

1) SSH

 SSO AUD

1. AUD7 Admin

☒ > () .

AUD Platform Admin

개정관리

그룹관리

부서관리

사용자관리

그룹별 사용자관리

자원관리

폴더/파일관리

변주관리

서버관리

데이터베이스관리

템플릿관리

공통데이터 소스관리

권한관리

각제별 권한 조회 수정

주제별 권한 조회 수정

사용자별 권한 보기

권한복제

제용별 버튼 관리

보고서 버튼 그룹 관리

버튼 권한 관리

시스템 관리

시스템 옵션

라이선스 관리

시스템 로그 보기

시스템 정보조회

I-OLAP 설정

FTP정보 관리

반응형 설정

Schema Manager 관리

지도 컴포넌트 관리

이클릭에이션 관리

시스템 운영관리

개정관리 > 사용자관리

가져오기

전체 내보내기

이동

신규

삭제

차량

부서트리	계정	이름	비밀번호	부서	메일 주소	로그인 실패 횟수	잠금상태	사용종지	설정	DB User	DB Password	정...
전사	matrix	손정훈	*****	전사		0	☐	☐				
백연도팀	matrix1	matrix1	*****	전사		3	☒	☐				
프로토타입팀	matrix2	matrix2	*****	전사		0	☐	☐				
	matrix3	matrix3	*****	백연도팀		0	☐	☐				
	matrix4	matrix4	*****	백연도팀		0	☐	☐				
	matrixadm	Administrator	*****	전사		0	☐	☐		matrixadm	*****	0
	songj	songj	*****	전사		0	☐	☐				
	songj2	songj2	*****	전사			☐	☐				
	Songj2	Songj2	*****	전사			☐	☐				
	N			DEFAULT			☐	☐				

총 9 개

< 1 / 1 >

페이지 1

☒
>
>
'

AUD Platform Admin

관리자관리

공통데이터 소스관리

권한관리

각제별 권한 조회 수정

주제별 권한 조회 수정

사용자별 권한 보기

권한복제

제용별 버튼 관리

보고서 버튼 그룹 관리

버튼 권한 관리

시스템 관리

시스템 옵션

라이선스 관리

시스템 로그 보기

시스템 정보조회

I-OLAP 설정

FTP정보 관리

반응형 설정

Schema Manager 관리

지도 컴포넌트 관리

인증 키 관리

시스템 운영관리

관리자 메뉴 설정

제용 별 옵션 설정

모듈 관리자

모듈별주관리

PORTAL 설정

Repository 데이터점 점검

서버 구간별 응답시간

실행 프로세스 관리

백업/복원

보고서 백업/복원

시스템 운영관리 > 관리자 메뉴 설정

종류	보고서명	사용여부	관리자 보고서 목록	보고서 코드
G-MATRIX 관리	공통 코드 관리	☐		GTX_000
	학습 데이터 관리	☐		GTX_120
	메타 학습 관리	☐		GTX_140
	피드백 관리	☐		GTX_170
	로그 관리	☐		GTX_180
	추천 질의 관리	☐		GTX_220
	도메인 학습 관리	☐		GTX_230
	외도 학습 관리	☐		GTX_240
	학습 데이터 백업/복원	☐		GTX_250
	업무 영역 관리	☐		GTX_260
메타관리	메타보고서 관리	☒		MTA_110
	메타출산 관리	☒		MTA_120
	메타 수동 업데이트	☒		MTA_130
시스템 운영관리	서버 구간별 응답시간	☒		OPE_130
	실행 프로세스 관리	☒		OPE_280
스케줄 관리	I-MATRIX 스케줄 관리	☒		SCH_133
	I-AUD 스케줄 관리	☒		SCH_152
시스템관리	작업 스케줄 관리	☒		SCH_132
	FTP정보 관리	☒		SYS_150
	반응형 설정	☒		RPL_100
	Schema Manager 관리	☒		OPE_260
	지도 컴포넌트 관리	☒		GEO_110
	인증 키 관리	☒		SYS_180

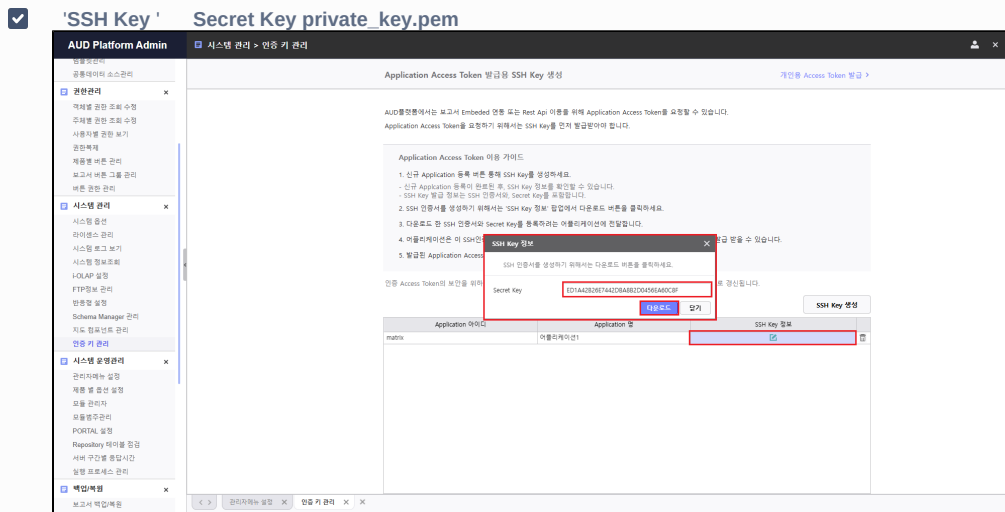
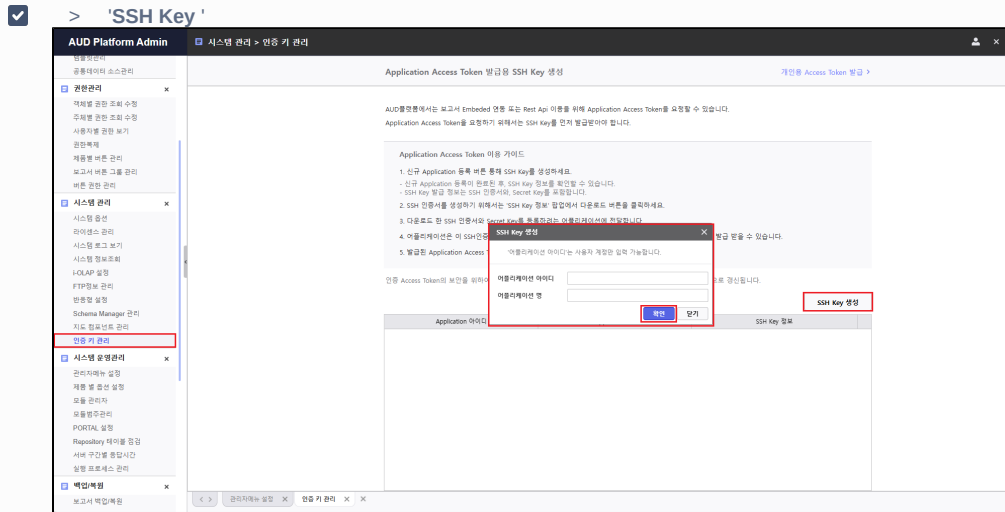
< >

관리자 메뉴 설정

X

X

Application Access Token SSH Key



Secret Key Private.pem

2. AUD CORS Allowed origin url

- AUD /WEB-INF/classes/framework/service/service_api.properties
- matrix.prop.location=true matrix.prop.location.path service_api.properties

- Header Origins URL ()

- matrix.cross.origin.allowed PORTAL URL

AUD /WEB-INF/classes/framework/service/service_api.properties

```
# cross origin url ( )
matrix.cross.origin.allowed=http://1.1.1.1,https://2222:8080/XXX,http://a.co.kr
```

- service_api.properties AUD

3. CORS

- sitePortalXXX.jsp JSP

- JSP AP . AP Java
- AUD7 private key Secret Key Header Key AUD7 API .
- AUD 1 Application User Secret key .

✓ /aud/aud.embedded.setting.jsp

- AUD_CONFIG_DATA.bimatrix_server_url = {AUD7 . Protocol + DNSNAME }
- AUD_CONFIG_DATA.webRoot = { webRoot . aud }
- AUD_CONFIG_DATA.cookie_domain = {Ap }
- AUD_CONFIG_DATA.button_disable_ids = {title }

- AP PORTAL () AUD7 PORTAL Domain name .
- POTAL AP Cookie bimatrix_ap_accessToken .
- AP SSH key AUD7 Rest API Private.pem Secret Key ()

Application AP (bimatrix_ap_accessToken) AUD Rest API .

AUD PORTAL .

- AUD CORS Application CORS .



AP API

1. AUD Application Application JWT

- .
- 1 .
- . (key = bimatrix_ap_accessToken)

URL		Header			
{AUD }/api/auth/sign/ap/token	POST	API Request Header		String	AUD SSH Private.pem Secret Key Header Secret Key
		Key			AUD .
		JWT	AUD		
		X-AUD-AP-Id			
		X-AUD-AP-Secret-SSH	AUD7 ssh private pem Key		
		X-AUD-USER	- Application - AUD id		
X-AP-UPDATE-ADDR	() ,				

Application API .

> sitePortalAUD7Em_iAudSample.jsp

```
/**
 * AUD AP Token JAVA
 */
public class AudGetApToken {
    private static final String AUD_AP_TOKEN_URL = "aud7 /matrix/api/auth/sign/ap/token";
```

```

/**
 * audApId , audApPw AUD Application
 * Application id / pw
 */
public static String getAPToken(String audApId , String audApSecretKey , String userCode) {
    HttpURLConnection connection = null;
    // aud jwt .
    String AUD_AP_TOKEN_UPDATE_URL = "bimatrix.co.kr";

    try{
        if ((audApId == null || audApId.isEmpty()) || (audApSecretKey == null || audApSecretKey.
isEmpty())){
            System.out.println("AUD application id,pw .");
            return null ;
        }

        // aud7 secret key ssh private key .
        PrivateKey privateKey = loadPrivateKey("D:\\tomcat\\apache-tomcat-7.0.107-
windows-x64\\apache-tomcat-7.0.107\\webapps\\examples\\aud\\apSSH\\private_key.pem");
        // aud7 secret key
        String signedMessage = signMessage(audSecretKey, privateKey);

        // header .
        Map<String, String> requestHeaders = new HashMap();
        requestHeaders.put("X-AUD-AP-Id", audApId);
        requestHeaders.put("X-AUD-AP-Secret-SSH", signedMessage);
        requestHeaders.put("X-AP-UPDATE-ADDR" , AUD_AP_TOKEN_UPDATE_URL);
        if (userCode != null || !userCode.isEmpty())
            requestHeaders.put("X-AUD-USER" , userCode);

        try{
            // URL
            URL url = new URL(AUD_AP_TOKEN_URL);
            connection = (HttpURLConnection) url.openConnection();
        }catch(MalformedURLException e){
            System.out.println("AUD .");
            return null;
        }catch(IOException e){
            System.out.println(" [api url: "+ AUD_AP_TOKEN_URL +"]");
            return null ;
        }

        // HTTP
        connection.setRequestMethod("POST");
        for(Map.Entry<String, String> header :requestHeaders.entrySet()) {
            connection.setRequestProperty(header.getKey(), header.getValue());
        }
        //
        int responseCode = connection.getResponseCode();

        System.out.println("Response Code: " + responseCode);
        if (responseCode == HttpURLConnection.HTTP_OK){
            String apAccessToken = connection.getHeaderField("bimatrix_ap_accessToken");
            if (apAccessToken == null){
                System.out.println("ap token ");
                return null;
            }
            else{
                return apAccessToken;
            }
        }else if (responseCode == HttpURLConnection.HTTP_UNAUTHORIZED){
            System.out.println(" ap token .");
            return null;
        }

    }catch(Exception e){
        e.printStackTrace();
    }finally {
        if (connection != null) {
            connection.disconnect();
        }
    }
}

```

```

    }

    return null;
}

//
private static PrivateKey loadPrivateKey(String path) throws Exception {
    //      (    )
    System.out.println("Private Key path: " + path);
    String keyPEM = new String(Files.readAllBytes(Paths.get(path)))
        .replaceAll("-----BEGIN PRIVATE KEY-----", "")
        .replaceAll("-----END PRIVATE KEY-----", "")
        .replaceAll("\\s", ""); //

    byte[] keyBytes = Base64.getDecoder().decode(keyPEM);
    PKCS8EncodedKeySpec keySpec = new PKCS8EncodedKeySpec(keyBytes);
    KeyFactory keyFactory = KeyFactory.getInstance("RSA");
    return keyFactory.generatePrivate(keySpec);
}

//
private static String signMessage(String message, PrivateKey privateKey) throws Exception {
    Signature signature = Signature.getInstance("SHA256withRSA");
    signature.initSign(privateKey);
    signature.update(message.getBytes());

    byte[] signedBytes = signature.sign();
    return Base64.getEncoder().encodeToString(signedBytes);
}
}

```

2. AUD JWT

.



✓ Rest API Header api



API CORS API
ex) fetch

```
fetch(biServerUrl + apiUrl, {
  method: 'POST', // 'POST', 'GET'
  headers: {
    'Content-Type': 'application/json',
    'bimatrix_ap_accessToken': ' AUD JWT '
  },
  credentials: 'include', //
  body: JSON.stringify(params)
})
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    return response.json(); // JSON
  })
  .then(data => {
    alert(JSON.stringify(data));
  })
  .catch(error => {
    console.error('There has been a problem with your fetch operation:',
error);
    alert(' .');
  });
```



(i-AUD) Sample



i-AUD_Em_20250211.zip

