

ASTERIA CELF Adapter *for EAI* 操作ガイド

SCSK SCSK株式会社

第 1. 0 版

目次

1. はじめに.....	2
1.1. 本書について.....	2
1.2. 動作環境.....	3
2. 機能概要.....	4
2.1. CELF アクション.....	4
3. 作成方法.....	5
3.1. CELF アクションを用いた ASTERIA とのサンプル連携 (XML 形式)	5
3.1.1. サンプルシナリオ.....	5
3.1.2. 事前準備.....	5
3.1.3. シートの作成.....	5
3.1.4. CELF から呼び出されるフローの作成.....	13
3.2. CELF アクションを用いた ASTERIA とのサンプル連携 (JSON 形式)	21
3.2.1. サンプルシナリオ.....	21
3.2.2. 事前準備.....	21
3.2.3. シートの作成.....	21
3.2.4. CELF から呼び出されるフローの作成.....	29
4. 操作方法.....	36
4.1. CELF からのテスト実行.....	36
4.1.1. テスト実行.....	36

1. はじめに

1.1. 本書について

本書では、ASTERIA CELF Adapter *for EAI* をインストールするユーザーを対象として、ASTERIA CELF Adapter *for EAI* の設定方法、操作方法について説明します。

1.2. 動作環境

対象 OS

ASTERIA CELF Adapter *for EAI*は以下の OS での動作をサポートしています。

- Microsoft Windows Server 2003 R2 以降 (x86)
- Microsoft Windows Server 2008 (x64)、R2 (x64)
- Microsoft Windows Server 2012 (x64)
- Microsoft Windows 7 (x64)
- Red Hat Enterprise Linux 5 (x64)
- Red Hat Enterprise Linux 6 (x64)
- Cent OS 6 (x64)

ASTERIA WARP バージョン

ASTERIA CELF Adapter *for EAI*は以下のバージョンの ASTERIA が導入されている環境を前提としています。

- ASTERIA WARP Standard Edition 4.9.1、1610 以降
- ASTERIA WARP Enterprise Edition 4.9.1、1610 以降

対象 CELF バージョン

ASTERIA CELF Adapter Option *for EAI*は以下バージョンの CELF バージョンをサポートしています。

- 1.0 以降

CELF 拡張オプション

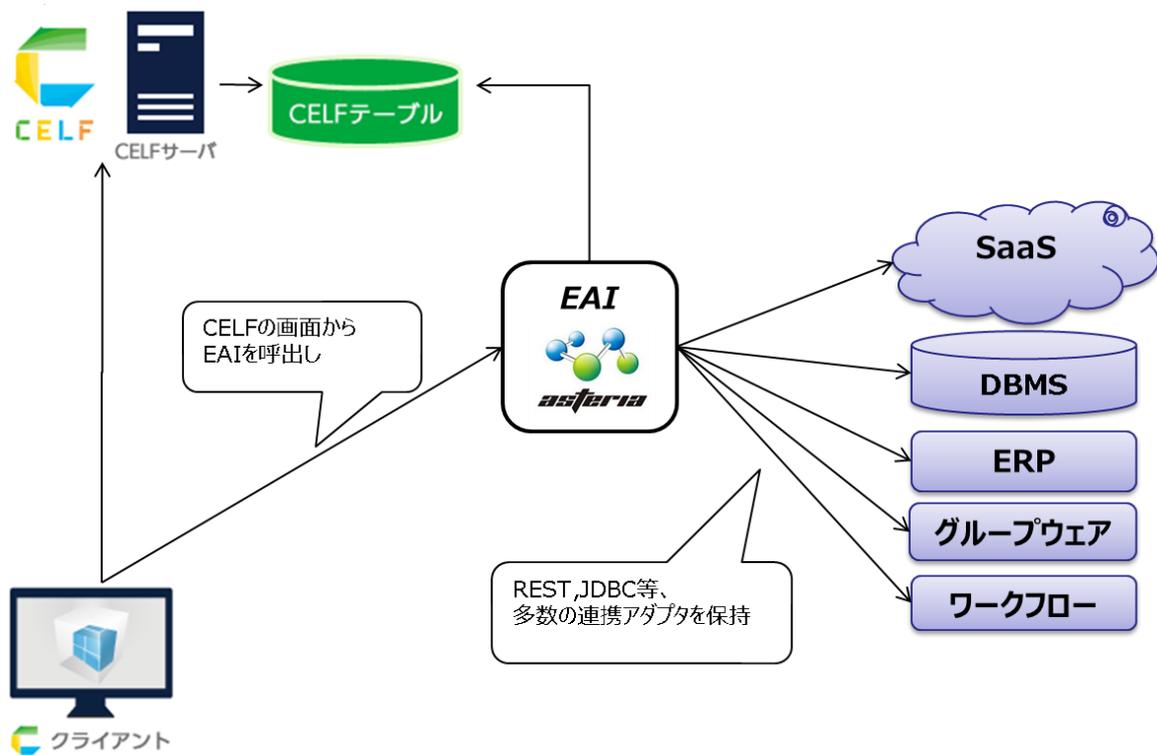
CELF の拡張オプションに以下のファイルが登録されていることを前提とします。

- celf-asteria-extension.zip

2. 機能概要

2.1. CELF アクション

ASTERIA CELF Adapter for EAIは、CELFのASTERIA専用アクションから、ASTERIAに対してHTTPリクエストを行う機能です。



3. 作成方法

3.1. CELF アクションを用いた ASTERIA とのサンプル連携 (XML 形式)

3.1.1. サンプルシナリオ

CELF 上の画面の「入力データ」に値を入力し、「実行」ボタンを押下すると、実行時刻と「入力データ」に入力した値と 1 から 9 までの数字を足した値を表示するサンプル連携を実装する手順を以下に示します。

■ 処理概要



3.1.2. 事前準備

- ① 以下の設定値を予め設定しておきます。

URL 名 : ASTERIA のフローを実行する際の URL

3.1.3. シートの作成

- ① 「企業 ID」、「ユーザーID」、「パスワード」を入力し、CELF にログインします。

CELF

企業ID

ユーザーID

パスワード

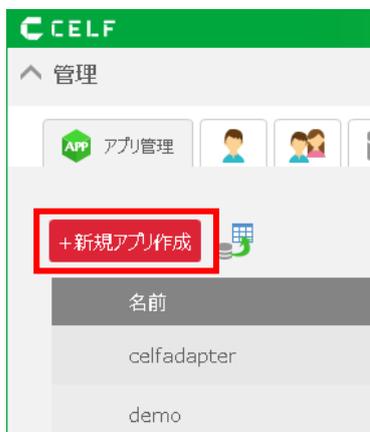
ログイン

CELF©2017

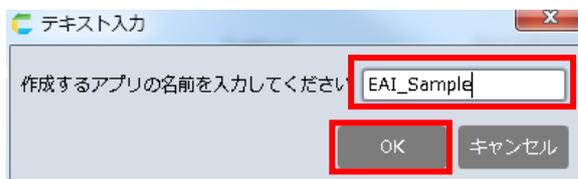
- ② 左上の「管理」を押下します。



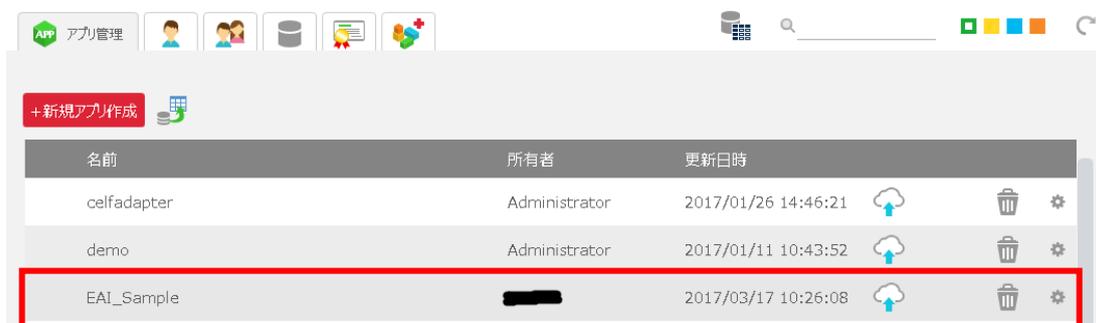
- ③ 「+新規アプリ作成」を押下します。



- ④ アプリ名を設定します。



- ⑤ アプリ一覧画面に作成したアプリが追加されていることを確認し、追加したアプリを押下します。



- ⑥ 「+新規シート作成」を押下します。



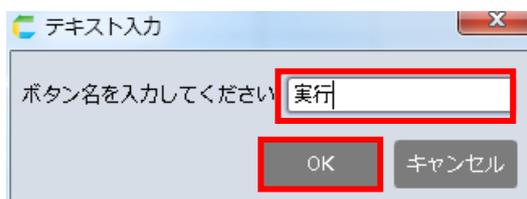
- ⑦ 以下のようなシートを作成します。

	A	B	C	D	E	F
1		CELL-ASTERIA連携サンプル				
2						
3		入力データ				
4						
5		実行時刻				
6		出力データ				
7						
8						
9						

- ⑧ ボタンを設定したいセルを選択した状態で、メニュータブの「入力形式」の「ボタン」を押下します。



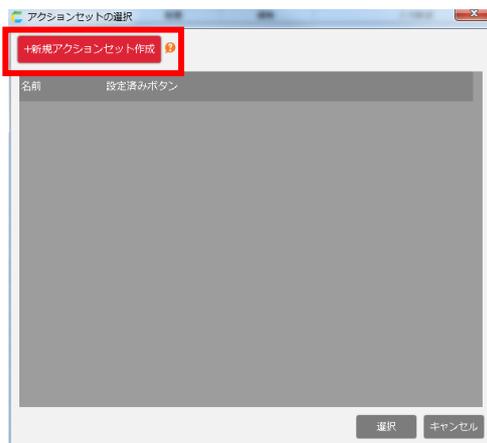
- ⑨ ボタン名を入力し、「OK」を押下します。



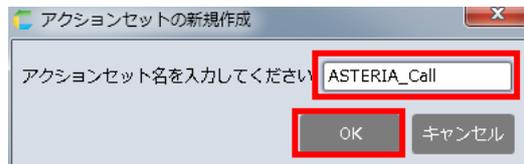
- ⑩ 上記で設定したボタンを押下し、対象のボタンにアクションセットを設定します。

	A	B	C	D	E	F
1		CELf-ASTERIA連携サンプル				
2						
3		入力データ			実行	
4						
5		実行時刻				
6		出力データ				
7						
8						
9						
10						

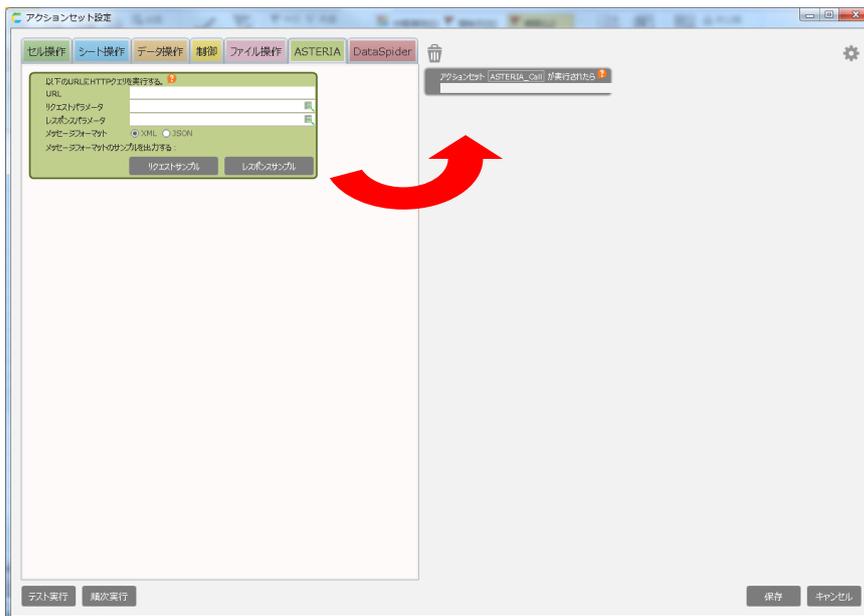
- ⑪ 「+新規アクションセット作成」を選択します。



- ⑫ 「アクションセット名」を入力し、「OK」を押下します。



- ⑬ 「ASTERIA」 タブを選択し、アクションをアクションセットにドラッグ&ドロップします。



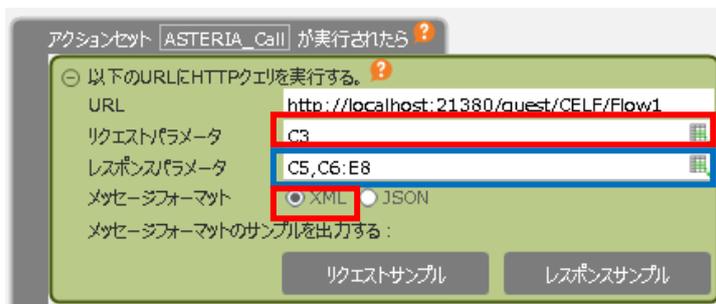
- ⑭ 「3.1.2 事前準備」にて決めた値を基に、各の値を入力し、「リクエストサンプル」を押下します。

URL : http://[ASTERIA サーバホスト]:[ポート番号]/[オーナー]/[URL]

リクエストパラメータ : ASTERIA への入力値を入力するセル

レスポンスパラメータ : ASTERIA からの出力値を出力するセル

メッセージフォーマット : XML

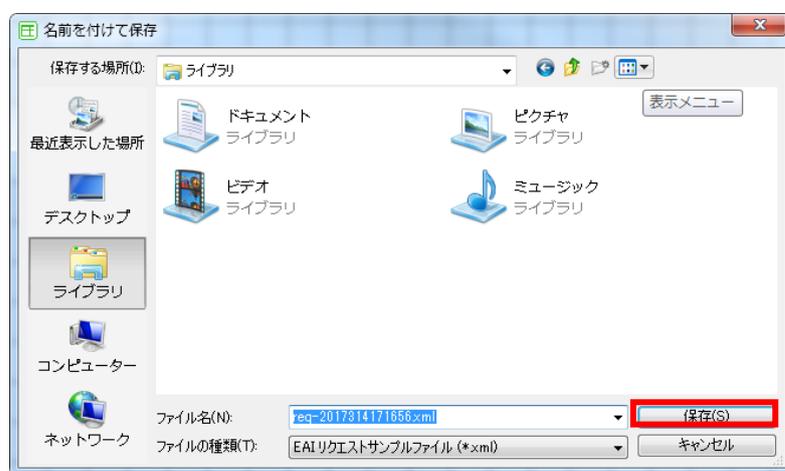


※サンプルの場合、リクエストパラメータとレスポンスパラメータは以下の通りとなります。

	A	B	C	D	E	F
1		CELLF-ASTERIA連携サンプル				
2			リクエストパラメータ			
3		入力データ	1		実行	
4						
5		実行時刻				
6		出力データ				
7						
8						
9						
10						

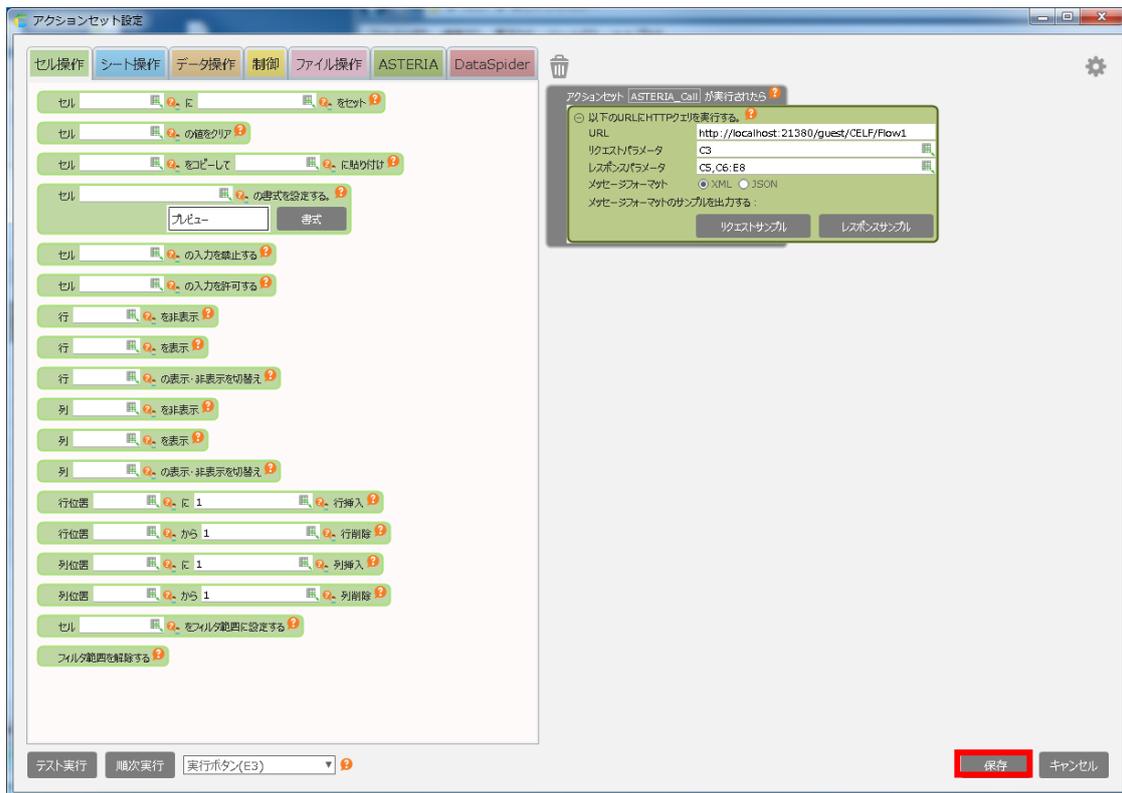
レスポンスパラメータ

- ⑮ 「保存」を押下し、リクエストパラメータのサンプルデータをファイル出力します。

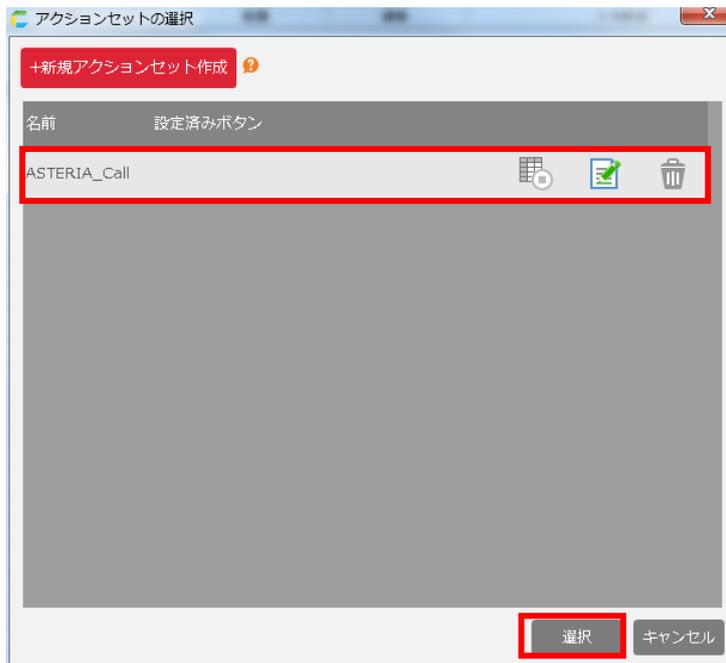


- ⑯ 同様に、「レスポンスサンプル」についてもファイルを出力します。

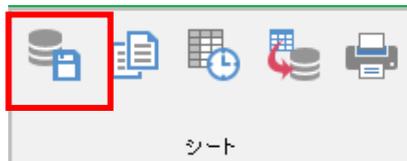
⑰ 元の画面にて、「保存」を押下します。



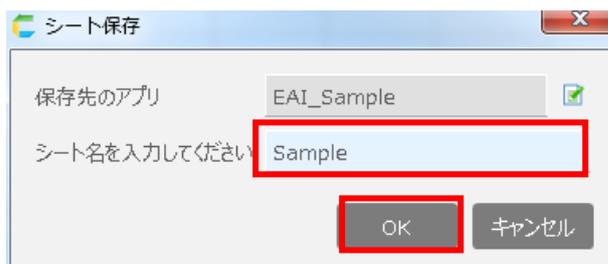
- ⑱ 上記で作成したアクションセットを選択し、「選択」を押下します。



- ⑲ メニュータブの「シート」の「保存」を押下し、作成したシートを保存します。

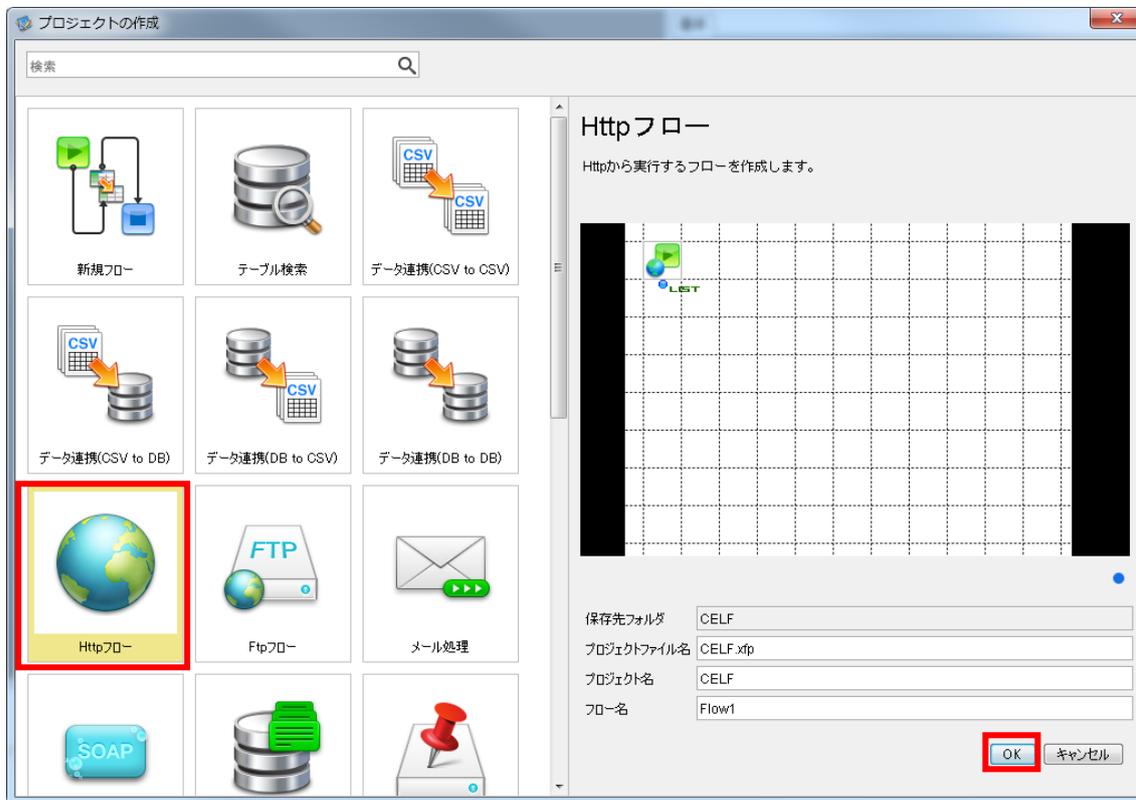


- ⑳ シート名を設定し、「OK」を押下します。

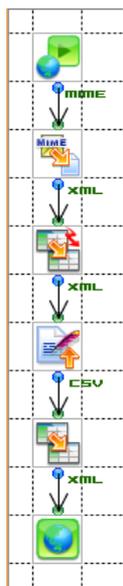


3.1.4. CELF から呼び出されるフローの作成

- ① ASTERIA フローデザイナーにログインし、「プロジェクトの作成」から「HTTP フロー」を選択し、「OK」を押下します。



- ② 以下のようなフローを作成します。各コンポーネントのプロパティは以下の通りです。



I. HTTPStart コンポーネント

基本	
プロパティ名	値
名前	Flow1
タイムアウト (秒)	180
汎用エラー処理	(なし)
実行を許可するユーザー	
<input type="checkbox"/> アプリケーションログ設定	
└ ログ設定名	(なし)
└ ログレベル	システム設定
└ システムログに出力	はい
セッション	保持する
HTTPでの呼出しを許可	はい

II. MIMEDecord コンポーネント

基本	
プロパティ名	値
名前	MIMEDecode1
処理対象とする条件式	
取得対象とする条件式	part[0]
ループを開始	いいえ
エラー処理	
汎用	(なし)

III. Mapper コンポーネント①

基本		Mapper変数
プロパティ名	値	
名前	Mapper1	
ループを開始	いいえ	
入力をそのまま出力	いいえ	
入力が無い場合エラー	いいえ	
エラー処理		
汎用	(なし)	

IV. Velocity コンポーネント

基本		ローカル変数	拡張クラス
プロパティ名	値		
名前	Velocity1		
テンプレートの指定方法	直接入力		
テンプレートの内容	1,2,3,4,5,6,7,8,9,		
XPath式のプレフィックス	fv		
エラー処理			
汎用	(なし)		

V. Mapper コンポーネント②

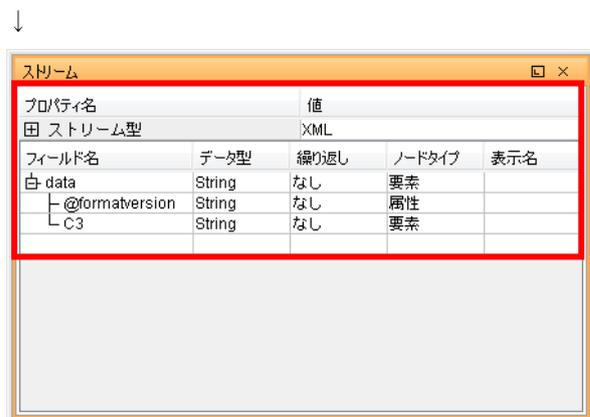
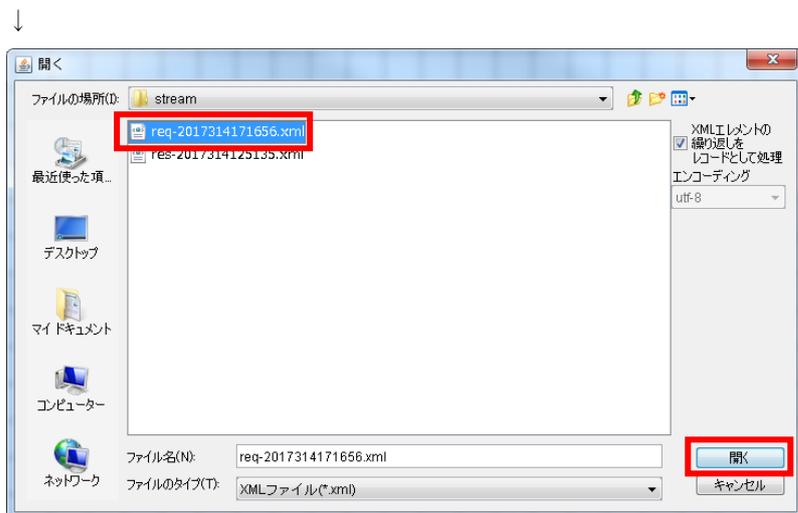
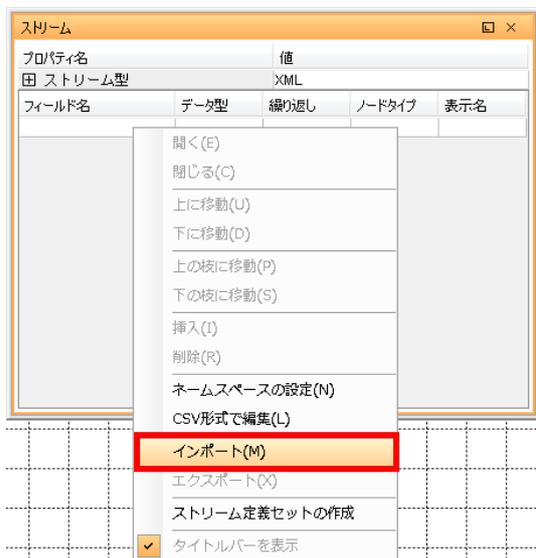
基本		Mapper変数
プロパティ名	値	
名前	Mapper2	
入力をそのまま出力	はい	
入力が無い場合エラー	いいえ	
エラー処理		
汎用	(なし)	

VI. HTTPEnd コンポーネント

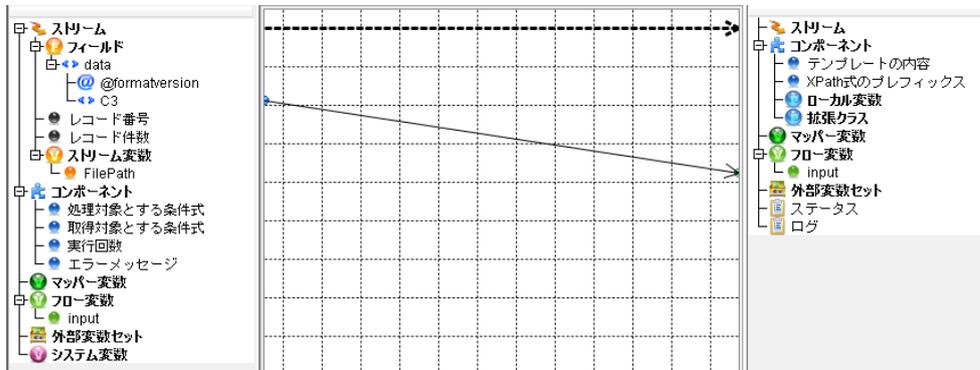
基本		Cookie	追加するヘッダー
プロパティ名	値		
名前	HttpEnd1		
トランザクション	コミット		
ステータスコード	200 OK		
Content-Type	自動判別		
田 ダウンロード情報			
JSON変換	しない		

基本		Cookie	追加するヘッダー
ヘッダー名	データ型	値	
Cache-Control	String	no-cache	

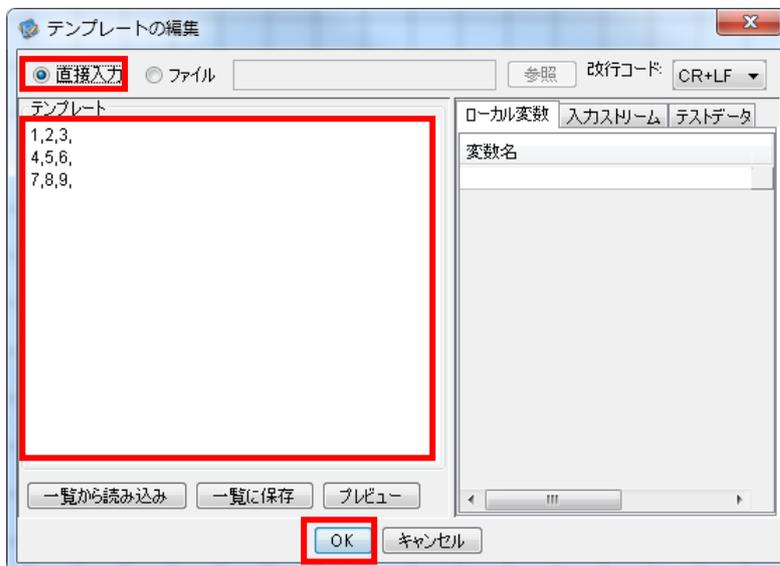
- ③ MIMEDecode コンポーネントの、「ストリームペイン」の右クリックメニューから「インポート」を選択し、「3.1.3 ⑮」にて出力したファイルを選択し、「開く」を押下し、CELLF からのリクエストパラメータの形式を設定します。



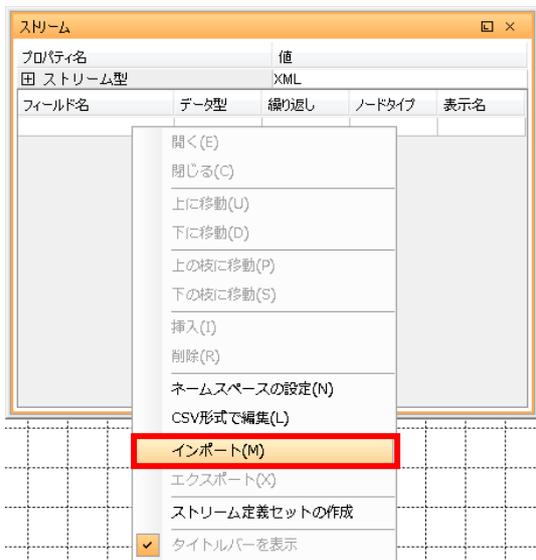
- ④ Mapper コンポーネント①で以下のようにマッピングし、リクエストパラメータから取得した値をフロー変数 (input) へ格納します。



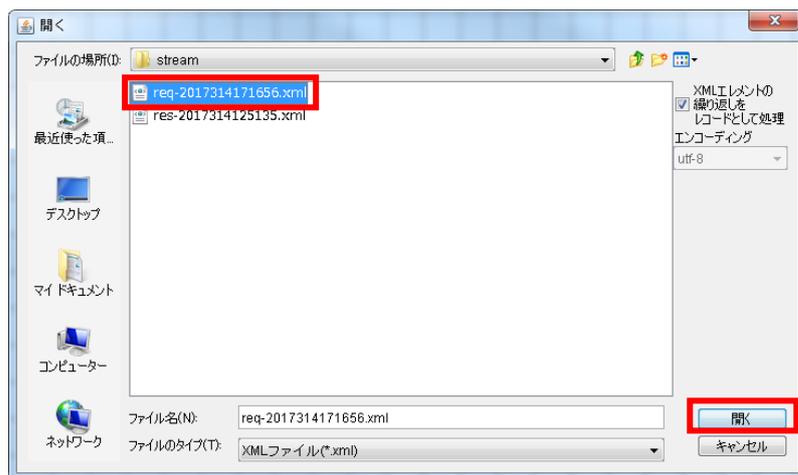
- ⑤ Velocity コンポーネントに、以下のように1から9までの数字を定義します。



- ⑥ Mapper コンポーネント②で、「ストリームペイン」の右クリックメニューから「インポート」を選択し、「3.1.3 ⑩」にて出力したファイルを選択し、「開く」を押下し、CELF からのレスポンスパラメータの形式を設定します。



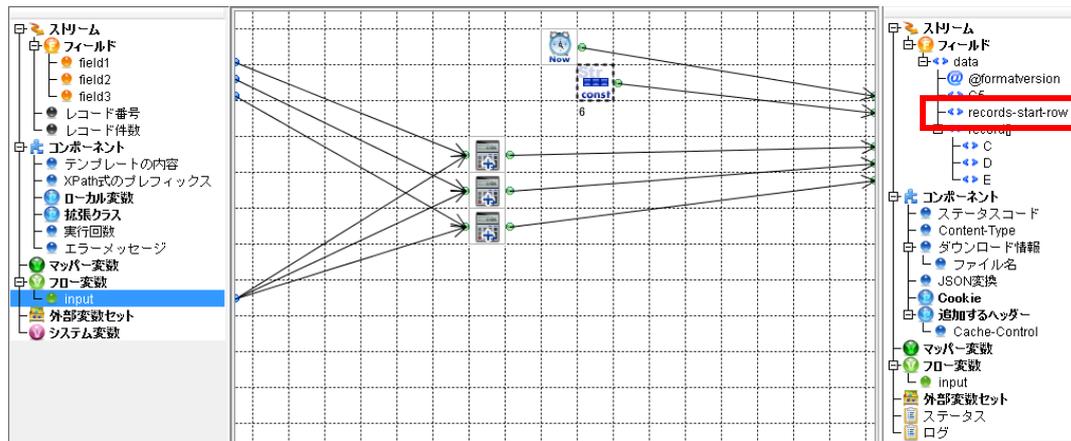
↓



↓



⑦ レスポンスパラメータの形式をインポートした後、以下のようにマッピングします。



※ 「records-start-row」 は繰り返し項目が始まる行を指定します。

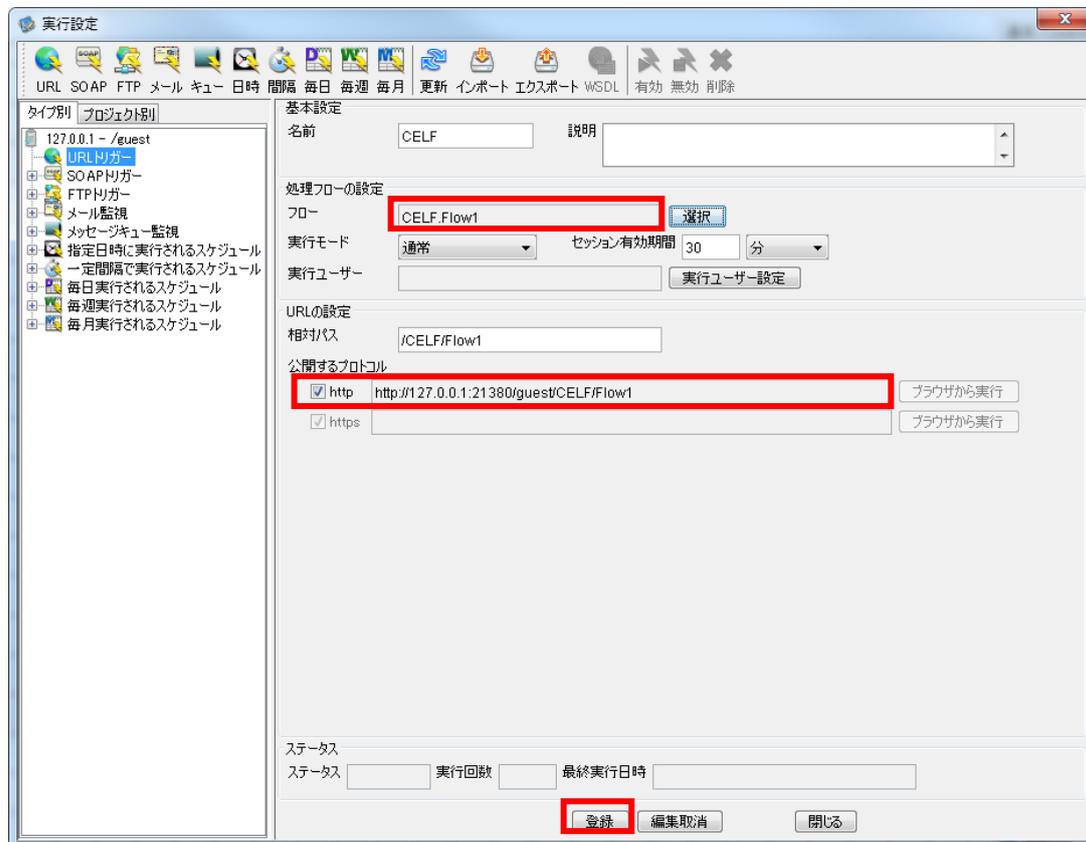
サンプルの例では 6 行目から繰り返し項目が始まるため、「6」を代入しています。

CELf-ASTERIA連携サンプル			
入力データ	1		実行
実行時刻	2017-03-16T17:51:28.881 IST		
出力データ	2	3	4
	5	6	7
	8	9	10

← 6 行目

⑧ フローを保存します。

- ⑨ 上記で作成したフローを URL トリガーに設定します。ここで、「公開するプロトコル」が「3.1.3 ⑭」で設定した URL と同じになるように設定し、登録します。

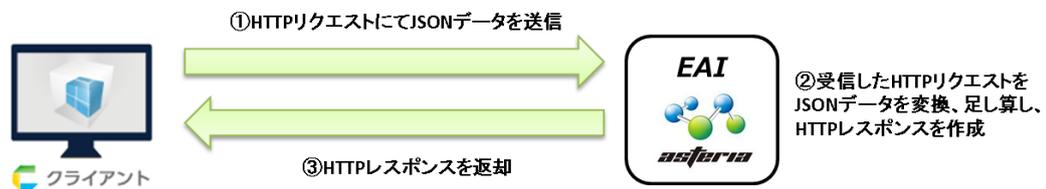


3.2. CELF アクションを用いた ASTERIA とのサンプル連携 (JSON 形式)

3.2.1. サンプルシナリオ

CELF 上の画面の「入力データ」に値を入力し、「実行」ボタンを押下すると、実行時刻と「入力データ」に入力した値と 1 から 9 までの数字を足した値を表示するサンプル連携を実装する手順を以下に示します。

■ 処理概要



3.2.2. 事前準備

- ① 以下の設定値を予め設定しておきます。

URL 名 : ASTERIA のフローを実行する際の URL

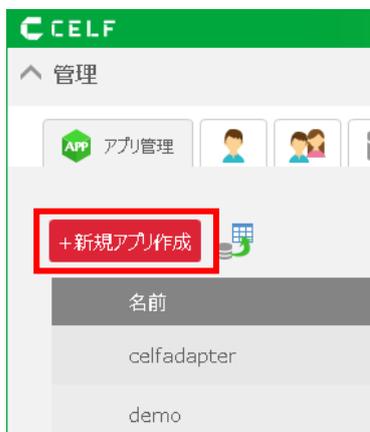
3.2.3. シートの作成

- ① 「企業 ID」、「ユーザーID」、「パスワード」を入力し、CELF にログインします。

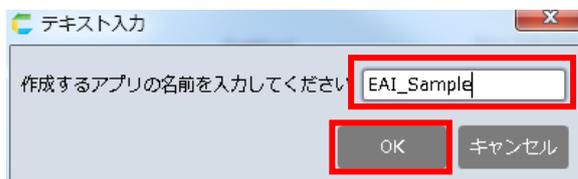
- ② 左上の「管理」を押下します。



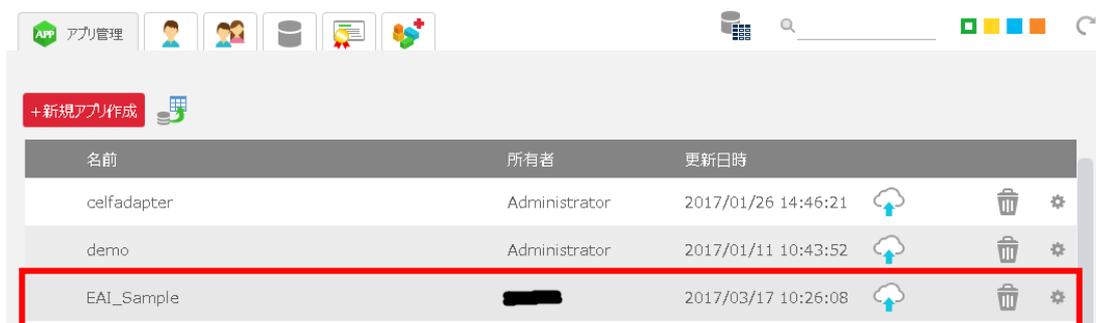
- ③ 「+新規アプリ作成」を押下します。



- ④ アプリ名を設定します。



- ⑤ アプリ一覧画面に作成したアプリが追加されていることを確認し、追加したアプリを押下します。



- ⑥ 「+新規シート作成」を押下します。



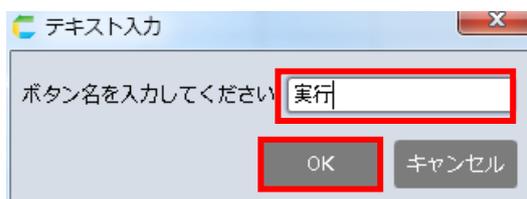
- ⑦ 以下のようなシートを作成します。

	A	B	C	D	E	F
1		CELL-ASTERIA連携サンプル				
2						
3		入力データ				
4						
5		実行時刻				
6		出力データ				
7						
8						
9						

- ⑧ ボタンを設定したいセルを選択した状態で、メニュータブの「入力形式」の「ボタン」を押下します。



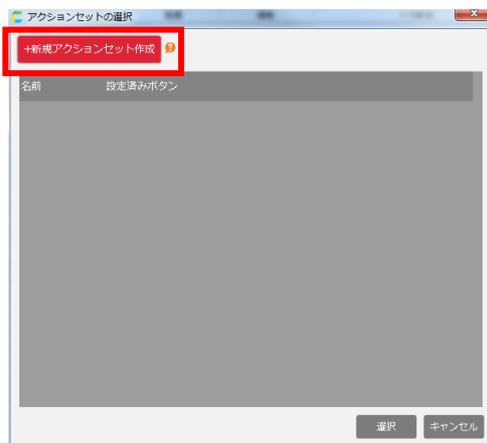
- ⑨ ボタン名を入力し、「OK」を押下します。



- ⑩ 上記で設定したボタンを押下し、対象のボタンにアクションセットを設定します。

	A	B	C	D	E	F
1		CELF-ASTERIA連携サンプル				
2						
3		入力データ			実行	
4						
5		実行時刻				
6		出力データ				
7						
8						
9						
10						

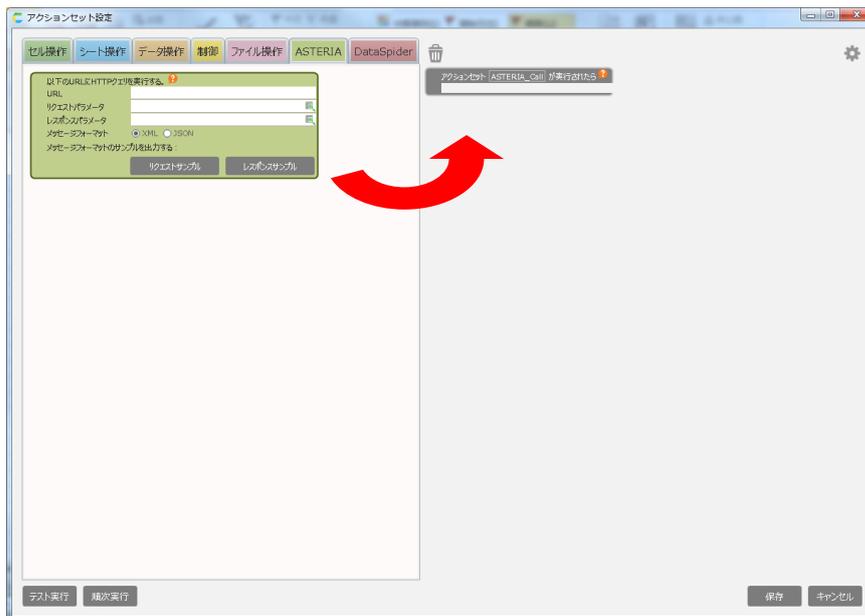
- ⑪ 「+新規アクションセット作成」を選択します。



- ⑫ 「アクションセット名」を入力し、「OK」を押下します。



- ⑬ 「ASTERIA」 タブを選択し、アクションをアクションセットにドラッグ&ドロップします。



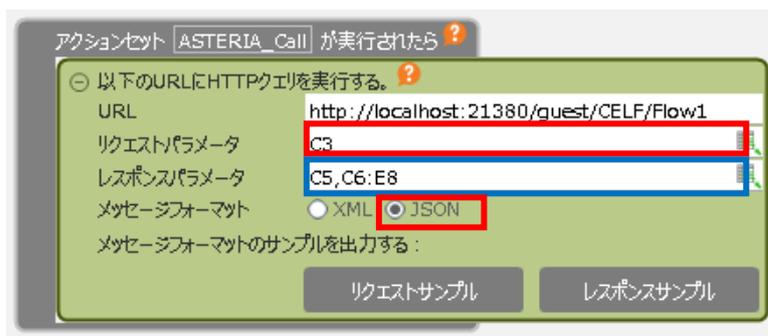
- ⑭ 「3.2.2 事前準備」にて決めた値を基に、各の値を入力し、「リクエストサンプル」を押下します。

URL : http://[ASTERIA サーバホスト]:[ポート番号]/[オーナー]/[URL]

リクエストパラメータ : ASTERIA への入力値を入力するセル

レスポンスパラメータ : ASTERIA からの出力値を出力するセル

メッセージフォーマット : JSON



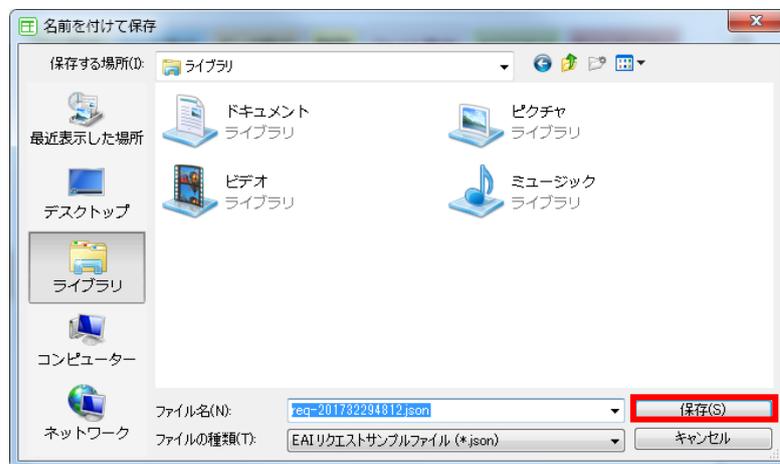
※サンプルの場合、リクエストパラメータとレスポンスパラメータは以下の通りとなります。

	A	B	C	D	E	F
1		CELLF-ASTERIA連携サンプル				
2						
3		入力データ	1		実行	
4						
5		実行時刻				
6		出力データ				
7						
8						
9						
10						

リクエストパラメータ

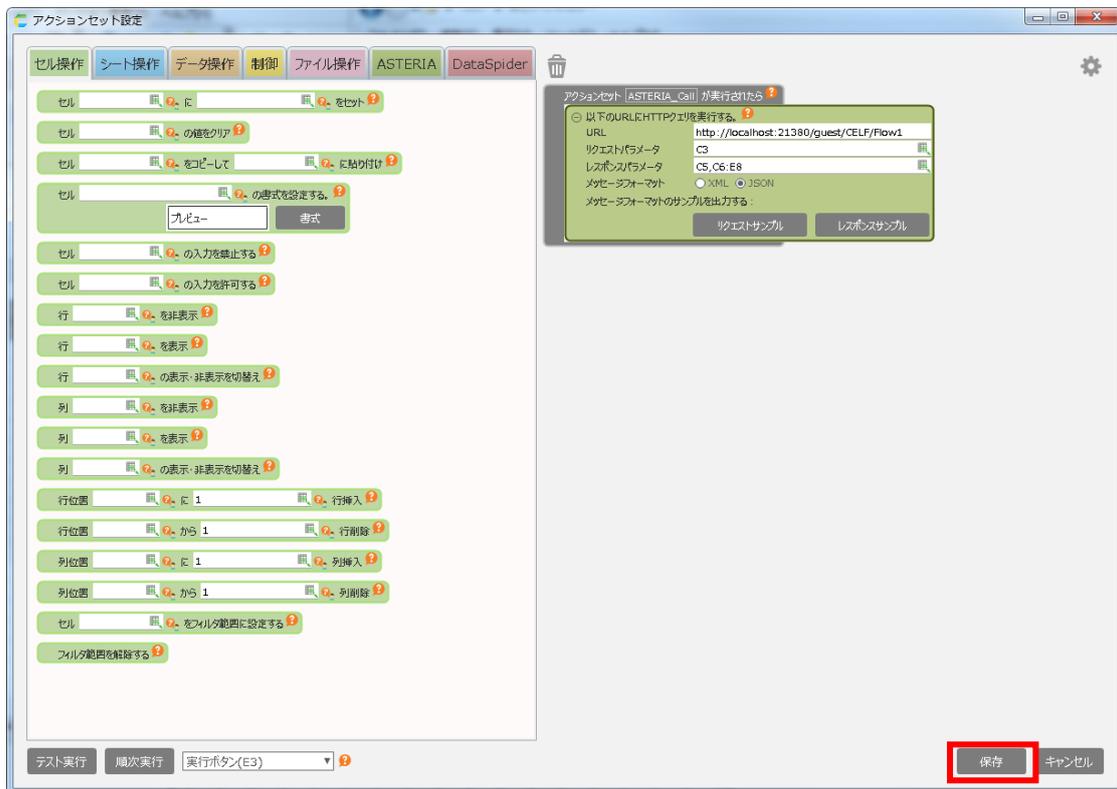
レスポンスパラメータ

- ⑮ 「保存」を押下し、リクエストパラメータのサンプルデータをファイル出力します。

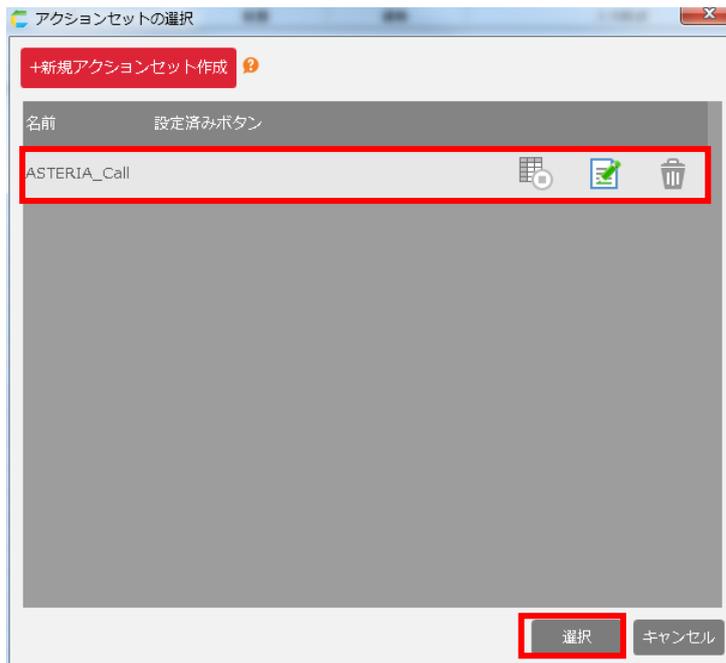


- ⑯ 同様に、「レスポンスサンプル」についてもファイルを出力します。

⑰ 元の画面にて、「保存」を押下します。



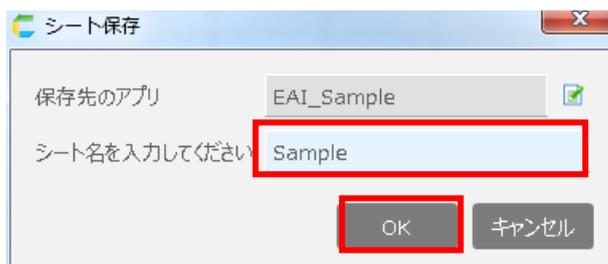
- ⑱ 上記で作成したアクションセットを選択し、「選択」を押下します。



- ⑲ メニュータブの「シート」の「保存」を押下し、作成したシートを保存します。

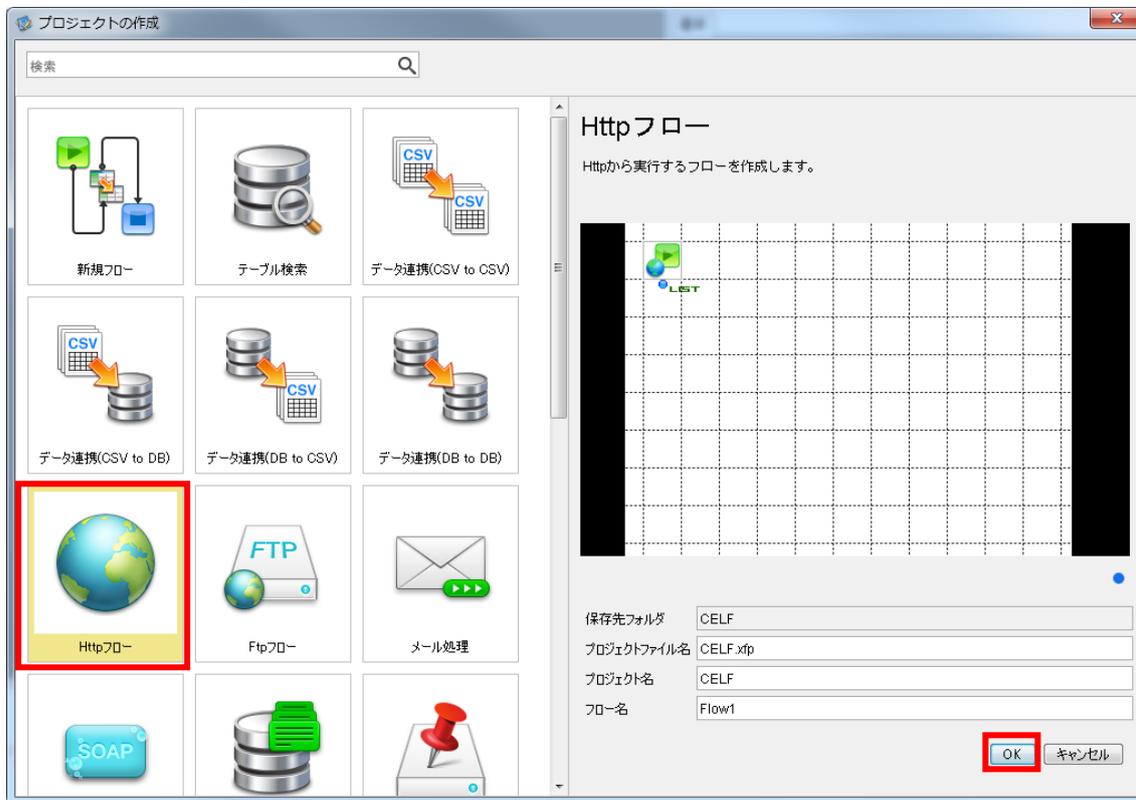


- ⑳ シート名を設定し、「OK」を押下します。

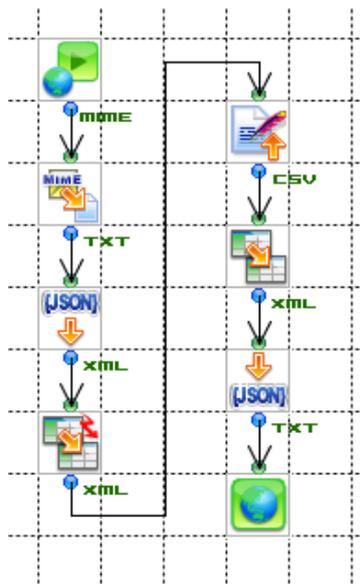


3.2.4. CELF から呼び出されるフローの作成

- ① ASTERIA フローデザイナーにログインし、「プロジェクトの作成」から「HTTP フロー」を選択し、「OK」を押下します。



- ② 以下のようなフローを作成します。各コンポーネントのプロパティは以下の通りです。



I. HTTPStart コンポーネント

基本	
プロパティ名	値
名前	Flow1
タイムアウト (秒)	180
汎用エラー処理	(なし)
実行を許可するユーザー	
<input type="checkbox"/> アプリケーションログ設定	
<input type="checkbox"/> ログ設定名	(なし)
<input type="checkbox"/> ログレベル	システム設定
<input type="checkbox"/> システムログに出力	はい
セッション	保持する
HTTPでの呼出しを許可	はい

II. MIMEDecord コンポーネント

基本	
プロパティ名	値
名前	MIMEDecode1
処理対象とする条件式	
取得対象とする条件式	part[0]
ループを開始	いいえ
エラー処理	
汎用	(なし)

III. JSONDecord コンポーネント

基本	
プロパティ名	値
名前	JSONDecode1
JSON種別	オブジェクト
ルート要素名	root
エラー処理	
汎用	(なし)

IV. Mapper コンポーネント①

基本		Mapper変数
プロパティ名	値	
名前	Mapper1	
ループを開始	いいえ	
入力をそのまま出力	いいえ	
入力が無い場合エラー	いいえ	
エラー処理		
汎用	(なし)	

V. Velocity コンポーネント

基本		ローカル変数	拡張クラス
プロパティ名	値		
名前	Velocity1		
テンプレートの指定方法	直接入力		
テンプレートの内容	1,2,3,4,5,6,7,8,9,		
XPath式のプレフィックス	fv		
エラー処理			
汎用	(なし)		

VI. Mapper コンポーネント②

基本		Mapper変数
プロパティ名	値	
名前	Mapper2	
入力そのまま出力	はい	
入力が無い場合エラー	いいえ	
エラー処理		
汎用	(なし)	

VII. JSONEncord コンポーネント

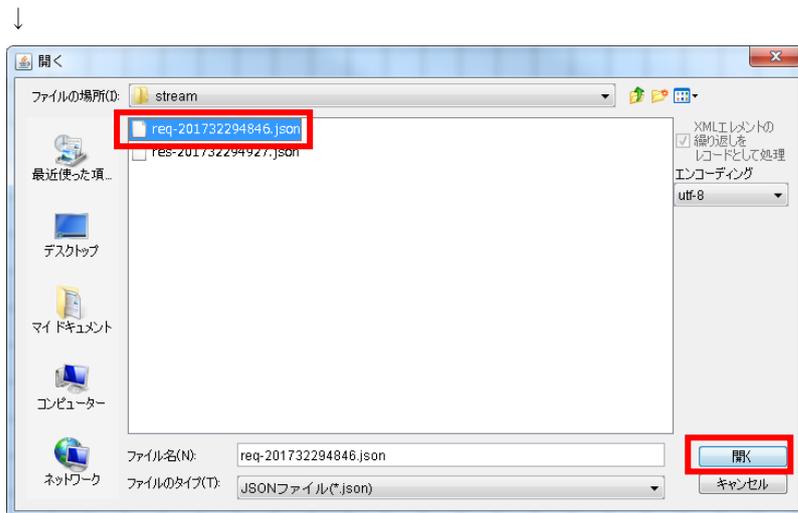
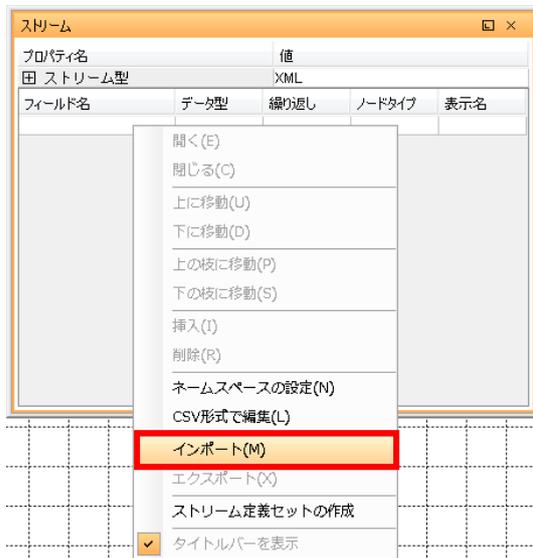
基本	
プロパティ名	値
名前	JSONEncode1
インデントする	はい
JSONトップレベル名	root
名前空間プレフィックスを削除する	はい
ルート要素を削除する	はい
エラー処理	
汎用	(なし)

VIII. HTTPEnd コンポーネント

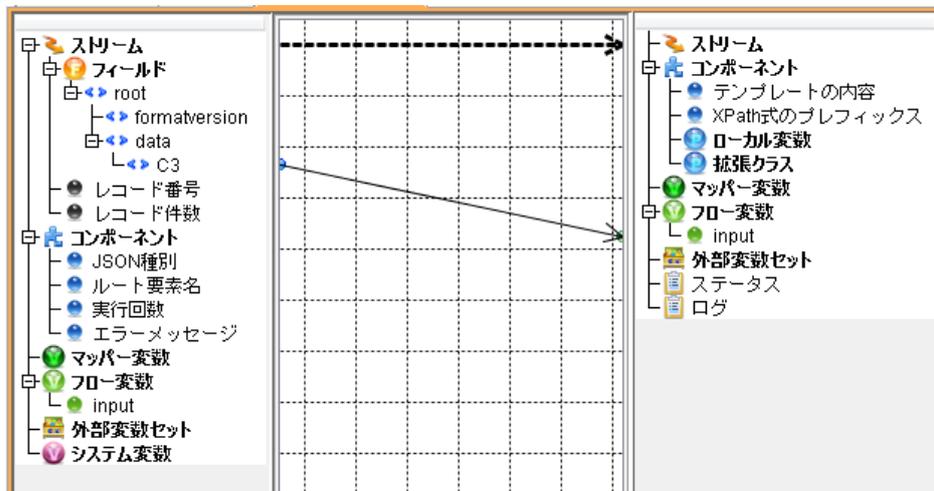
基本		Cookie	追加するヘッダー
プロパティ名	値		
名前	HttpEnd1		
トランザクション	コミット		
ステータスコード	200 OK		
Content-Type	自動判別		
田 ダウンロード情報			
JSON変換	しない		

基本		Cookie	追加するヘッダー
ヘッダー名	データ型	値	
Cache-Control	String	no-cache	

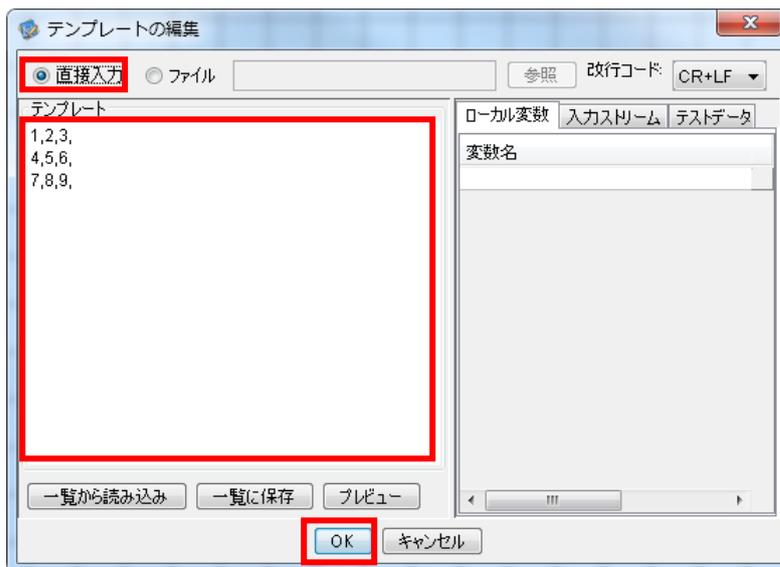
- ③ JSONDecode コンポーネントの「ストリームペイン」の右クリックメニューから「インポート」を選択し、「3.2.3 ⑮」にて出力したファイルを選択し、「開く」を押下し、CELF からのリクエストパラメータの形式を設定します。



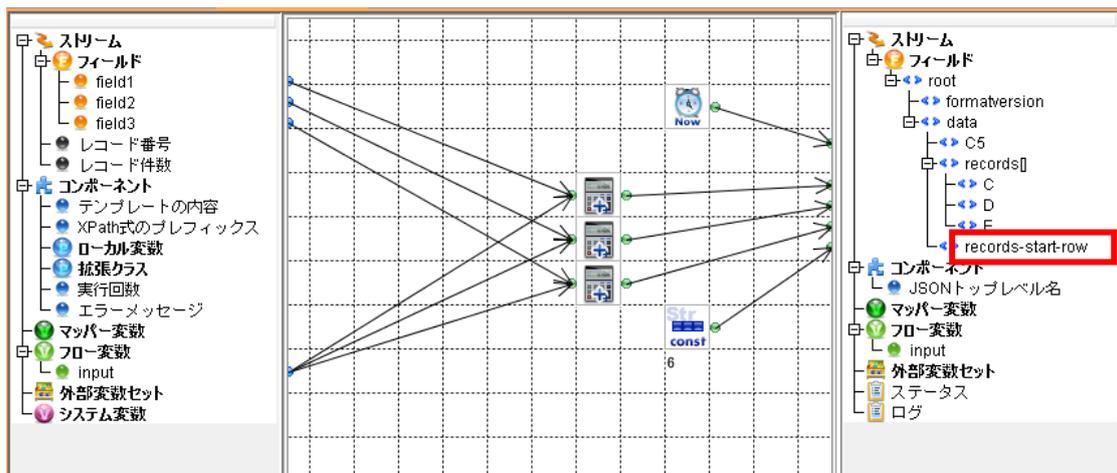
- ④ Mapper コンポーネント①で以下のようにマッピングし、リクエストパラメータから取得した値をフロー変数 (input) へ格納します。



- ⑤ Velocity コンポーネントに、以下のように1から9までの数字を定義します。



⑥ レスポンスパラメータの形式をインポートした後、以下のようにマッピングします。



※ 「records-start-row」 は繰り返し項目が開始される行を指定します。

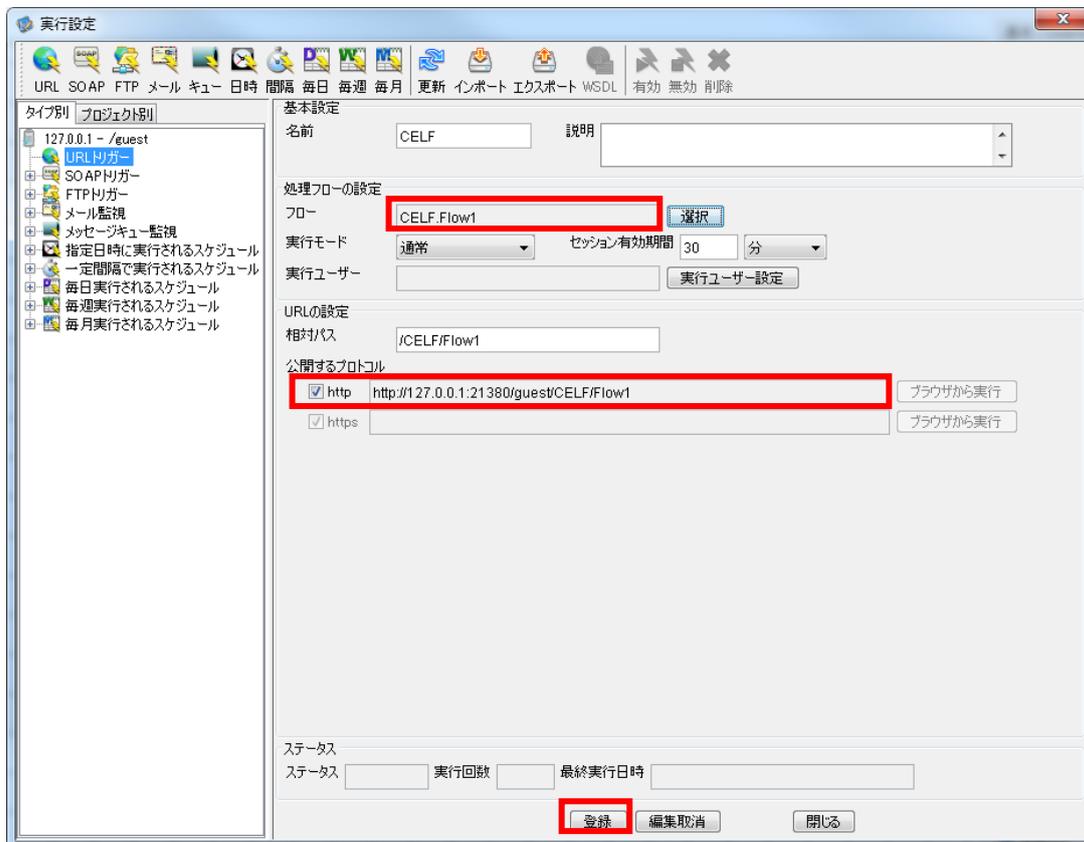
サンプルの例では6行目から繰り返し項目が開始されるため、「6」を代入しています。

CELf-ASTERIA連携サンプル			
入力データ	1		実行
実行時刻	2017-03-16T17:51:28.881 IST		
出力データ	2	3	4
	5	6	7
	8	9	10

← 6行目

⑦ フローを保存します。

- ⑩ 上記で作成したフローを URL トリガーに設定します。ここで、「公開するプロトコル」が「3.2.3 ⑭」で設定した URL と同じになるように設定し、登録します。

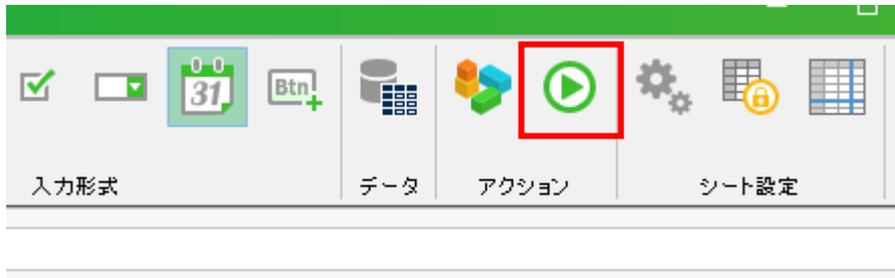


4. 操作方法

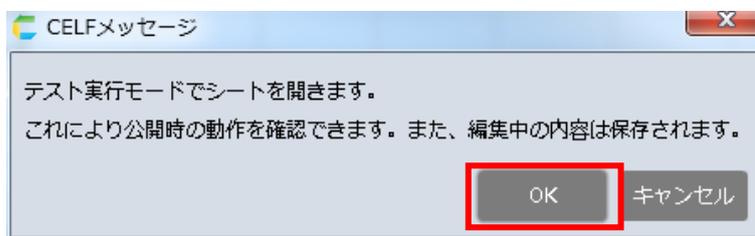
4.1. CELF からのテスト実行

4.1.1. テスト実行

- ① 作成したシートを開き、メニュータブの「アクション」の「テスト実行」を押下します。



- ② 以下のポップアップが表示されるので、「OK」を押下します。



- ③ テスト画面に遷移するので、入力データに値を入力後、ボタンを押下することでボタンに設定されているアクションセットが実行されます。

CELLF-ASTERIA連携サンプル				
入力データ				実行
実行時刻				
出力データ				

↓

CELLF-ASTERIA連携サンプル				
入力データ	2			実行
実行時刻	2017-03-22T12:25:02.452 JST			
出力データ	3	4	5	
	6	7	8	
	9	10	11	