

**API 連携機能ガイド**  
**-APICollaborationGuide-**

**株式会社野村総合研究所**

**Nomura Research Institute, Ltd.**

- 本書は、Senju Service Manager システムバージョン 2023.0.6 の API 連携機能について説明します。なお、万一ご不明な点や記載誤り・漏れなど、お気づきの点がございましたら弊社までお知らせ下さい。
- Senju Service Manager システムバージョン 2023.0.6 の API 連携機能に対応するプロダクトのバージョンについては、Senju Service Manager のリリースノートの稼働環境を参照してください。
- 本書は、Senju Service Manager システムをインストールまたは利用する前に一読して下さい。なお、万一ご不明な点や記載誤り・漏れなど、お気づきの点がございましたら弊社までお知らせ下さい。
- 本書に記載した内容は予告無く変更することがあります。
- 本書の内容の一部または全部を無断でコピーすることは法律で禁止されています。
- Senju Service Manager は、株式会社野村総合研究所の登録商標です。  
Adobe 及び Acrobat は、Adobe Systems Incorporated(アドビ システムズ社)の商標です。  
Microsoft Windows は、米国 Microsoft Corporation の米国及び他の国における登録商標です。  
ORACLE は、米国 Oracle Corporation の登録商標です。  
Oracle Developer/2000 は、米 Oracle Corporation の登録商標です。  
Oracle Applications は、オラクル社の商標です。  
UNIX は、The Open Group の米国ならびに他の国における登録商標です。  
Intel および Pentium は、Intel Corporation の登録商標です。  
iPhone は Apple Inc.の登録商標です。  
PostgreSQL は、PostgreSQL の米国およびその他の国における商標または登録商標です。  
ANACONDA は、ANACONDA, INC.の登録商標です。  
Python は、Python Software Foundation の登録商標です。  
Apache は、The Apache Software Foundation の登録商標または商標です。  
その他のすべての会社名や製品名は、それぞれの会社の商標、登録商標または、サービスマークです。
- 本書では、便宜上 Microsoft 社の Windows ファミリーを Windows と表記しています。  
また、特に断りが無い場合、Windows NT とは”Windows NT Workstation”, ”Windows NT Server”を、Windows 2000 とは”Windows 2000 Professional”, ”Windows 2000 Server”, ”Windows 2000 Advanced Server”を、Windows XP とは”Windows XP Professional”を、Windows Server 2003 とは”Windows Server 2003 Standard Edition”, ”Windows Server 2003 Enterprise Edition”を、Windows Vista とは”Windows Vista Ultimate”, ”Windows Vista Business”を、Windows 7 とは”Windows 7 Ultimate”, ”Windows 7 Professional”を、Windows 8.1 とは”Windows 8.1 Pro”を、Windows 10 とは”Windows 10 Pro”を、Windows Server 2008 とは”Windows Server 2008 Standard Edition”, ”Windows Server 2008 Enterprise Edition”を、Windows Server 2012 とは”Microsoft Windows Server 2012 Standard Edition”を、Windows Server 2016 とは”Microsoft Windows Server 2016 Standard Edition”を、Windows Server 2019 とは”Microsoft Windows Server 2019 Standard Edition” , ”Microsoft Windows Server 2019 Datacenter Edition” を、Windows Server 2022 とは”Microsoft Windows Server 2022 Standard Edition” , ”Microsoft Windows Server 2022 Datacenter Edition” , ”Microsoft Windows Server 2022 Datacenter: Azure Edition” を指します。
- 本書では、便宜上、Senju Service Manager を SSM、Senju Operation Conductor を SOC、Senju Enterprise Navigator を SEN と表記している箇所があります。また、Senju DevOperation Conductor と Senju Operation Conductor を合わせて Senju Operation Conductor と表記しています。

発行日 2024年 3月 29日

第 1.0.8 版

著作、編集、発行

**株式会社野村総合研究所**

お問合せ先

マルチクラウドインテグレーション事業本部

クラウド運用ソリューション事業部

〒220-0012 横浜市西区みなとみらい 4-4-1 横浜野村ビル

Copyright © Nomura Research Institute, Ltd.

本マニュアルの一部又は全部を無断で複製する事を禁じます。

Senju Service Manager 2023.0.6 .....	1-1
<b>1 API 連携機能ガイド .....</b>	<b>1-4</b>
1.1 API 連携機能の概要 .....	1-4
1.2 稼働環境.....	1-5
1.2.1 API サーバー環境 .....	1-5
1.2.2 その他の環境.....	1-6
1.2.3 通信ポート .....	1-7
1.2.4 設定項目一覧.....	1-8
1.3 システム構成図.....	1-9
1.4 事前準備.....	1-10
1.4.1 Python の導入.....	1-10
1. 対象バージョン .....	1-10
2. Python のインストール .....	1-10
3. 環境変数の設定 .....	1-12
4. 仮想環境の作成.....	1-17
5. Python パッケージのインストール .....	1-18
1.4.2 Apache の導入 .....	1-23
1. 対象バージョン .....	1-23
2. Microsoft VC++ 再頒布可能パッケージのインストール .....	1-23
3. Apache のインストール .....	1-24
4. Apache の設定(SSL 通信の設定).....	1-27
1.5 API 連携機能の導入 .....	1-28
1.5.1 WEB アプリケーションのインストール .....	1-28
1. WEB アプリケーションのインストール .....	1-28
1.5.2 WEB アプリケーションの設定.....	1-30
1. 資材の格納 .....	1-30
2. WEB アプリケーションの設定変更.....	1-30
1.5.3 WEB サーバーと WEB アプリケーションの接続設定 .....	1-33
1. WEB サーバーの WSGI 対応.....	1-33
2. WEB サーバー起動時に実行される WEB アプリケーションの起動スクリプト登録.....	1-34
3. WEB サーバーの再起動.....	1-34
1.5.4 ウィルススキャンの除外.....	1-35
1.5.5 WEBAPI の複数導入設定 .....	1-35
1. 追加 WEBAPI のインストール.....	1-35
2. 追加 Apache のインストール.....	1-35
3. 追加 WEBAPI と Apache の接続設定 .....	1-37
1.6 API 連携機能 利用ガイド.....	1-37
1.7 バージョンアップ .....	1-38
1.7.1 2023.0.0 へバージョンアップ .....	1-38
1.7.1.1 2022.0.0.0 環境のアンインストール .....	1-38
1.7.1.2 2023.0.0 API 連携機能の導入 .....	1-38
1.7.2 Apache のバージョンアップ.....	1-38
1.8 制限事項.....	1-41

# 1 API 連携機能ガイド

## 1.1 API連携機能の概要

Senju Service Manager では、Senju Service Manager のデータにアクセス可能な WEBAPI を提供します。

Senju Service Manager のデータにアクセス可能な WEBAPI を API 連携機能と表現します。

API 連携機能は、WEB サーバー Apache と Python で作成された WEB アプリケーションで構成されます。



補足説明

本マニュアルでは、WEBAPI という用語を以下のように定義します。  
「HTTP プロトコルを利用してネットワーク越しに呼び出す API」

## 1.2 稼働環境

Senju Service Manager で API 連携機能を利用するにあたり、Apache、Python をインストールするサーバーの構成と、インストールする製品、バージョンを確認してください。



仕様補足

本機能は 2021.0.0.0 以降でご利用いただけます。

### 1.2.1 API サーバー環境

導入に必要となる構成は以下の通りです。

構成種別	構成詳細
WEB サーバー	Apache
Python	Python



制限事項

SSMWEB サーバーがインストールされたノードにのみ構築することができます。



仕様補足

詳しいサポートバージョンについては「リリースノート」を参照してください。

## 1.2.2 その他の環境

---

クライアント、サーバー環境については、リリースノートを参照してください。

## 1.2.3 通信ポート

---

ファイアウォールサービスについて、停止するか、「リリースノート」のポート情報を参考に該当するポートを開放して下さい。

## 1.2.4 設定項目一覧

Oracle Database を利用する場合、API 連携機能のインストールで必要になる設定項目を決定します。

設定項目	マニュアルでの設定例	お客様の設定値
API サーバー・ホスト名※1	ssmapihost	
SSM WEB サーバー・ホスト名※2	ssmwebhost	
Oracle Database ホスト名※3	oracledbhost	
データベース名	ssmdb	
DB ユーザー名	ssmuser	
DB ユーザーパスワード	ssmpwd	
リスナーポート番号	1522	
API ポート番号※4	49152	

※1 <APIサーバーホスト名>は、API 連携機能をインストールするノードを指定します。

※2 <SSM WEB サーバーホスト名>は、SSM WEB モジュールをインストールするノードを指定します。

※3 <Oracle Database ホスト名>は、Oracle Database をインストールするノードを指定します。

※4 構築したい API のポート番号を 49152～65535 の間で設定してください。

PostgreSQL を利用する場合、API 連携機能のインストールで必要になる設定項目を決定します。

設定項目	マニュアルでの設定例	お客様の設定値
API サーバー・ホスト名※1	ssmapihost	
SSM WEB サーバー・ホスト名※2	ssmwebhost	
データベース名	ssmdb	
スーパーユーザー	postgres	
DB パスワード※3	123456	
DB ユーザー名	ssmuser	
DB ユーザーパスワード	ssmpwd	
DB ポート番号	5432	
API ポート番号※4	49152	

※1 <APIサーバーホスト名>は、API 連携機能をインストールするノードを指定します。

※2 <SSM WEB サーバーホスト名>は、SSM WEB モジュールをインストールするノードを指定します。

※3 <DB パスワード>は、PostgreSQL の postgres ユーザーのパスワードを指定してください。

※4 構築したい API のポート番号を 49152～65535 の間で設定してください。



マニュアル中「<ホスト名>」のように記載している場合は、お客様の設定値をご入力いただくことを示します。このような記載がない場合は、マニュアルでの設定例と同じものをご入力いただくことをお勧めします。

「お客様の設定値」欄は、備忘のためのメモのスペースとしてご利用ください。



DB ユーザーパスワードには 28 桁以下の値を指定してください。

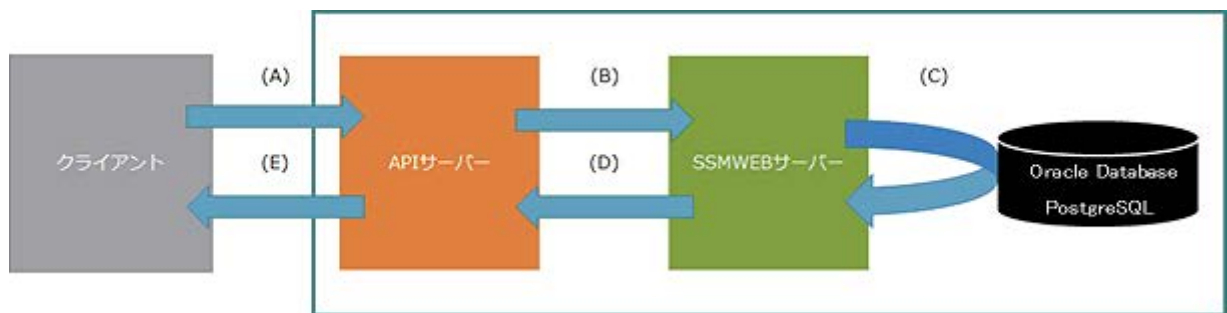


## 1.3 システム構成図

クライアント、API サーバー、SSM WEB サーバー、Oracle Database(PostgreSQL)間の簡単なシステム構成図を示します。

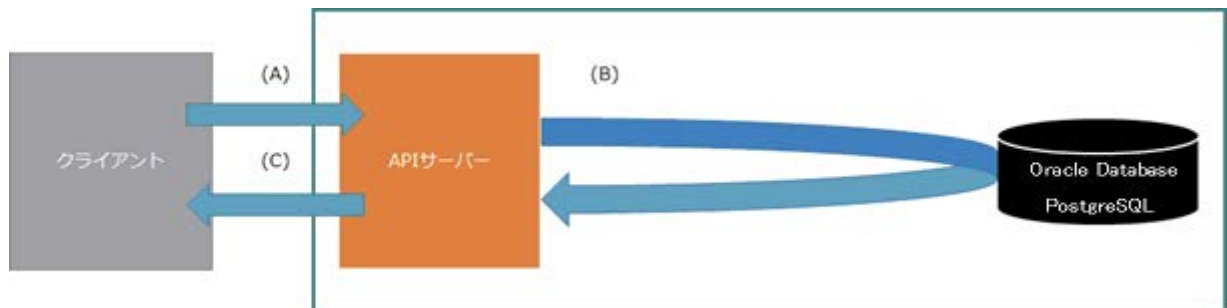
認証処理

通信	説明
(A)	ユーザーは API サーバーに対して Senju Service Manager の認証情報を送信し、アクセストークン発行を要求します。
(B)	API サーバーは SSMWEB サーバーに対して Senju Service Manager の認証情報を送信し、認証を依頼します。
(C)	SSMWEB サーバーは Oracle Database もしくは PostgreSQL を参照し、送信された認証情報で認証処理を行います。
(D)	認証に成功した場合、SSMWEB サーバーは API サーバーに対し認証成功のレスポンスを返します。
(E)	SSMWEB サーバーから認証成功のレスポンスが返った場合、API サーバーはアクセストークン・リフレッシュトークンを生成しユーザーに送信します。



Senju Service Manager の API 実行処理

通信	説明
(A)	ユーザーは API サーバーに対して認証処理で発行されたアクセストークンを送信し、Senju Service Manager の API 実行を要求します。
(B)	API サーバーは Oracle Database もしくは PostgreSQL にアクセスし、Senju Service Manager の API 処理を行います。
(C)	API サーバーはユーザーに対して Senju Service Manager の API 実行結果を返却します。



## 1.4 事前準備

### 1.4.1 Python の導入

Senju Service Manager と連携する API サーバーの WEB アプリケーションは Python で作成されています。

WEB アプリケーションの稼働に必要なソフトウェアである Python をインストールするための手順について説明します。

#### 1. 対象バージョン

Senju Service Manager と連携する API サーバーでは Windows プラットフォームの 64 ビット Python を使用します。

ここでは、Python3.11.3 を例として導入手順を説明します。



仕様補足

サポート対象となる Python のバージョンについては リリースノート を参照してください。



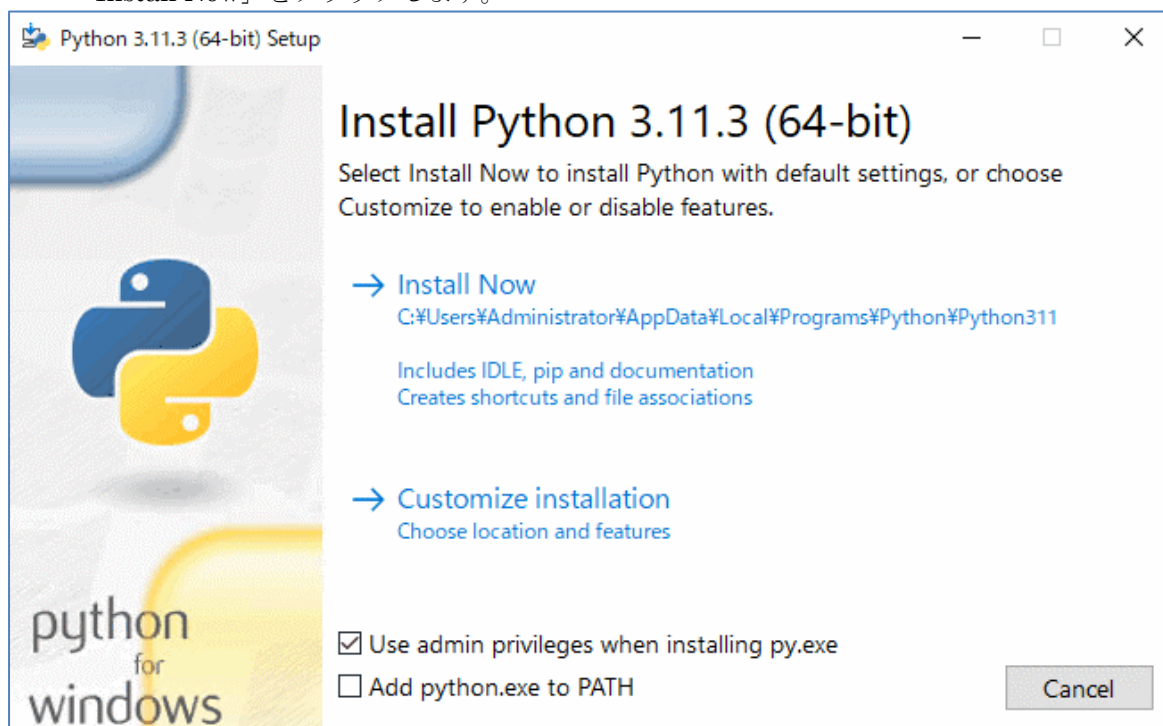
仕様補足

例として手順を説明するバージョンは開発時点のバージョンになります。インストール時には、セキュリティ上の観点を考慮してその時点での最適なバージョンを導入してください。

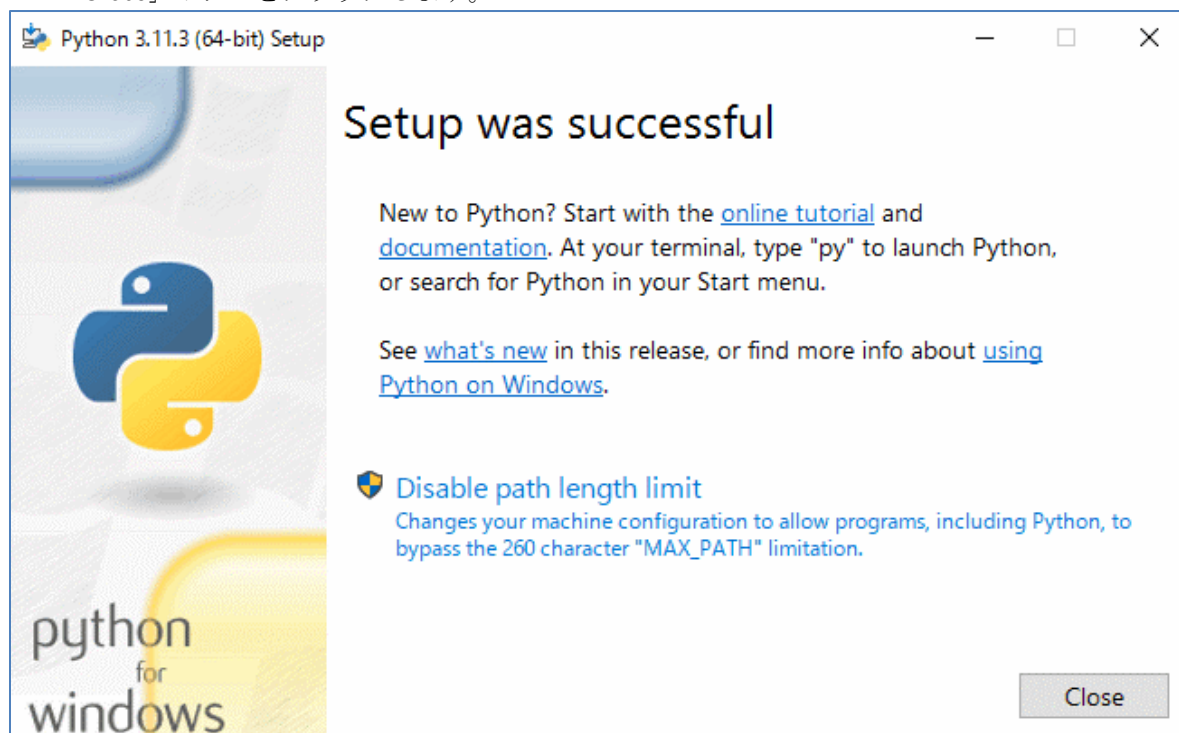
#### 2. Python のインストール

1. インストーラーを入手し、インストールするサーバーの任意のフォルダに格納します。
  - ・ダウンロード先 : <https://www.python.org/downloads/windows/>
  - ・ファイル名 : python-3.11.3-amd64.exe
2. インストールするサーバーに管理者権限のアカウントでログインします。

3. インストーラーを起動します。  
「Install Python 3.11.3(64-bit)」画面で「Add Python x to PATH」にチェックを入れ、「Install Now」をクリックします。



4. インストールが完了すると「Setup was successful」画面が表示されます。  
「Close」ボタンをクリックします。



以上で Python のインストールは終了です。



仕様補足

手順 3 にて「Add Python x to PATH」にチェックを入れなかった場合、手動で Python の PATH を設定する必要があります。

### 3. 環境変数の設定

Python で作成された WEB アプリケーションを Apache 上で動作させるためには、以下のシステム環境変数を設定する必要があります。

環境変数名	値
Path	[Python をインストールしたフォルダ] [Python をインストールしたフォルダ]¥Scripts
PYTHONHOME	[Python をインストールしたフォルダ]

環境変数名	値の例
Path	C:¥Users¥Administrator¥AppData¥Local¥Programs¥Python¥Python311 C:¥Users¥Administrator¥AppData¥Local¥Programs¥Python¥Python311¥Scripts
PYTHONHOME	C:¥Users¥Administrator¥AppData¥Local¥Programs¥Python¥Python311



仕様補足

「2.Python のインストール」手順 3 にて「Add Python x to PATH」にチェックを入れた場合、環境変数[Path]の設定は不要です。  
本項目の手順 6, 7 を省略してください。

下記手順に従って環境変数の設定を行います。

1. コントロールパネルを起動します。  
[スタート]を右クリック→[コントロールパネル]を選択します。
2. [コントロールパネル]の中から[システムとセキュリティ]をクリックします。

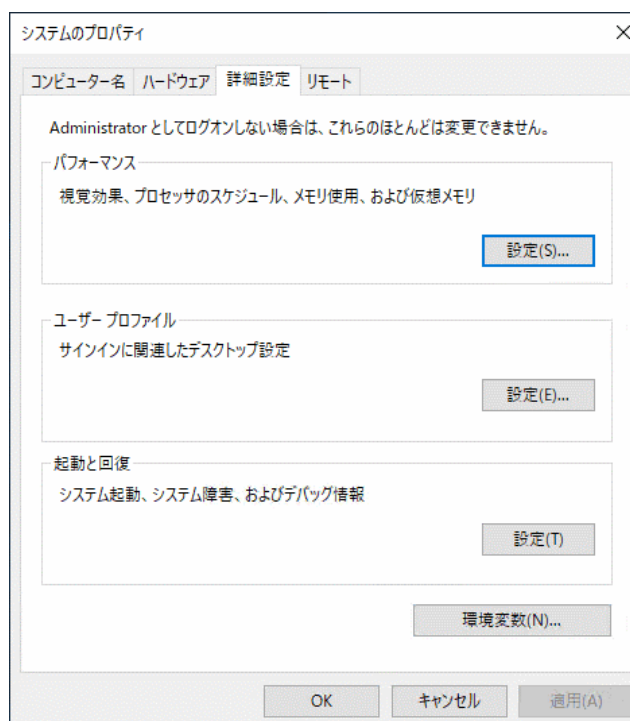


3. [システムとセキュリティ]の中から[システム]をクリックします。

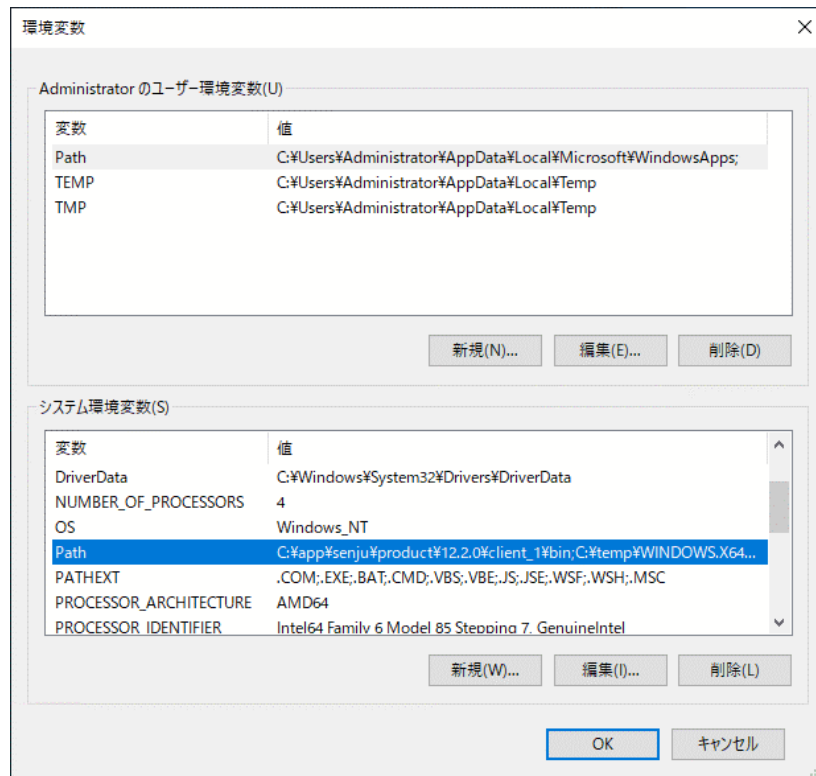
4. [システム]の左側メニューから[システムの詳細設定]をクリックします。



5. 「システムのプロパティ」画面が表示されるので、「環境変数」をクリックします。



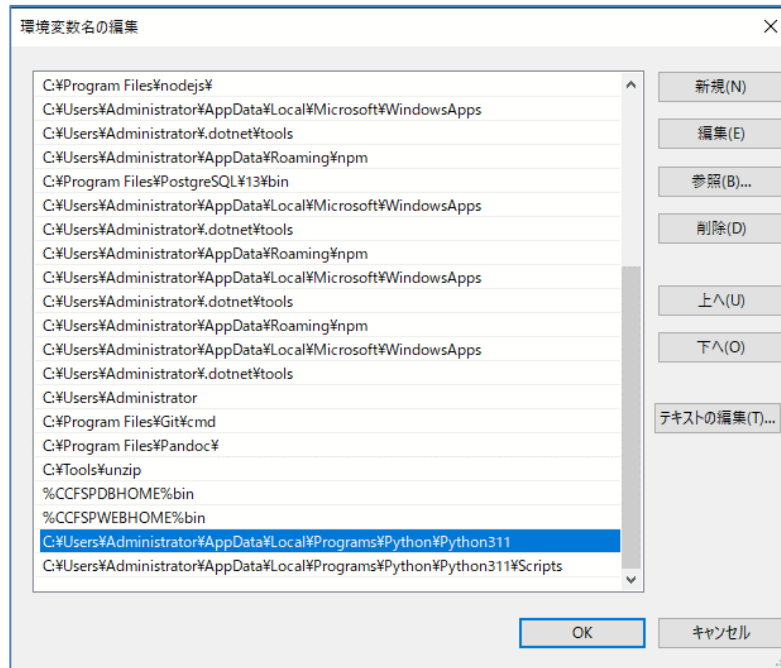
6. 「環境変数」画面の下部にある「システム環境変数」から、「変数」に「Path」と書かれた箇所を選択した状態で「編集」ボタンをクリックします。



7. 「環境変数名の編集」画面が表示されます。  
「新規」ボタンをクリックし、下記2つのパスを指定してください。

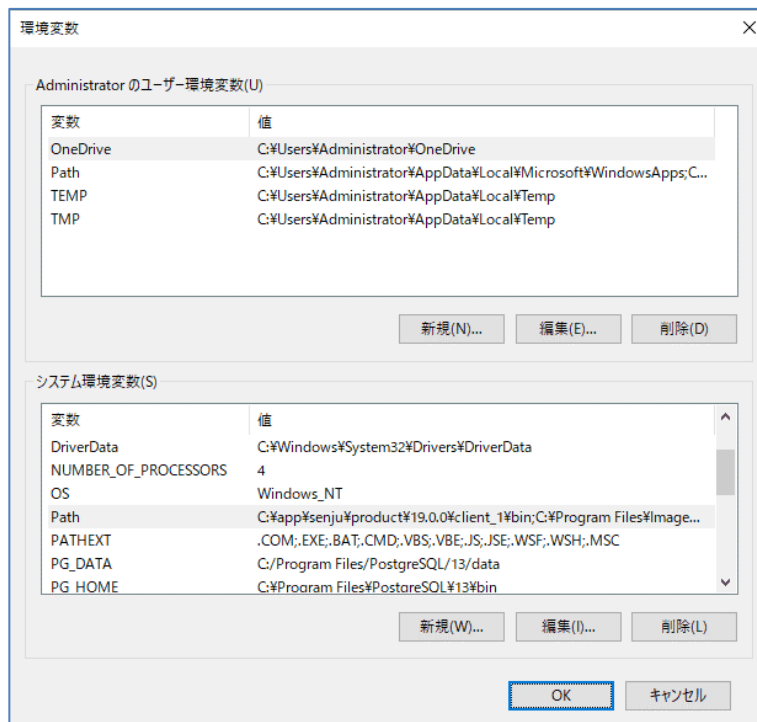
パス
[Python をインストールしたフォルダ]
[Python をインストールしたフォルダ]¥Scripts

パス例
C:\Users\Administrator\AppData\Local\Programs\Python\Python311
C:\Users\Administrator\AppData\Local\Programs\Python\Python311¥Scripts

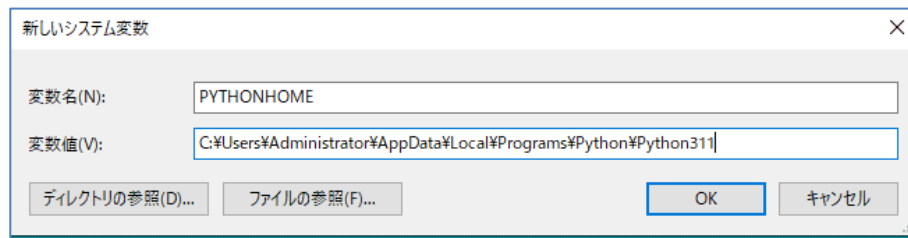


「OK」ボタンをクリックし、編集画面を閉じます。

8. 「環境変数」画面の下部にある「システム環境変数」から、「新規」ボタンをクリックします。



9. 「新しいシステム変数」画面にて、「変数名」に[PYTHONHOME]、「変数値」に[Pythonをインストールしたフォルダ]を入力します。



「OK」ボタンをクリックし、編集画面を閉じます。

以上で環境変数の設定は終了です。

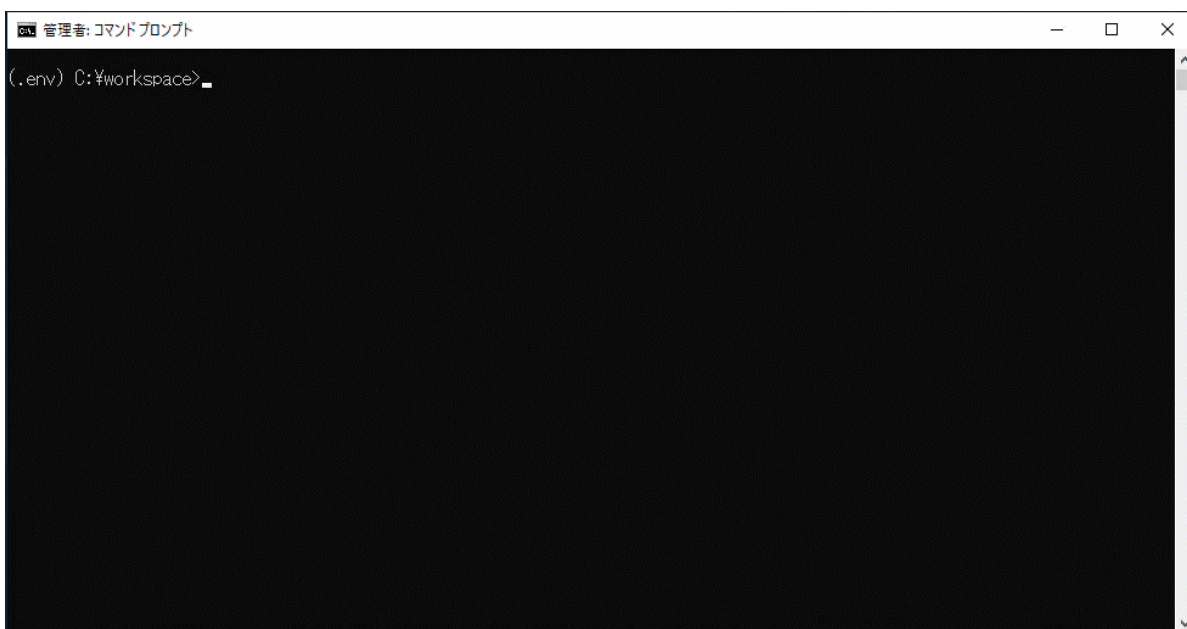


#### 4. 仮想環境の作成

1. ドライブ C 直下に仮想環境用のフォルダを作成します。(フォルダ名は任意です。)
2. コマンドプロンプトを新規に起動し、下記のコマンドを実行します。(“△”は半角スペースを示します。)

コマンド
<code>cd△C:¥[仮想環境用のフォルダ名]</code>
<code>py△-3.11△-m△venv△.env</code>
<code>.env¥Scripts¥activate</code>

3. コマンドプロンプトの画面が下図のように表示されることをご確認ください。



以上で仮想環境の作成は終了です。

## 5. Python パッケージのインストール

Senju Service Manager と連携する、API サーバーの WEB アプリケーションは以下の Python パッケージの機能を利用しています。

No	Python パッケージ名	バージョン	ファイル名
1	aniso8601	9.0.1	aniso8601-9.0.1-py2.py3-none-any.whl
2	attrs	19.3.0	attrs-19.3.0-py2.py3-none-any.whl
3	certifi	2022.12.7	certifi-2022.12.7-py3-none-any.whl
4	cffi	1.15.1	cffi-1.15.1-cp311-cp311-win_amd64.whl
5	chardet	3.0.4	chardet-3.0.4-py2.py3-none-any.whl
6	Click	7	Click-7.0-py2.py3-none-any.whl
7	cryptography	40.0.1	cryptography-40.0.1-cp36-abi3-win_amd64.whl
8	Flask	1.1.1	Flask-1.1.1-py2.py3-none-any.whl
9	Flask-JWT-Extended	3.24.1	Flask-JWT-Extended-3.24.1.tar.gz
10	flask-marshmallow	0.12.0	flask_marshmallow-0.12.0-py2.py3-none-any.whl
11	Flask-RESTful	0.3.8	Flask_RESTful-0.3.8-py2.py3-none-any.whl
12	Flask-SQLAlchemy	2.4.2	Flask_SQLAlchemy-2.4.2-py2.py3-none-any.whl
13	idna	2.8	idna-2.8-py2.py3-none-any.whl
14	itsdangerous	1.1.0	itsdangerous-1.1.0-py2.py3-none-any.whl
15	Jinja2	2.11.1	Jinja2-2.11.1-py2.py3-none-any.whl
16	jsonschema	3.2.0	jsonschema-3.2.0-py2.py3-none-any.whl
17	MarkupSafe	1.1.1	MarkupSafe-1.1.1.tar.gz
18	marshmallow	3.6.0	marshmallow-3.6.0-py2.py3-none-any.whl
19	marshmallow-sqlalchemy	0.23.0	marshmallow_sqlalchemy-0.23.0-py2.py3-none-any.whl
20	mod-wsgi	4.9.2	mod_wsgi-4.9.2-cp311-cp311-win_amd64.whl
21	oracledb	1.3.0	oracledb-1.3.0-cp311-cp311-win_amd64.whl
22	pip	23.0.1	pip-23.0.1-py3-none-any.whl
23	psycopg2	2.9.6	psycopg2-2.9.6-cp311-cp311-win_amd64.whl
24	pycparser	2.21	pycparser-2.21-py2.py3-none-any.whl
25	PyJWT	1.7.1	PyJWT-1.7.1-py2.py3-none-any.whl
26	pyrsistent	0.15.7	pyrsistent-0.15.7.tar.gz
27	pytz	2019.3	pytz-2019.3-py2.py3-none-any.whl
28	pywin32	306	pywin32-306-cp311-cp311-win_amd64.whl
29	requests	2.22.0	requests-2.22.0-py2.py3-none-any.whl
30	six	1.14.0	six-1.14.0-py2.py3-none-any.whl
31	SQLAlchemy	1.3.17	SQLAlchemy-1.3.17-cp37-cp37m-win_amd64.whl
32	tzlocal	2.1	tzlocal-2.1-py2.py3-none-any.whl
33	urllib3	1.25.11	urllib3-1.25.11-py2.py3-none-any.whl
34	Werkzeug	1.0.0	Werkzeug-1.0.0-py2.py3-none-any.whl
35	wheel	0.40.0	wheel-0.40.0-py3-none-any.whl



仕様補足

例として手順を説明するバージョンは開発時点のバージョンになります。インストール時には、セキュリティ上の観点からその時点での最適なバージョンを導入してください。

- Python パッケージを入手し、インストールするサーバーの C:\TEMP\modules に格納します。

No	Python パッケージ名	ダウンロード先	No
1	aniso8601	<a href="https://pypi.org/project/aniso8601/9.0.1/#files">https://pypi.org/project/aniso8601/9.0.1/#files</a>	1
2	attrs	<a href="https://pypi.org/project/attrs/19.3.0/#files">https://pypi.org/project/attrs/19.3.0/#files</a>	2
3	certifi	<a href="https://pypi.org/project/certifi/2022.12.7/#files">https://pypi.org/project/certifi/2022.12.7/#files</a>	3
4	cfffi	<a href="https://pypi.org/project/cffi/1.15.1/#files">https://pypi.org/project/cffi/1.15.1/#files</a>	4
5	chardet	<a href="https://pypi.org/project/chardet/3.0.4/#files">https://pypi.org/project/chardet/3.0.4/#files</a>	5
6	Click	<a href="https://pypi.org/project/click/7.0/#files">https://pypi.org/project/click/7.0/#files</a>	6
7	cryptography	<a href="https://pypi.org/project/cryptography/40.0.1/#files">https://pypi.org/project/cryptography/40.0.1/#files</a>	7
8	Flask	<a href="https://pypi.org/project/Flask/1.1.1/#files">https://pypi.org/project/Flask/1.1.1/#files</a>	8
9	Flask-JWT-Extended	<a href="https://pypi.org/project/Flask-JWT-Extended/3.24.1/#files">https://pypi.org/project/Flask-JWT-Extended/3.24.1/#files</a>	9
10	flask-marshmallow	<a href="https://pypi.org/project/flask-marshmallow/0.12.0/#files">https://pypi.org/project/flask-marshmallow/0.12.0/#files</a>	10
11	Flask-RESTful	<a href="https://pypi.org/project/Flask-RESTful/0.3.8/#files">https://pypi.org/project/Flask-RESTful/0.3.8/#files</a>	11
12	Flask-SQLAlchemy	<a href="https://pypi.org/project/Flask-SQLAlchemy/2.4.2/#files">https://pypi.org/project/Flask-SQLAlchemy/2.4.2/#files</a>	12
13	idna	<a href="https://pypi.org/project/idna/2.8/#files">https://pypi.org/project/idna/2.8/#files</a>	13
14	itsdangerous	<a href="https://pypi.org/project/itsdangerous/1.1.0/#files">https://pypi.org/project/itsdangerous/1.1.0/#files</a>	14
15	Jinja2	<a href="https://pypi.org/project/Jinja2/2.11.1/#files">https://pypi.org/project/Jinja2/2.11.1/#files</a>	15
16	jsonschema	<a href="https://pypi.org/project/jsonschema/3.2.0/#files">https://pypi.org/project/jsonschema/3.2.0/#files</a>	16
17	MarkupSafe	<a href="https://pypi.org/project/MarkupSafe/1.1.1/#files">https://pypi.org/project/MarkupSafe/1.1.1/#files</a>	17
18	marshmallow	<a href="https://pypi.org/project/marshmallow/3.6.0/#files">https://pypi.org/project/marshmallow/3.6.0/#files</a>	18
19	marshmallow-sqlalchemy	<a href="https://pypi.org/project/marshmallow-sqlalchemy/0.23.0/#files">https://pypi.org/project/marshmallow-sqlalchemy/0.23.0/#files</a>	19
20	mod-wsgi	<a href="https://www.lfd.uci.edu/~gohlke/pythonlibs/#mod_wsgi">https://www.lfd.uci.edu/~gohlke/pythonlibs/#mod_wsgi</a>	20
21	oracledb	<a href="https://pypi.org/project/oracledb/1.3.0/#files">https://pypi.org/project/oracledb/1.3.0/#files</a>	21
22	pip	<a href="https://pypi.org/project/pip/23.0.1/#files">https://pypi.org/project/pip/23.0.1/#files</a>	22
23	psycogp2	<a href="https://pypi.org/project/psycogp2/2.9.6/#files">https://pypi.org/project/psycogp2/2.9.6/#files</a>	23
24	pycparser	<a href="https://pypi.org/project/pycparser/2.21/#files">https://pypi.org/project/pycparser/2.21/#files</a>	24
25	PyJWT	<a href="https://pypi.org/project/PyJWT/1.7.1/#files">https://pypi.org/project/PyJWT/1.7.1/#files</a>	25
26	pyrsistent	<a href="https://pypi.org/project/pyrsistent/0.15.7/#files">https://pypi.org/project/pyrsistent/0.15.7/#files</a>	26
27	pytz	<a href="https://pypi.org/project/pytz/2019.3/#files">https://pypi.org/project/pytz/2019.3/#files</a>	27
28	pywin32	<a href="https://pypi.org/project/pywin32/306/#files">https://pypi.org/project/pywin32/306/#files</a>	28
29	requests	<a href="https://pypi.org/project/requests/2.22.0/#files">https://pypi.org/project/requests/2.22.0/#files</a>	29
30	six	<a href="https://pypi.org/project/six/1.14.0/#files">https://pypi.org/project/six/1.14.0/#files</a>	30
31	SQLAlchemy	<a href="https://pypi.org/project/SQLAlchemy/1.3.17/#files">https://pypi.org/project/SQLAlchemy/1.3.17/#files</a>	31
32	tzlocal	<a href="https://pypi.org/project/tzlocal/2.1/#files">https://pypi.org/project/tzlocal/2.1/#files</a>	32
33	urllib3	<a href="https://pypi.org/project/urllib3/1.25.11/#files">https://pypi.org/project/urllib3/1.25.11/#files</a>	33
34	Werkzeug	<a href="https://pypi.org/project/Werkzeug/1.0.0/#files">https://pypi.org/project/Werkzeug/1.0.0/#files</a>	34
35	wheel	<a href="https://pypi.org/project/wheel/0.40.0/#files">https://pypi.org/project/wheel/0.40.0/#files</a>	35

※…ダウンロードするファイルの選択肢が複数存在します。

以下を参考にファイル名に含まれる文字列からファイルを選択してください。

- ・ cp[Python のバージョン]
- 例) cp311…Python3.11 用
- ・ win\_amd64 : Windows(64-bit)用



補足説明

- Python パッケージのファイル名一覧をテキストファイルに記載し、C:\TEMP\modules に保存します。
  - ファイル名 : requirements.txt

requirements.txt の記載内容
Flask==1.1.1
Flask-JWT-Extended==3.24.1
flask-marshmallow==0.12.0
Flask-RESTful==0.3.8
Flask-SQLAlchemy==2.4.2
jjsonschema==3.2.0
marshmallow==3.6.0
marshmallow-sqlalchemy==0.23.0
mod-wsgi==4.9.2
oracledb==1.3.0
psycopg2==2.9.6
pywin32==306
requests==2.22.0
SQLAlchemy==1.3.17
tzlocal==2.1

- コマンドプロンプトを新規に起動し、下記のコマンドを実行します。(“△”は半角スペースを示します。)

コマンド
cd△C:¥[仮想環境用のフォルダ名]
.env¥Scripts¥activate

4. 仮想環境上で以下のコマンドを実行し、Python パッケージのインストールを行います。（“△”は半角スペースを示します。）

コマンド
<code>cd△C:¥TEMP¥modules</code>
<code>pip△install△--no-index△--find-links△C:¥TEMP¥modules△-r△requirements.txt</code>

```

管理: C:\Windows\System32\cmd.exe

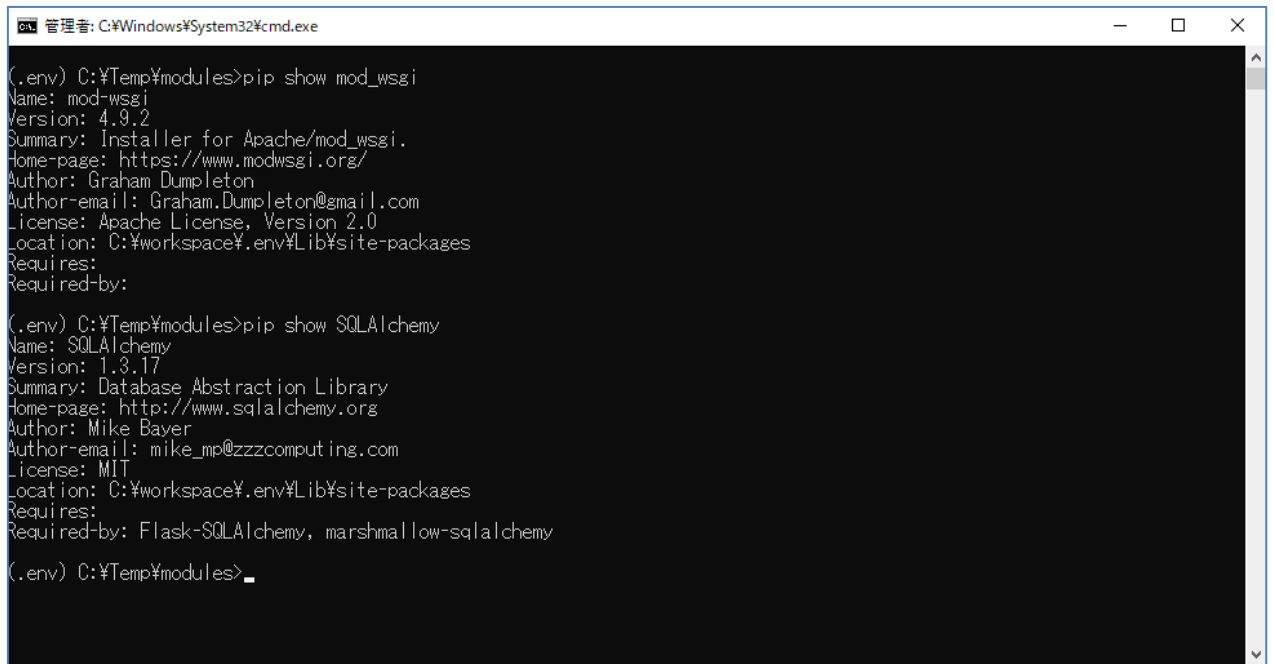
(.env) C:¥Temp¥modules>python -m pip install pip-23.0.1-py3-none-any.whl
Processing c:¥temp¥modules¥pip-23.0.1-py3-none-any.whl
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3.1
    Uninstalling pip-22.3.1:
      Successfully uninstalled pip-22.3.1
Successfully installed pip-23.0.1
WARNING: There was an error checking the latest version of pip.

(.env) C:¥Temp¥modules>pip install wheel-0.40.0-py3-none-any.whl
Processing c:¥temp¥modules¥wheel-0.40.0-py3-none-any.whl
Installing collected packages: wheel
Successfully installed wheel-0.40.0

(.env) C:¥Temp¥modules>pip install --no-index --find-links . -r requirements.txt
Looking in links: .
Processing c:¥temp¥modules¥flask-1.1.1-py2.py3-none-any.whl
Processing c:¥temp¥modules¥flask-jwt-extended-3.24.1.tar.gz
  Preparing metadata (setup.py) ... done
Processing c:¥temp¥modules¥flask-marshmallow-0.12.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥flask-restful-0.3.8-py2.py3-none-any.whl
Processing c:¥temp¥modules¥flask-sqlalchemy-2.4.2-py2.py3-none-any.whl
Processing c:¥temp¥modules¥jsonschema-3.2.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥marshmallow-3.6.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥marshmallow-sqlalchemy-0.23.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥mod_wsgi-4.9.2-cp311-cp311-win_amd64.whl
Processing c:¥temp¥modules¥oracledb-1.3.0-cp311-cp311-win_amd64.whl
Processing c:¥temp¥modules¥psycopy2-2.9.6-cp311-cp311-win_amd64.whl
Processing c:¥temp¥modules¥pywin32-306-cp311-cp311-win_amd64.whl
Processing c:¥temp¥modules¥requests-2.22.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥sqlalchemy-1.3.17.tar.gz
  Preparing metadata (setup.py) ... done
Processing c:¥temp¥modules¥tzlocal-2.1-py2.py3-none-any.whl
Processing c:¥temp¥modules¥werkzeug-1.0.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥jinja2-2.11.1-py2.py3-none-any.whl
Processing c:¥temp¥modules¥itsdangerous-1.1.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥click-7.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥pyjwt-1.7.1-py2.py3-none-any.whl
Processing c:¥temp¥modules¥six-1.14.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥anis8601-9.0.1-py2.py3-none-any.whl
Processing c:¥temp¥modules¥pytz-2019.3-py2.py3-none-any.whl
Processing c:¥temp¥modules¥attrs-19.3.0-py2.py3-none-any.whl
Processing c:¥temp¥modules¥pyrsistent-0.15.7.tar.gz
  Preparing metadata (setup.py) ... done
Requirement already satisfied: setuptools in c:¥workspace¥senjsum-api¥.env¥lib¥site-packages (from jsonschema==3.2.0->-r requirements.txt (line 6)) (65.5.0)
Processing c:¥temp¥modules¥cryptography-40.0.1-cp36-abi3-win_amd64.whl
Processing c:¥temp¥modules¥chardet-3.0.4-py2.py3-none-any.whl
Processing c:¥temp¥modules¥idna-2.8-py2.py3-none-any.whl
Processing c:¥temp¥modules¥urllib3-1.25.11-py2.py3-none-any.whl
Processing c:¥temp¥modules¥certifi-2022.12.7-py3-none-any.whl
Processing c:¥temp¥modules¥cffi-1.15.1-cp311-cp311-win_amd64.whl
Processing c:¥temp¥modules¥markupsafe-1.1.1.tar.gz
  Preparing metadata (setup.py) ... done
Processing c:¥temp¥modules¥pyparser-2.21-py2.py3-none-any.whl
Building wheels for collected packages: Flask-JWT-Extended, SQLAlchemy, pyrsistent, MarkupSafe
  Building wheel for Flask-JWT-Extended (setup.py) ... done
  Created wheel for Flask-JWT-Extended: filename=Flask_JWT_Extended-3.24.1-py2.py3-none-any.whl size=21549 sha256=cb6fecf1c91a9e3260f5f2b614dd0fc4350697f5c5c9552d42187db1657bd6ef
  Stored in directory: c:¥users¥administrator¥appdata¥local¥pip¥cache¥wheels¥83¥05¥73¥1f1f145e211a54937c0d5c58cddb28cffa77234ecbf9b7f33
  
```

5. 仮想環境上で以下のコマンドを実行し、Python パッケージがインストールされたことを確認します（“△” は半角スペースを示します。）

コマンド
<code>pip△show△&lt;Python パッケージ名&gt;</code>



```
管理: C:\Windows\System32\cmd.exe
(.env) C:\Temp\modules>pip show mod_wsgi
Name: mod-wsgi
Version: 4.9.2
Summary: Installer for Apache/mod_wsgi.
Home-page: https://www.modwsgi.org/
Author: Graham Dumbleton
Author-email: Graham.Dumbleton@gmail.com
License: Apache License, Version 2.0
Location: C:\workspace\.env\Lib\site-packages
Requires:
Required-by:

(.env) C:\Temp\modules>pip show SQLAlchemy
Name: SQLAlchemy
Version: 1.3.17
Summary: Database Abstraction Library
Home-page: http://www.sqlalchemy.org
Author: Mike Bayer
Author-email: mike_mp@zzzcomputing.com
License: MIT
Location: C:\workspace\.env\Lib\site-packages
Requires:
Required-by: Flask-SQLAlchemy, marshmallow-sqlalchemy

(.env) C:\Temp\modules>
```

以上で Python パッケージのインストールは終了です。



## 1.4.2 Apache の導入

Senju Service Manager と連携する API サーバーは WEB サーバーに Apache を使用します。  
Apache をインストールするための手順について説明します。

### 1. 対象バージョン

Senju Service Manager と連携する API サーバーでは Windows プラットフォームの 64 ビット Apache を使用します。  
ここでは、64 ビット Apache 2.4.58 を例として導入手順を説明します。



仕様補足

サポート対象となる Apache のバージョンについては リリースノート を参照してください。



仕様補足

例として手順を説明するバージョンは開発時点のバージョンになります。インストール時には、セキュリティ上の観点を考慮してその時点での最適なバージョンを導入してください。

### 2. Microsoft VC++ 再頒布可能パッケージのインストール

1. インストーラーを入手し、インストールするサーバーの C:\TEMP に格納します。
  - ・ダウンロード先：  
<https://support.microsoft.com/ja-jp/help/2977003/the-latest-supported-visual-c-downloads>
  - ・ファイル名： VC\_redist.x64.exe



補足説明

Windows 版の Apache は Visual Studio でビルドされているため、ランタイムパッケージである Visual Studio 2015-2019 用の Microsoft VC++再頒布可能パッケージが必要となります。

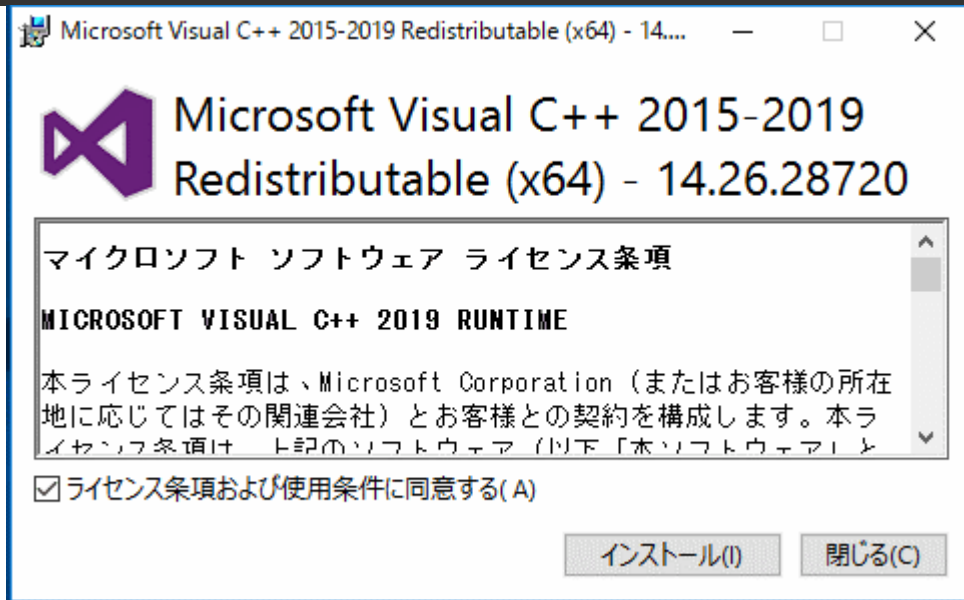


仕様補足

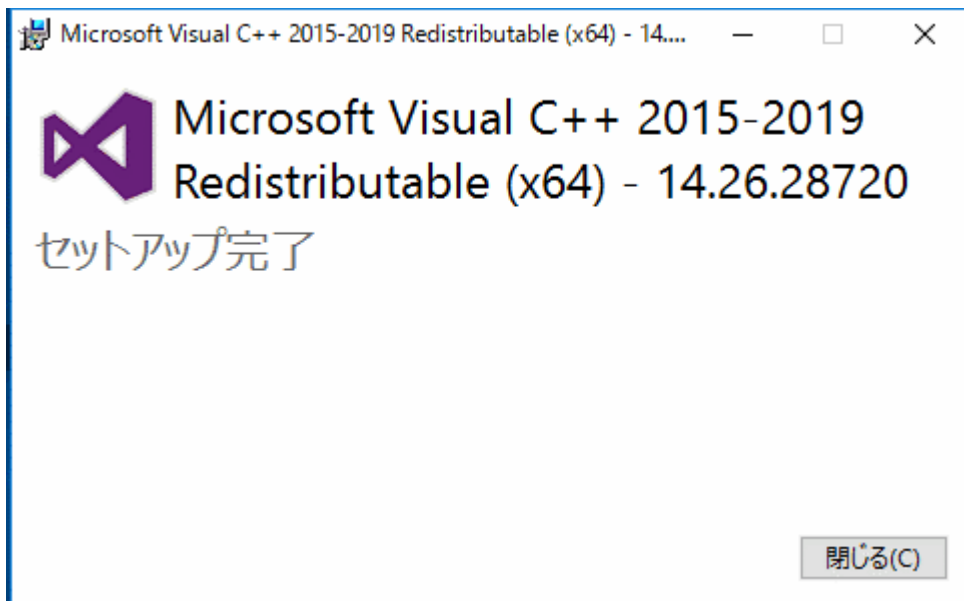
例として手順を説明するバージョンは開発時点のバージョンになります。インストール時には、セキュリティ上の観点を考慮してその時点での最適なバージョンを導入してください。

2. インストーラーを起動します。

「マイクロソフト ソフトウェア ライセンス条項」画面で「ライセンス条項および使用条件に同意する(A)」にチェックを入れ、「インストール(I)」 ボタンをクリックします。



3. 「セットアップ完了」画面で「閉じる(C)」ボタンをクリックします。



以上で Microsoft VC++再頒布可能パッケージのインストールは終了です。

### 3. Apache のインストール

1. インストール用の zip ファイルを入手し、インストールするサーバーの C:\TEMP に格納します。
  - ・ダウンロード先 : <https://www.apachelounge.com/download/>
  - ・ファイル名 : httpd-2.4.58-win64-VS17.zip
2. インストールするサーバーに管理者権限のアカウントでログインします。



- zip ファイルを展開し、「Apache24」フォルダを C ドライブ直下に格納します。



仕様補足

Apache のインストール先は設定ファイルに記載されており、初期状態では C:\Apache24 となっています。

インストール先を変更する場合、設定ファイル中でインストール先の記述を修正してください。

設定ファイル : Apache24\conf\httpd.conf

- Apache の設定ファイル(httpd.conf)をテキストエディタで開き、以下の内容を編集します。

- 1 ポート番号を設定します。

```
#Listen 80
Listen 49152
```

- 2 サーバー名を設定します。

```
#ServerName www.example.com:80
ServerName ssmapihost:49152
```

- コマンドプロンプトを新規に起動し、下記のコマンドを実行します。(“△” は半角スペースを示します。)

コマンド
cd△C:¥Apache24¥bin
httpd.exe

- Web ブラウザから Apache のドキュメントルートページを表示し、「It works!」が画面に表示されていることを確認します。

URL
http://ssmapihost:49152

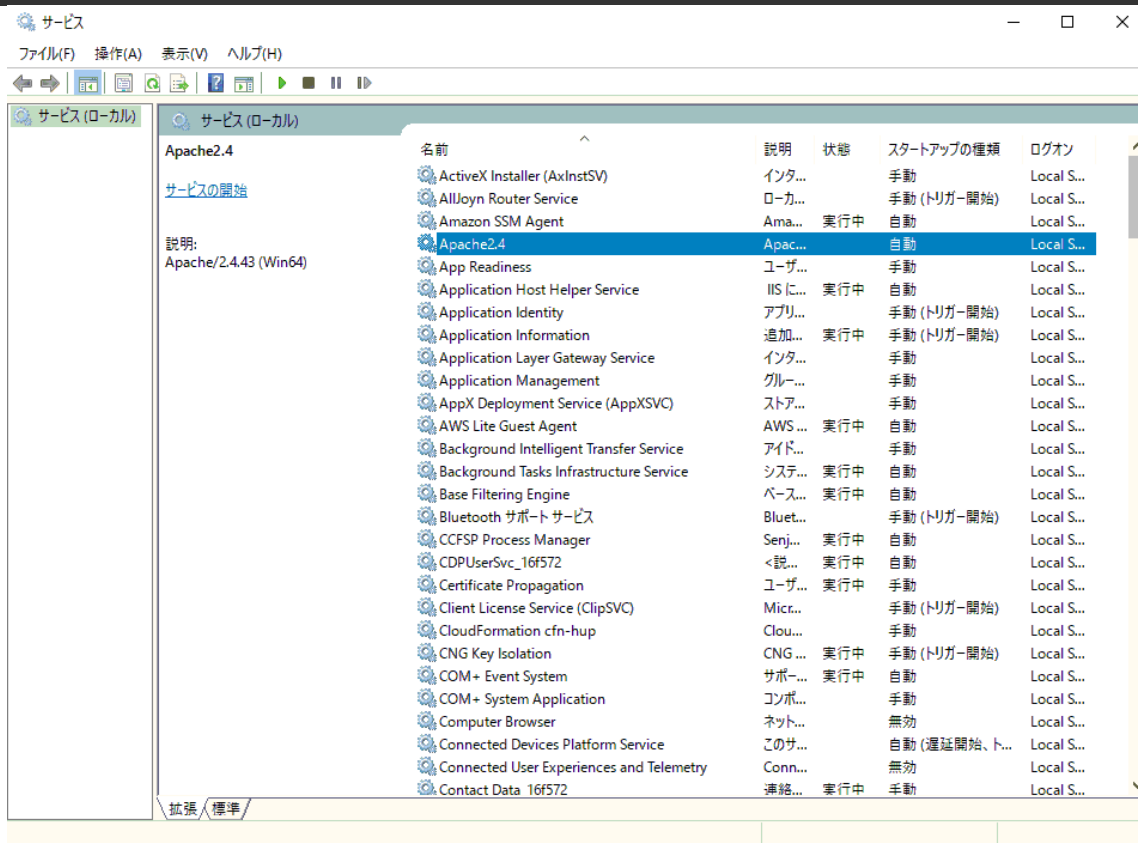
- コマンドプロンプトで `ctrl+c` を入力し、起動中の `httpd.exe` を終了させます。

- コマンドプロンプトで下記のコマンドを実行し、Windows のサービスに「Apache24」を登録します。(“△” は半角スペースを示します。)

コマンド
httpd.exe△-k△install

- 「スタート」メニュー>「管理ツール」>「サービス」を選択し、「サービス」画面を表示します。

「Apache2.4」がサービスに登録されていることを確認します。



以上で Apache のインストールは終了です。

## 4. Apache の設定(SSL 通信の設定)



この手順は SSL を利用した接続を行わない場合は不要です。



Apache の SSL 設定ファイルは以下の場所に格納されています。  
C:\Apache24\conf\extra\httpd-ssl.conf

API サーバーへ SSL を利用して接続を行うための設定を行います。

SSL を利用して接続を行う場合に必要となるサーバー証明書・サーバーの秘密鍵の用意は、サーバー証明書発行機関の手順書に沿って事前に実施してください。

- 1 Apache の設定ファイル(httpd.conf)をテキストエディタで開き、以下の内容を編集します。

ポート番号を無効にします。

```
#Listen 49152
```

SSL モジュールを有効にします。

```
#LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
#LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule ssl_module modules/mod_ssl.so
```

SSL 設定ファイル”httpd-ssl.conf”を有効にします。

```
#Include conf/extra/httpd-ssl.conf
Include conf/extra/httpd-ssl.conf
```

- 2 Apache の SSL 設定ファイル(httpd-ssl.conf)をテキストエディタで開き、以下の内容を編集します。

ポート番号を設定します。

```
#Listen 443
Listen 49152
```

サーバー名を設定します。

```
#ServerName www.example.com:443
ServerName ssmapihost:49152
```

CGI の実行を許可します。

```
#<Directory "${SRVROOT}/cgi-bin">
#   SSLOptions +StdEnvVars
#</Directory>
```

- 3 サーバー証明書・サーバーの秘密鍵を以下の場所に配置します。

```
C:\Apache24\conf\server.key
C:\Apache24\conf\ssl.crt
```

- 4 Apache を再起動します。  
「スタート」メニュー>「管理ツール」>「サービス」を選択し、「サービス」画面を表示します。

「Apache2.4」を右クリックし、「再起動(E)」をクリックします。

以上で Apache の設定 (SSL 通信の設定) は終了です。

## 1.5 API連携機能の導入

### 1.5.1 WEB アプリケーションのインストール

API サーバー上で動作する WEB アプリケーションをインストールするための手順について説明します。

#### 1. WEB アプリケーションのインストール

1. インストール用の whl ファイルを入手して、インストールするサーバーの C:\TEMP に格納します。

・ファイル名: ssmwebapi-2023.0.1.1-py3-none-any.whl



仕様補足

Web アプリケーションは Python で作成されています。  
Python のパッケージ管理システムを用いてインストールします。

2. インストールするサーバーに管理者権限のアカウントでログインします。
3. コマンドプロンプトを新規に起動し、下記のコマンドを実行します。（“△”は半角スペースを示します。）

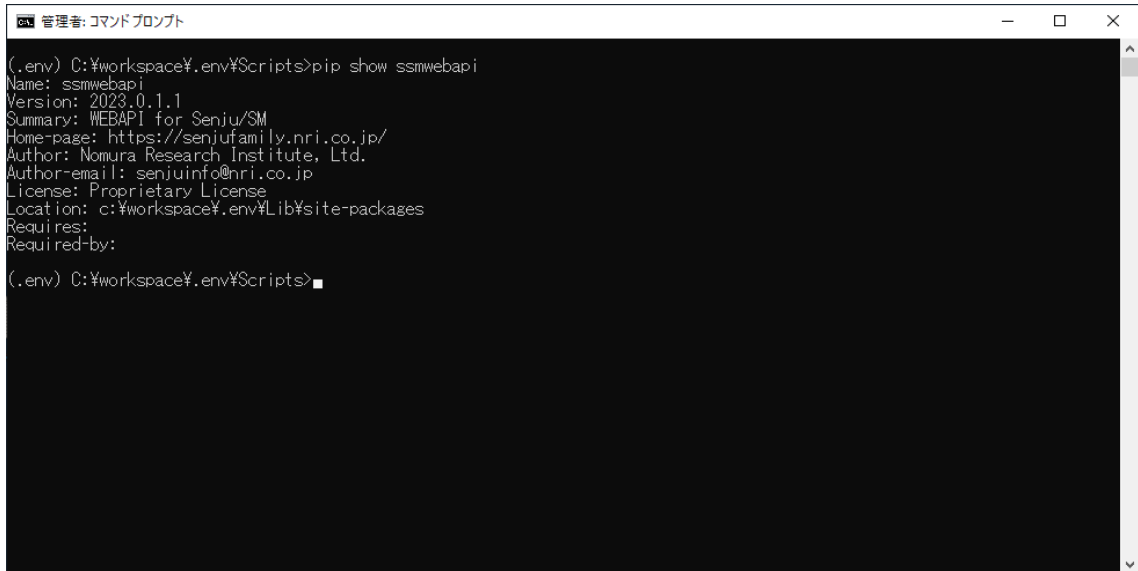
コマンド
cd△C:¥[仮想環境用のフォルダ名]
.env¥Scripts¥activate

4. 仮想環境で下記のコマンドを実行します。（“△”は半角スペースを示します。）

コマンド
cd△C:¥TEMP
pip△install△-U△ssmwebapi-2023.0.1.1-py3-none-any.whl

5. 仮想環境で下記のコマンドを実行し、WEB アプリケーションがインストールされていることを確認します。（“△”は半角スペースを示します。）

コマンド
pip△show△ssmwebapi



```
管理: コマンドプロンプト
(.env) C:\workspace¥.env¥Scripts>pip show ssmwebapi
Name: ssmwebapi
Version: 2023.0.1.1
Summary: WEBAPI for Senju/SM
Home-page: https://senjufamily.nri.co.jp/
Author: Nomura Research Institute, Ltd.
Author-email: senjuinfo@nri.co.jp
License: Proprietary License
Location: c:\workspace¥.env¥Lib¥site-packages
Requires:
Required-by:

(.env) C:\workspace¥.env¥Scripts>
```

以上で WEB アプリケーションのインストールは終了です。

## 1.5.2 WEB アプリケーションの設定

API サーバー上で動作する WEB アプリケーションの設定手順について説明します。

### 1. 資材の格納

インストールする資材を入手して、インストールするサーバーの C: ¥Program Files¥CCFSP 配下に格納します。

ファイルおよびフォルダ	説明
ccfspwebapi	WEB アプリケーションのルートフォルダ
ccfspwebapi¥conf	WEB アプリケーションの設定ファイル格納フォルダ
ccfspwebapi¥conf¥ssmwebapi.conf.template	WEB アプリケーションの設定ファイルテンプレート
ccfspwebapi¥logs	WEB アプリケーションのログ出力フォルダ
ccfspwebapi¥wsgy.py	WEB アプリケーションの起動スクリプト

### 2. WEB アプリケーションの設定変更

WEB アプリケーションの設定は、以下のファイルに記載されています。

- ・ ファイルパス:C: ¥Program Files¥CCFSP¥ccfspwebapi¥conf¥ssmwebapi.conf.template

設定ファイル変更前に、必ずテンプレートファイルの拡張子の `template` を外して、`conf` 拡張子に修正してください。

```
# conf 拡張子へ修正前
ssmwebapi.conf.template
# conf 拡張子へ修正後
ssmwebapi.conf
```

拡張子の修正が終わった設定ファイルは必要に応じて設定を変更してください。



仕様補足

ファイルは Python 形式で記載されています。

- ・ #から行末までの記載はコメントとして扱われます。
- ・ タブやスペースを行頭に追加しないでください。  
文法エラーとなり API サーバーが起動しなくなる可能性があります。



仕様補足

WEB アプリケーションの設定変更を反映するためには、ファイル編集後に WEB サーバーを再起動する必要があります。



仕様補足

既に使用している `conf` ファイルが存在する場合、上記テンプレートファイルを参考にして各バージョンごとに追加された項目を適宜追加してください。

1. SSMWEB サーバーのプロトコルを指定します。

```
# http の場合
SSMWEB_SERVER_PROTOCOL = "http"
# https の場合
SSMWEB_SERVER_PROTOCOL = "https"
```

- SSMWEB サーバーのホスト名を指定します。

```
SSMWEB_SERVER_HOST = "ssmwebhost"
```

- データベースの接続設定を記載します。

```
# Oracle Database の場合
SQLALCHEMY_DATABASE_URI = "oracle://ssmuser:ssmpwd@oracledbhost:1522/ssmdb"
# PostgreSQL の場合
SQLALCHEMY_DATABASE_URI =
"postgresql+psycopg2://ssmuser:ssmpwd@localhost:5432/ssmdb"
```

データベースの接続設定は以下の形式で指定します。



仕様補足

- Oracle Database の場合  
"oracle://<DB ユーザー名>:<DB ユーザーパスワード>@<Oracle Database ホスト名>:<リスナーポート番号>/<データベース名>"
- PostgreSQL の場合  
"postgresql+psycopg2://<DB ユーザー名>:<DB ユーザーパスワード>@<PostgreSQL ホスト名>:<DB ポート番号>/<データベース名>"

- トークンを暗号化する際のシード値を指定します。

```
SECRET_KEY = b'p¥x8eT¥x8e¥xb8bk¥xe2 ¥xb8¥xee¥xdb¥x0c¥x0f¥t¥xef'
```

以下の操作でシード値を生成してください。  
仮想環境上で以下のコマンドを実行します。(“△”は半角スペースを示します。)

```
コマンド
python△-c△"import△os;print(os.urandom(24))"
```



操作説明

```
管理者: コマンドプロンプト
(.env) C:¥workspace>python -c "import os;print(os.urandom(24))"
b'S_¥xcc¥xea¥xbd# ¥.¥xb4¥x86cP¥x03¥xa5M¥x1f¥x98¥xa7R¥xe2¥x15DT'
```

- アクセストークン・リフレッシュトークンの有効期限を指定します。

```
# アクセストークンの有効期限
JWT_ACCESS_TOKEN_EXPIRES = datetime.timedelta(hours=1)
# リフレッシュトークンの有効期限
JWT_REFRESH_TOKEN_EXPIRES = datetime.timedelta(days=30)
```

6. ヘルスチェック API の公開/非公開を制御します。

```
# ヘルスチェック API を公開する場合  
PUBLISH_HEALTHCHECK_API = True  
# ヘルスチェック API を非公開にする場合  
PUBLISH_HEALTHCHECK_API = False
```



仕様補足

設定ファイルに本項目が記載されていない場合、True(公開)が設定された扱いとなります。

7. SSL 認証設定の有効/無効を制御します。

```
# SSL 検証を行う場合(有効)  
SSL_CERT_VERIFICATION = True  
# SSL 検証を行う場合(無効)  
SSL_CERT_VERIFICATION = False
```



仕様補足

指定された値が上記以外の場合または設定ファイルに本項目が記載されていない場合、True(有効)が設定された扱いとなります。

8. アクセス URL の取得には以下を設定します。

```
# アクセス URL 設定項目に <SSM 本体の URL>を設定する。  
ACCESS_URL_SOURCE = "http://your-senjusm-url/ssm"
```



仕様補足

設定ファイルに本項目が記載されていない場合、アクセス URL ・ SAML 認証用アクセス URL の値は空文字が返却されます。

以上で WEB アプリケーションの設定は終了です。



## 1.5.3 WEB サーバーと WEB アプリケーションの接続設定

WEB サーバーと WEB アプリケーションを接続するための設定手順について説明します。

### 1. WEB サーバーの WSGI 対応

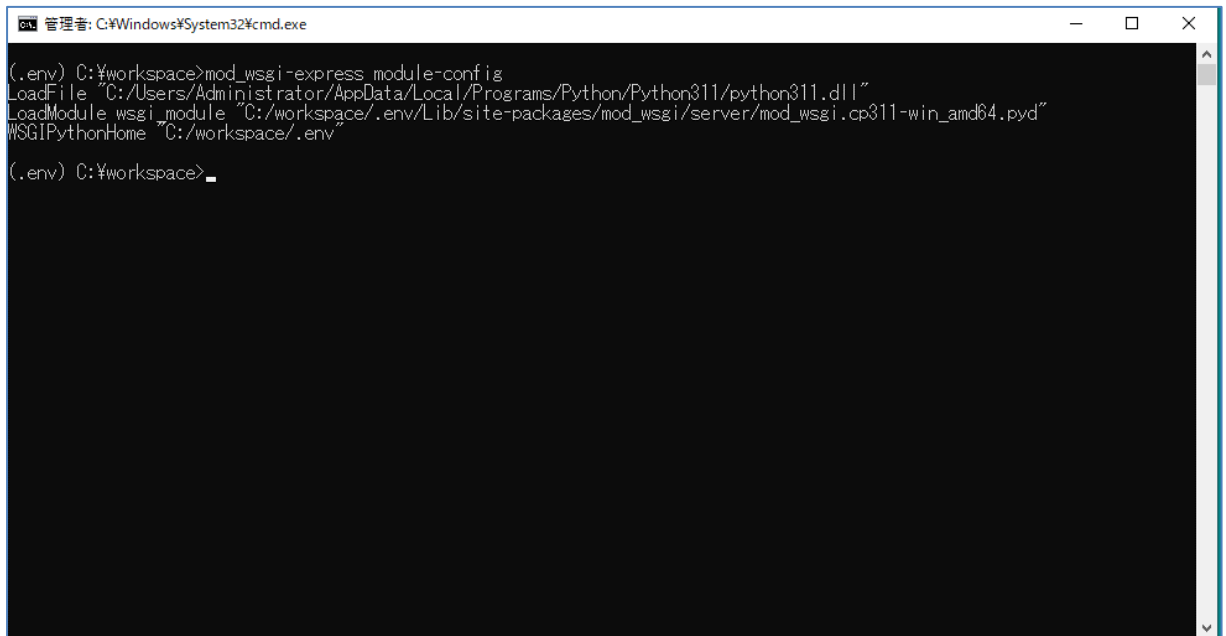
WEB サーバーと Python で作成された WEB アプリケーションを接続するためには、WEB サーバーが WSGI(Web Server Gateway Interface)に対応している必要があります。しかし、Apache は WSGI に対応していないため、以下の手順で WSGI に対応させます。

1. API サーバーに管理者権限のアカウントでログインします。
2. コマンドプロンプトを新規に起動し、下記のコマンドを実行します。(“△” は半角スペースを示します。)

コマンド
cd△C:¥[仮想環境用のフォルダ名] ¥.env¥Scripts¥activate

3. 仮想環境で下記のコマンドを実行します。(“△” は半角スペースを示します。)

コマンド
mod_wsgi-express△module-config



```
管理: C:\Windows\System32\cmd.exe
(.env) C:¥workspace>mod_wsgi-express module-config
LoadFile "C:/Users/Administrator/AppData/Local/Programs/Python/Python311/python311.dll"
LoadModule wsgi_module "C:/workspace/.env/Lib/site-packages/mod_wsgi/server/mod_wsgi.cp311-win_amd64.pyd"
WSGIPythonHome "C:/workspace/.env"
(.env) C:¥workspace>_
```

4. コマンドの出力結果をコピーし、Apache の設定ファイルに貼り付けます。



仕様補足

SSL 通信を行う場合は、Apache の SSL 設定ファイル (httpd-ssl.conf) に貼り付けてください。

## 2. WEB サーバー起動時に実行される WEB アプリケーションの起動スクリプト登録

WEB サーバー起動時に実行される Python のスクリプトを登録します。

1. Apache の設定ファイルに以下の内容を追記します。

```
<VirtualHost *:49152>
  DocumentRoot "C:/Program Files/CCFSP/ccfspwebapi/"
  WSGIScriptAlias / "C:/Program Files/CCFSP/ccfspwebapi/wsgi.py"
  <Directory "C:/Program Files/CCFSP/ccfspwebapi">
    Require all granted
  </Directory>
  SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1
</VirtualHost>
```



仕様補足

SSL 通信を行う場合は、Apache の SSL 設定ファイル (httpd-ssl.conf) に記述を追記してください。



仕様補足

Apache の設定ファイル中では、Windows でもパスセパレータとしてスラッシュを使用してください。

## 3. WEB サーバーの再起動

設定を反映させるため、WEB サーバーを再起動します。

1. 「スタート」メニュー>「管理ツール」>「サービス」を選択し、「サービス」画面を表示します。
2. 「Apache2.4」を右クリックし、「再起動(E)」をクリックします。

以上で WEB サーバーと WEB アプリケーションの接続設定は終了です。

## 1.5.4 ウィルススキャンの除外

API サーバー が稼働している環境において、アンチウイルスソフトのようなセキュリティ関連ソフトや、バックアップソフト等の予期しない動作により、パフォーマンスの影響や動作不調を起こす場合があります。

そのため、アプリケーションやミドルウェアのフォルダやファイルをリアルタイム検索から除外していただく必要があります。

各アンチウイルスソフトの除外設定手順に従い、設定を行ってください。  
対象は 資料集「1.1.1 Senju Service Manager システム」の 9)を参照してください。

## 1.5.5 WEBAPI の複数導入設定

Senju Service Manager が提供する WEBAPI はバージョンアップやパッチ適用により仕様が変わり、現在ご利用中の運用に影響がある場合がございます。急な仕様変更に対応するために、旧バージョンと新バージョンの WEBAPI を同時に稼働し、接続先をポート番号で分けることで両方の仕様を使い分けることが可能となります。

複数の WEBAPI を稼働する必要がある場合、以下の手順に従って追加の「Apache」と「WEBAPI」のインストールを行ってください。



仕様補足

2 台の Apache をインストールして複数の WEBAPI を稼働することで、サーバーへの負荷や性能に影響を及ぼす可能性があります。

### 1. 追加 WEBAPI のインストール



補足説明

別の WEBAPI をインストールするためには、既存と違う仮想環境からインストールする必要があります。仮想環境の作成方法は以下の手順を参照してください。  
「1.4.1.4 仮想環境の作成」

本マニュアルの「1.5.1 WEB アプリケーションのインストール」、「1.5.2 WEB アプリケーションの設定」の手順に従い、追加作成した仮想環境へ WEBAPI をインストールしてください。

### 2. 追加 Apache のインストール

1. インストール用の zip ファイルを入手し、インストールするサーバーの C:\TEMP に格納します。
  - ダウンロード先 : <https://www.apachelounge.com/download/>
  - ファイル名 : httpd-2.4.58-win64-VS17.zip
2. インストールするサーバーに管理者権限のアカウントでログインします。

- zip ファイルを展開し、既存インストールした Apache と区分するため、フォルダ「Apache24」のフォルダ名を「Apache24\_2」で変更して C ドライブ直下に格納してください。



仕様補足

追加作成した Apache のフォルダ名変更に従って、格納経路についての設定を変更する必要があります。以下の設定ファイルをテキストエディタで開いて設定ファイルを修正してください。

設定ファイル : Apache24\_2\conf\httpd.conf

```
Define SRVROOT "c:/Apache24_2"
ServerRoot "${SRVROOT}"
```

- Apache の設定ファイル(httpd.conf)をテキストエディタで開き、以下の内容を編集します。

- 1 既存使用中のポート番号と被らないポート番号を設定してください。

```
#Listen 80
Listen 49153
```

- 2 サーバー名を設定します。

```
#ServerName www.example.com:80
ServerName ssmapihost:49153
```

- コマンドプロンプトを新規に起動し、下記のコマンドを実行します。(“△” は半角スペースを示します。)

コマンド

```
cd△C:¥Apache24_2¥bin
httpd.exe
```

- Web ブラウザから Apache のドキュメントルートページを表示し、「It works!」が画面に表示されていることを確認します。

URL

```
http://ssmapihost:49153
```

- コマンドプロンプトで ctrl+c を入力し、起動中の httpd.exe を終了させます。
- コマンドプロンプトで下記のコマンドを実行し、Windows のサービスに「Apache24\_2」を登録します。(“△” は半角スペースを示します。)  
※Apache のサービス名は例です。

コマンド

```
httpd.exe△-k△install△-n△"Apache24_2"△-f△"C:/Apache24_2/conf/httpd.conf"
```



仕様補足

「Apache24\_2」を登録する前に、既存で運用している Apache が稼働中の場合、Windows のサービスから該当 Apache を停止してから登録してください。

- 「スタート」メニュー>「管理ツール」>「サービス」を選択し、「サービス」画面を表示します。  
「Apache24\_2」がサービスに登録されていることを確認します。

### 3. 追加 WEBAPI と Apache の接続設定

---

本マニュアルの「1.5.3 WEB サーバーと WEB アプリケーションの接続設定」の手順に従い、追加作成した「WEBAPI」と「Apache」の接続設定を行ってください。

以上で「1.5.5 WEBAPI の複数導入設定」は終了です。

## 1.6 API連携機能 利用ガイド

---

API 連携機能の利用方法については Senju Service Manager API リファレンスを参照してください。

## 1.7 バージョンアップ

### 1.7.1 2023.0.0 へバージョンアップ

2022.0.0.0 環境から 2023.0.0 環境へアップデートする際の環境設定手順について説明します。

#### 1.7.1.1 2022.0.0.0 環境のアンインストール

1. API サーバーに管理者権限のアカウントでログインします。
2. コマンドプロンプトを新規に起動し、下記のコマンドを実行します。（“△” は半角スペースを示します。）

コマンド
cd△C:¥[仮想環境用のフォルダ名] .env¥Scripts¥activate

3. 仮想環境で下記のコマンドを実行します。（“△” は半角スペースを示します。）

コマンド
pip△uninstall△-y△ssmwebapi

以上で WEB アプリケーションのアンインストールは終了です。

#### 1.7.1.2 2023.0.0 API 連携機能の導入

本章の手順は「1.5 API 連携機能の導入」を参照してください。

### 1.7.2 Apache のバージョンアップ

Apache のバージョンをアップデートする際の設定手順について説明します。

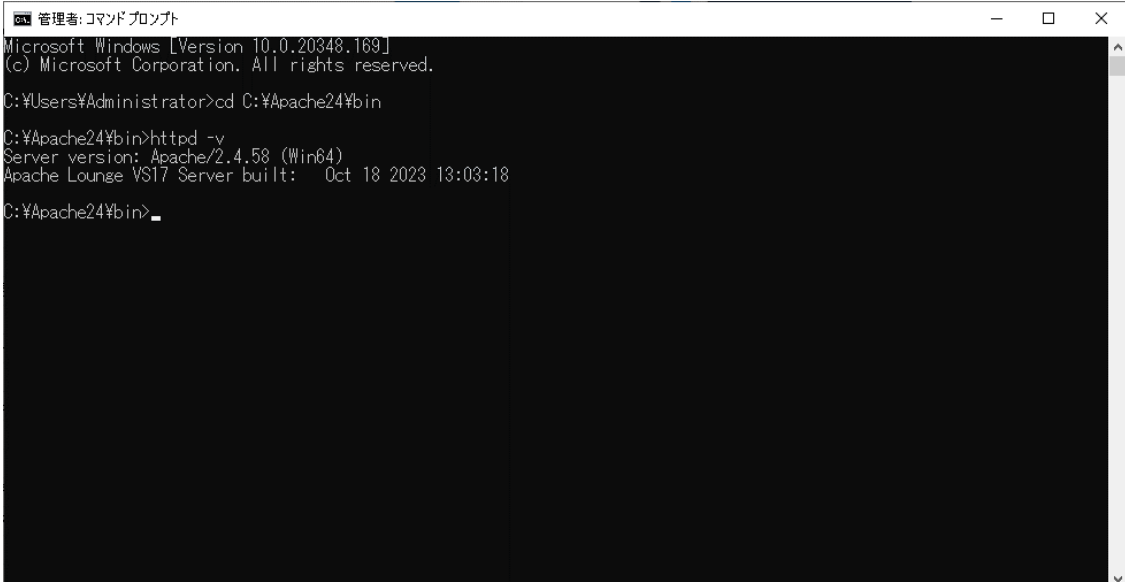
例として Apache 2.4.48 から Apache 2.4.58 にバージョンアップする手順を記載します。

1. 新バージョンの zip ファイルを入手し、インストールするサーバーの C:\TEMP に格納します。
  - ・ダウンロード先：<https://www.apachelounge.com/download/>
  - ・ファイル名：`httpd-2.4.58-win64-VS17.zip`
2. インストールするサーバーに管理者権限のアカウントでログインします。

- 「スタート」メニュー>「管理ツール」>「サービス」を選択し、「サービス」画面を表示します。
- 「Apache2.4」を右クリックし、「停止(O)」をクリックします。
- 旧バージョンの「Apache24」フォルダの名前を別名に変更します。  
例：Apache24→Apache24\_2.4.48
- 新バージョンの zip ファイルを展開し、「Apache24」フォルダをバージョンアップ前と同じ場所に格納します。
- 新バージョンの Apache の設定ファイル(httpd.conf)を旧バージョンの記載内容を転記します。  
また、SSL 通信利用中の場合は httpd-ssl.conf も同様に記載内容を転記します。
- 「スタート」メニュー>「管理ツール」>「サービス」を選択し、「サービス」画面を表示します。
- 「Apache2.4」を右クリックし、「開始(S)」をクリックします。
- コマンドプロンプトを新規に起動し、下記のコマンドを実行します。(“△”は半角スペースを示します。)

コマンド
cd△C:¥Apache24¥bin
httpd -v

- バージョンアップ後のバージョンを確認します。



```
管理: コマンドプロンプト
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\Apache24\bin

C:\Apache24\bin>httpd -v
Server version: Apache/2.4.58 (Win64)
Apache Lounge VS17 Server built: Oct 18 2023 13:03:18

C:\Apache24\bin>
```

- 任意のエンドポイントの API の実行に成功することを確認します。

13. 別名に変更した Apache のフォルダを削除します。

以上で Apache のバージョンアップは終了です。



## 1.8 制限事項

Senju Service Manager で提供する API 連携機能について制限事項を以下に示します。

- 1) API サーバーを SSMWEB サーバーと別筐体に構築することはできません。  
別筐体に構築した場合、Senju Service Manager の API を実行すると 500 Internal Server Error のレスポンスが返却されます。
- 2) 認証方式が SSM 基本認証以外のユーザーは Senju Service Manager の API を発行することはできません。
- 3) ユーザーが複数のアクセストークンを発行することはできません。  
以下の API が実行された場合、API 実行前のアクセストークンは取り消され使用できなくなります。
  - ・ **アクセストークン・リフレッシュトークンの発行**
  - ・ **アクセストークンの更新**
- 4) Senju Service Manager のパスワード期限切れ時、**アクセストークン・リフレッシュトークンの発行** API のレスポンスは、制御情報の設定値に従います。
  - ・ サービスデスクユーザーの場合

管理項目名	選択値	アクセストークン・リフレッシュトークンの発行
パスワード期限切れ時の動作 (サービスデスクユーザー)	変更しない(ログインする)	○：発行する
	変更しない(ログインしない)	×：発行しない
	変更する	×：発行しない

- ・ エンドユーザーの場合

管理項目名	選択値	アクセストークン・リフレッシュトークンの発行
パスワード期限切れ時の動作 (エンドユーザー)	変更しない(ログインする)	○：発行する
	変更しない(ログインしない)	×：発行しない
	変更する	×：発行しない



仕様補足

API は画面を持たないため、Senju Service Manager 画面のように強制的にパスワードを変更させ変更後のパスワードでログインするのと同等の処理が行えません。

そのため、管理項目の選択値が「変更する」の場合は認証失敗とみなしアクセストークン・リフレッシュトークンを発行しません。

上記の場合、Senju Service Manager 画面からパスワード変更を行ってください。

- 5) API サーバーと SSMWEB サーバーで使用している WEB サーバーが異なるため、Senju Service Manager 画面と Senju Service Manager の API で制御情報の設定値が反映されるタイミングが異なります。
- 6) SSL 通信を行う場合、SSM WEB サーバーには認証局から発行された証明書を利用してください

い。認証局から発行されていない証明書を利用した場合、Senju Service Manager の API を実行すると 500 Internal Server Error のレスポンスが返却されます。

- 7) API のリクエストボディに Shift-JIS 外の文字を入力すると、HTML エンコードされた状態に変換されます。また、Windows-31J (MS932) では文字コード範囲内の文字（高崎纘①(株)\*、…）もエンコードされてしまいます。
- 8) リッチテキスト項目には 1000 万バイトまで入力することが可能ですが、この上限値は指定された情報をエスケープ処理した後のサイズとなります。記号等を含む場合には余裕を持って指定してください。指定したファイルのサイズが上限を超えた場合、該当項目は登録及び更新されません。
- 9) リッチテキスト項目に指定した値を入れて API を実行すると、レスポンスされる値に特定 HTML のタグが付与されます。
- 10) リッチテキスト項目を入力する場合、制御情報「イベント連携のリッチテキスト項目の入力チェック設定」の設定値について以下の仕様になります。
  - ・「チェックしない」を指定した場合、許可しないタグが含まれているとサニタイズします。
  - ・「チェックする」を指定した場合、許可しないタグが含まれていると 500 エラーとなります。
- 11) API でリッチテキスト項目値の先頭に空白を入力して実行した場合、先頭の空白は削除されずにそのまま入力されます。
- 12) リッチテキスト項目に設定した値の中に Shift\_JIS で表示できない(機種依存)文字が含まれる場合、入力形式が HTML 形式の場合、画面表示時には正常に入力した文字形式で表示されます。ただし、入力形式がテキスト形式の場合、画面上からデコードされた文字が表示されることがあります。
- 13) プロセス承認 WEBAPI を利用されている場合は、メール送信時の接続形態で以下の設定値は利用できません。
  - ・管理者メニュー > 制御情報
    - メニュー区分：共通
    - 制御グループ：共通

管理項目名	設定値
メール送信時の接続形態	SMTP (STARTTLS) + SMTP 認証を使用しない SMTP (STARTTLS) + SMTP 認証を使用する

以下の設定値は利用可能となります。

- SMTP+SMTP 認証を使用しない
- SMTP+SMTP 認証を使用する
- SMTPs+SMTP 認証を使用しない
- SMTPs+SMTP 認証を使用する