

Magic xpi 4.14

新機能ご紹介

2023/11/16

マジックソフトウェア・ジャパン株式会社
コンサルティング部

陳 鍾三

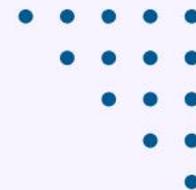




Agenda



- IMM(In Memory Middleware)
- 新Magicモニタ
- 新規/機能追加コンポーネント
- 今後の計画

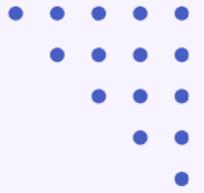




IMM (In Memory Middleware)



IMM(In Memory Middleware)

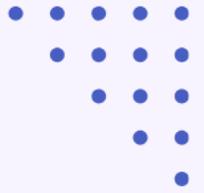


Magic xpi4.14ではメッセージング層を構成するミドルウェアが変更されました。

【ミドルウェアの変遷】

- ~Magic xpi3.x : Magic Request Broker : 独自製品
- ~Magic xpi 4.13 : In Memory DataGrid : GigaSpaces, サードパーティ製品
**GigaSpace Technology社との契約終了に伴い、2026年12月31日で
GigaSpacesはライセンス切れのため、動作しなくなります
2026年12月31日までにMagic xpi最新版へのUpgradeをお願いいたします**
- Magic xpi4.14~ : In Memory Middleware : 独自製品
 - パフォーマンスと信頼性の向上を目指して

IMM(In Memory Middleware)



【4.13と比較して】

- **軽い、早い**

⇒ たとえばHTTPトリガの最初のリクエストの実行時間が大幅に短縮

⇒ リソース使用量：少 下図はプロジェクト停止状態のWSL2上でのIMM使用リソース

タスク マネージャー

検索する名前、発行元、PID を入力してください

詳細

新しいタスクを実行する | タスクを終了する

名前	PID	状態	ユーザー名	セッシ...	CPU	メモリ (アクティブなプライベート...	コマンドライン
vmmemWSL	14976	実行中	B02C66E6-9089-48FC-A8EB-62C06...	0	00	2,303,588 K	
vmwp.exe	8712	実行中	B02C66E6-9089-48FC-A8EB-62C06...	0	00	2,888 K	"C:\WINDOWS\System32\vmwp.exe" B02C66E6-9089-48...

IMM(In Memory Middleware)



【IMMアーキテクチャ概要】

- メッセージング層として構成
 - 独自コンテナ(Docker/Kubernetes)として構成
 - **Docker/ Kubernetes : Linux上の最新コンテナ技術**
 - Linux上でIMMは稼働
- Magic xpi サーバ(プロジェクト)との通信
 - プロジェクトは今まで同様Windows上で稼働
 - IMMエージェント(Windows上のプロセス)を使用してIMMと通信
- IMMとMagic xpi サーバ
 - 1つのIMMに対し、複数のMagic xpiサーバが接続可能
 - 負荷分散、可用性

IMM(In Memory Middleware)

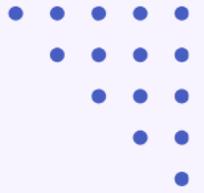


【 Docker/ Kubernetes 】

- ベースはコンテナ技術



IMM(In Memory Middleware)

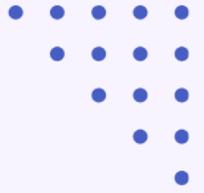


【 Docker/ Kubernetes 】

- コンテナはホストOS ごとに独立した環境を構築するため、別ホストで運用する場合には管理が煩雑



IMM(In Memory Middleware)



【 Docker/ Kubernetes 】

- Kubernetes : 複数ホストでコンテナを管理・運用

	Kubernetes	Docker
実行場所	クラスタ	単一ノード
利用目的	コンテナの管理	アプリケーションのコンテナ分離

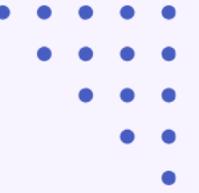
IMM(In Memory Middleware)



【 Docker/ Kubernetes 】

- Kubernetesのメリット
 - アプリケーションの効率的な新機能リリースが可能
アプリケーションの公開を自動化できる。万が一、アプリケーションに不具合があった場合には、修正後のバージョンへ入れ替えたり、ロールバックをしたりする作業も自動で行われる。
 - 障害に強い自己回復機能
Kubernetes には自己回復機能により、コンテナがダウンしたり、操作ミスで削除してしまったりした場合も、自動回復。
 - オンプレミス・クラウドの両環境で利用可能
オンプレミスでの利用の場合、Kubernetes 用のソフトウェア・ハードウェアを使用。
クラウドでの利用の場合は、パブリッククラウド上で提供される Kubernetes 用のサービスを利用。
 - 高速・軽量な起動で開発スピードを加速できる

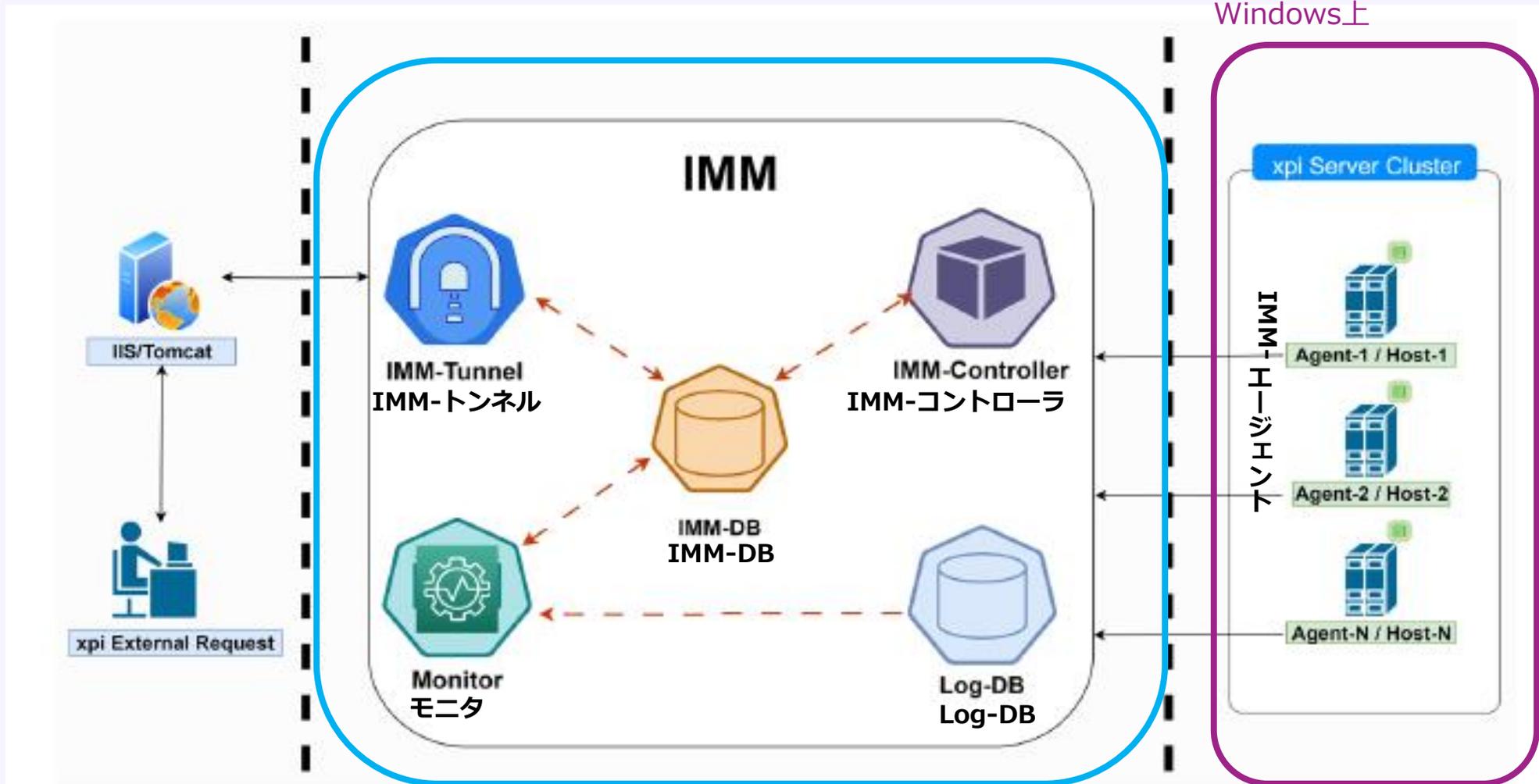
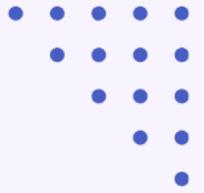
IMM(In Memory Middleware)



【 Docker/ Kubernetes 】

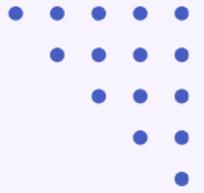
- Kubernetesのメリット
 - 柔軟にスケーリングできる
Kubernetes 搭載Podの「水平オートスケーリング機能」
 - サービスディスカバリーで動作を把握できる
サーバー上のアプリケーションのポート番号や IP アドレスを知らせてくれる機能
どのアプリケーションが動いているのかをすぐに把握
 - セキュリティを強化できる
パスワードや SSH キーなどの情報を保全する機能

IMM(In Memory Middleware)



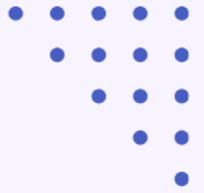
Linux上のコンテナ
複数サービスがIMM内のノードとしてデプロイ

IMM(In Memory Middleware)



- IMM-エージェント
 - xpiサーバ(プロジェクト)が実行されるサーバ上で必ず稼働
 - プロジェクトの起動/停止、正常性のチェック
 - 複数のエージェントもIMMで一括管理
- IMM-コントローラ
 - IMM エージェントと通信し、xpiサーバの開始/停止の要求を受け付ける
- IMM-DB
 - IMM が機能するために必要なすべての運用データが保存されるセントラルリポジトリすべての xpi 関連のランタイム データ、プロジェクト メタデータ、およびリアルタイム実行データが含まれます
 - インメモリデータベースのRedisを使用
- LogDB
 - xpiサーバによって生成されるすべてのアクティビティ ログの保管コンテナ
 - 高速なMongoDBを使用

IMM(In Memory Middleware)



- IMM-トンネル

- IMM アーキテクチャ内に存在する Webサービス
- すべての外部リクエスト (HTTPリクエスト、TCPリクエストなど) を処理し、リクエストをIMM コントローラにリダイレクト。

注: HTTP Web サーバー トリガーなどに必要なWeb サーバを代替するものではありません

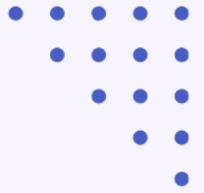
- モニタ

- IMM とその配下で実行されている xpi プロジェクトに関する情報と状態関連データを提供するインターフェイスです。
- 主な機能

ライセンスの使用状況、サーバーの負荷、フローの実行状況

注: 過去バージョンとは異なり、xpi モニタは IMM インフラストラクチャ内に常駐します(Windowsサービスとして提供されません)。

IMM(In Memory Middleware)



【開発環境構築】

- WSL2を使用
 - WSL : Windows Subsystem for Linux ⇒ Windows上でLinuxを動かす仕組み
 - PowerShell等のコマンドを実行する必要はありません
 - **Rancher Desktop(SUSE)**でらくらく環境構築
 - 自動的にDocker/Kubernetesをインストール/構成します
- 簡単デプロイ
 - Magic xpi4.14をインストール
 - インストールフォルダ内にデプロイ用のバッチファイルが用意 ⇒ 実行するだけ
 - 前述5つのサービスが自動的にデプロイ

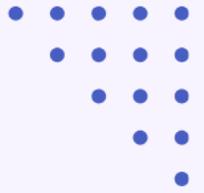
IMM(In Memory Middleware)



The screenshot shows the Rancher Desktop application window. The title bar reads "Rancher Desktop" and includes standard window controls. Below the title bar is a menu bar with "File", "Edit", "View", and "Help". The main interface features a sidebar on the left with the following menu items: "General" (selected), "Port Forwarding", "Images", "Troubleshooting", "Diagnostics" (with a red notification badge containing the number "1"), and "Extensions". The main content area is titled "Welcome to Rancher Desktop" and contains the following text and elements:

- A paragraph: "Rancher Desktop provides Kubernetes and image management through the use of a desktop application."
- A bulleted list:
 - Project Discussions: [#rancher-desktop](#) in [Rancher Users Slack](#)
 - Project Links: [Homepage Issues](#)
- A horizontal separator line.
- Version information: "Version: 1.10.0" on the left and a checked checkbox "Check for updates automatically" on the right.
- Another horizontal separator line.
- A checked checkbox "Allow collection of anonymous statistics to help us improve Rancher Desktop" followed by the text: "Send anonymized usage info, error reports, etc. to help improve Rancher Desktop. Your data will not be shared with anyone else, and no information about what specific resources or endpoints you are deploying is included."
- Another horizontal separator line.
- Network status: "Network status: online"

IMM(In Memory Middleware)

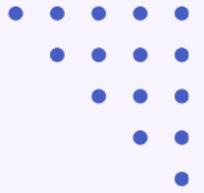


The screenshot shows the Rancher Desktop application interface with the Preferences dialog box open. The dialog box is titled "Rancher Desktop - Preferences" and has a "Kubernetes" tab selected. The settings are as follows:

Section	Setting	Value
Application	Enable Kubernetes	<input checked="" type="checkbox"/>
	Kubernetes version	v1.28.2 (latest)
Kubernetes	Kubernetes Port	6443
	Enable Traefik	<input type="checkbox"/>

At the bottom of the dialog box, there are "Cancel" and "Apply" buttons. A help icon (?) is also visible in the bottom right corner of the dialog box.

IMM(In Memory Middleware)



Rancher Desktop

File Edit View Help

RANCHER DESKTOP

General

Port Forwarding

Images

Troubleshooting

Diagnostics 1

Extensions

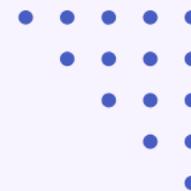
Port Forwarding

include Rancher Desktop services

Namespace	Name	Port	Local Port	
default	xpi-ingress-controller-ingress-nginx-controller	http		Forward
default	xpi-ingress-controller-ingress-nginx-controller	https		Forward
default	xpi-ingress-controller-ingress-nginx-controller	27017-tcp		Forward
default	xpi-ingress-controller-ingress-nginx-controller	6379-tcp		Forward
default	xpi-ingress-controller-ingress-nginx-controller-admission	https-webhook		Forward
magic-xpi-imm-ns	imm-controller-service	5116		Forward
magic-xpi-imm-ns	imm-db-service	6379		Forward
magic-xpi-imm-ns	imm-tunnel-service	5117		Forward
magic-xpi-imm-ns	logdb-service	27017		Forward
magic-xpi-imm-ns	xpi-monitor-service	8181		Forward

IMM5つのサービス

IMM(In Memory Middleware)



Magic xpi インストール

In Memory Middleware パラメータ

アプリケーションに共通のサービスと機能を提供するIMMインフラストラクチャの詳細を入力します。

IMMの詳細を指定する

IMM ホスト	chin830g8.jp
IMM ポート	6379

Log DBの詳細を指定する

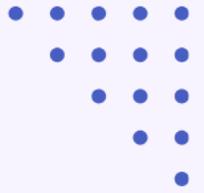
Log DB ホスト	chin830g8.jp
Log DB ポート設定	27017

InstallShield

<戻る 次へ> キャンセル

Magic xpi4.14インストール時に
接続するIMMを指定

IMM(In Memory Middleware)



デプロイ用バッチファイル

```
C:\Magicxpi4.14\InMemoryMiddleware\deploy\deploy-imm.bat - 秀丸
ファイル(F) 編集(E) 表示(V) 検索(S) ウィンドウ(W) マクロ(M) その他(O)
>>
1  :: Script provide easy install of IMM on Kubernetes.
2  :: Requires kubectl and helm to be locally installed, and the .kubeconfig file in place.
3  :: Please refer to the "Magicxpi.4.x-Advanced_Deployment_Guide.pdf" for more details.
4  @echo off
5  :: <consts
6  set C_note=[97mNOTE:[0m
7  set C_info=[92mINFO:[0m
8  set C_warn=[93mWARN:[0m
9  set C_err=[91mERROR:[0m
10 set C_done=echo [92m== Done == [0m
11 set_errBit=0
12 set GLOBAL_SETTINGS_FILE="..%config%globalsettings.properties"
13 set DBALERT_SETTINGS_FILE="..%Runtime%Config%db_alert_config.properties"
14 set LOGBACK_SETTINGS_FILE="..%config%logback.xml"
15 set MONITOR_CONFIG_FILE="..%config%config.json"
16 set KEY_FILE_PATH="..%config%imm.key"
17 set CERT_FILE_PATH="..%config%imm.crt"
18 set_PROTOCOL=http
19 :: >consts
20 echo.
21 :: Check kube config
22 kubectl >nul 2>&1 | | (
23 >     echo %C_err%: kubectl command not available. This is a prerequisite. Please make sure kubectl is installed.
24 >     goto :exLoop
25 )
26 helm >nul 2>&1 | | (
27 >     echo %C_err%: helm command not available. please make sure helm is installed.
28 >     goto :exLoop
29 )
30 IF NOT EXIST "%USERPROFILE%\%kubeconfig%" (
31 >     echo %C_err%: Kube Config not found, please make sure that you have kubeconfig file at "%USERPROFILE%\%kubeconfig%"
32 )&& goto :exLoop
33 )
34 set k8sPlatform=microk8s
35 kubectl cluster-info | findstr "CoreDNS" | findstr ".azmk8s.io:" && set k8sPlatform=AKS
36 rem kubectl cluster-info | findstr "CoreDNS" | findstr ".amazonaws.com" && set k8sPlatform=EKS# Not validated yet.
37 IF [%k8sPlatform%] = [AKS] (
38 kubectl delete validatingwebhookconfigurations xpi-ingress-controller-ingress-nginx-admission
39 )
```

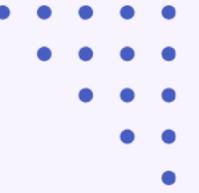
IMM(In Memory Middleware)



```
C:\Magicxpi4.14\InMemoryMiddleware\deploy\EndPoints.txt - 秀丸
ファイル(F) 編集(E) 表示(V) 検索(S) ウィンドウ(W) マクロ(M) その他(O)
===== Please make note of the deployed Endpoints =====
1 IMM-DB at: chin830g8.jp:6379
2 LOG-DB at: chin830g8.jp:27017
3 IMM Controller URL: chin830g8.jp/controller
4 IMM-Tunnel URL: http://chin830g8.jp/immtunnel
5 Magic xpi monitor URL: http://chin830g8.jp/magicmonitor
6 ===== Action needed =====
7 IMP Add the following to you System host file for name resolution.
8 for e.g. in windows "C:\WINDOWS\system32\drivers\etc\hosts"
9 -----
10 172.31.14.217 chin830g8.jp
11 -----
12
13
```

デプロイされたサービスとポート

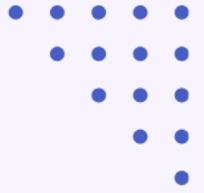
IMM(In Memory Middleware)



【実行環境構築】

- Linux(Ubuntu)を準備
 - MicroK8Sインストール
 - Kubernetes 接続情報を出力し、Windowsへ
- 簡単デプロイ
 - 接続情報をLinuxからコピー
 - Magic xpi4.14をインストール
 - インストールフォルダ内にデプロイ用のバッチファイルが用意 ⇒ 実行するだけ
WindowsからLinux上のKubernetesに接続し、デプロイ
 - 前述5つのサービスがLinux上にデプロイ

IMM(In Memory Middleware)



【システム要件】

- 開発環境
 - CPU : 2.66Ghz以上 4コア以上
 - メモリ : 16G以上
 - 空きディスク容量 : 25G以上
 - OS : Windows 10 pro, Windows 11 pro WSL2が有効である必要あり
(クラウド/仮想環境ではWSL2が無効の可能性あり)
Ubuntu 22.04.x LTS

注意) 上記仕様は、xpi 開発環境とIMMを同一環境で実行する場合を想定。

IMM(In Memory Middleware)



【システム要件】

- xpi 実行環境とIMMを同一環境で実行する場合
 - CPU : 2.66Ghz以上 8コア以上
 - メモリ : 32G以上
 - 空きディスク容量 : 25G以上
 - OS : Windows Server 2022 WSL2が有効である必要あり (クラウド/仮想環境ではWSL2が無効の可能性あり)
Ubuntu 22.04.x LTS

IMM(In Memory Middleware)



【システム要件】

- xpi実行環境とIMMを別環境で実行する場合

【Windowsサーバ】

- CPU : 2.66Ghz以上 4コア以上
- メモリ : 32G以上
- 空きディスク容量 : 25G以上
- OS : Windows Server 2016、2019、2022

【Linuxサーバ】

- CPU : 2.66Ghz以上 4コア以上
- メモリ : 16G以上
- 空きディスク容量 : 10G以上
- OS : Ubuntu 22.04.x LTS

注意 1) AWSでは、EKSでもIMMは動作するが、料金体系が高額なので、Linuxでの動作が現実的。

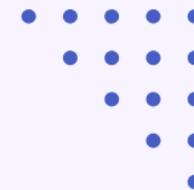
注意 2) 本社開発部では、引き続き、WSL2以外にもWindowsOS上(Hyper-Vなど) でIMMを実行できる環境を検討/検証中。
正式サポートが確認された場合は、追って通知。



新Magic毛二夕



新Magicモニタ



- IMM内で稼働
- 提供機能
 - Angularで作成された新モニタ
 - 過去バージョンと同等(ただしGigaSpaces情報はなし)
 - 刷新され、使いやすさが向上したユーザーインターフェイス : Angularベース
 - 以前のモニタとくらべ、高速

新Magicモニタ



【ログイン】

セキュリティ保護なし | chin830g8.jp/magicmonitor/#/auth/lo...

Magic xpi へようこそ

ユーザー*
administer

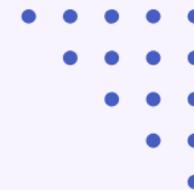
パスワード*
.....

ログイン

Or

Azure-SSO でログインする

新Magicモニタ



【プロジェクトの選択】

ブラウザアドレスバー: chin830g8.jp/magicmonitor/#/dashboard

ページタイトル: Magic xpi モニット

ダッシュボード

プロジェクトのロードと保留中のメッセージ

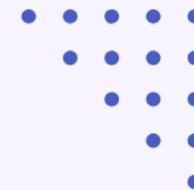
Magic XPI プロジェクトリスト

プロジェクトを検索

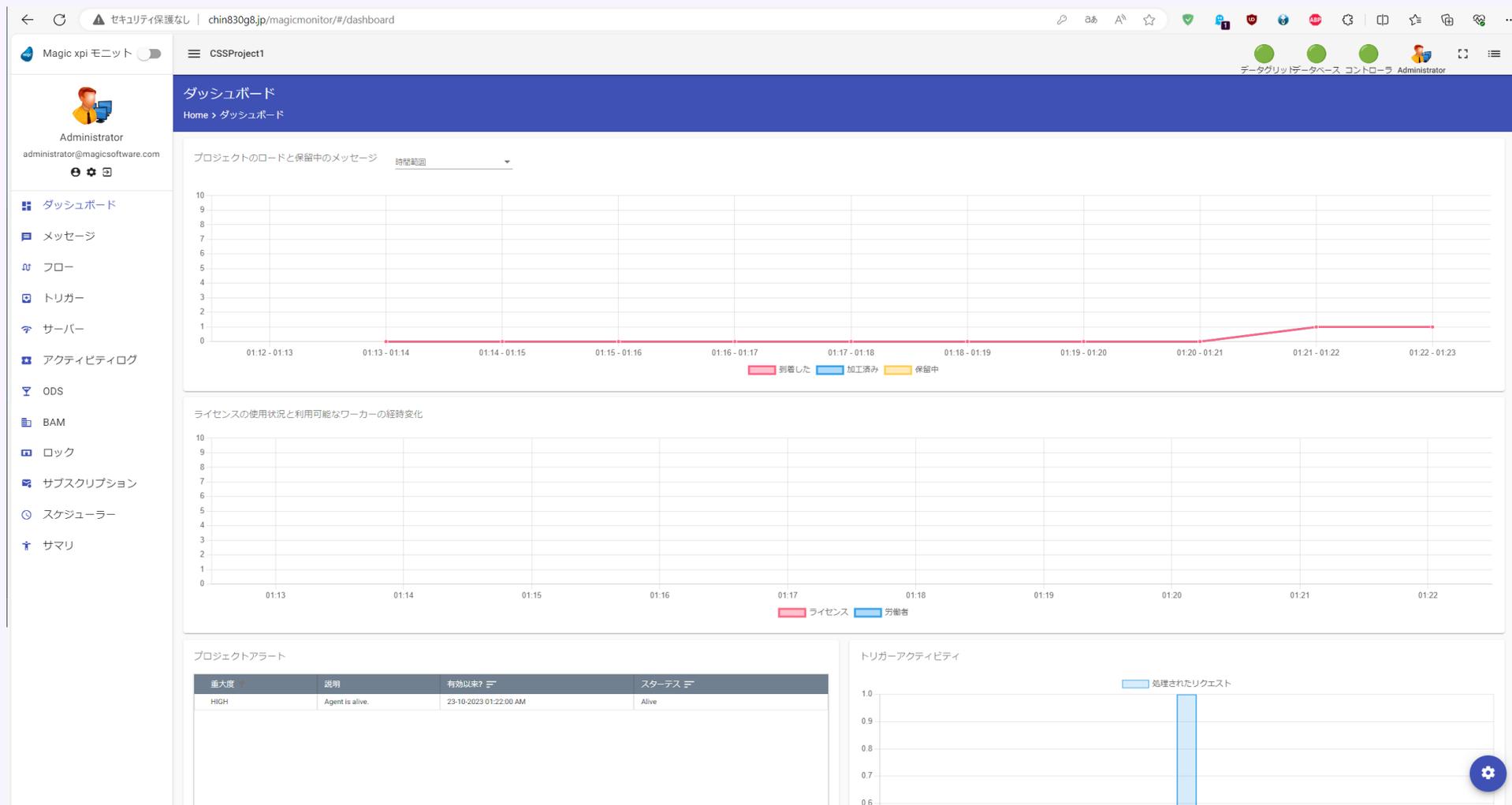
- AS400_Test
- AS400_TestProject
- bat_test
- BoardToSF
- BOMTEST
- CheckProject1
- ConvertFile
- CSSProject1
- D365_JS
- D365_Test
- D365FOSanity_IP
- DataLink_DB2400_MySQLWin
- DBCSDDataConverter
- DELETE
- DemoSAPSFDC413
- DSTest
- Dynamics365FODemo
- EDIFACTProcess
- ExcelTest
- EXCELXPI

始める

新Magicモニタ



【ダッシュボード】



新Magicモニタ

【アクティビティログ】

← 刷新 🔒 セキュリティ保護なし | chin830g8.jp/magicmonitor/#/activitylog

Magic xpi モニット CSSProject1

Administrator administrator@magicsoftware.com

データグリッドデータベース コントローラ Administrator

アクティビティログ
Home > アクティビティログ

サーバーID ▼ ビジネスプロセス ▼ フロー ▼ ステップ ▼ アドオンフィルター ▼

生産活動

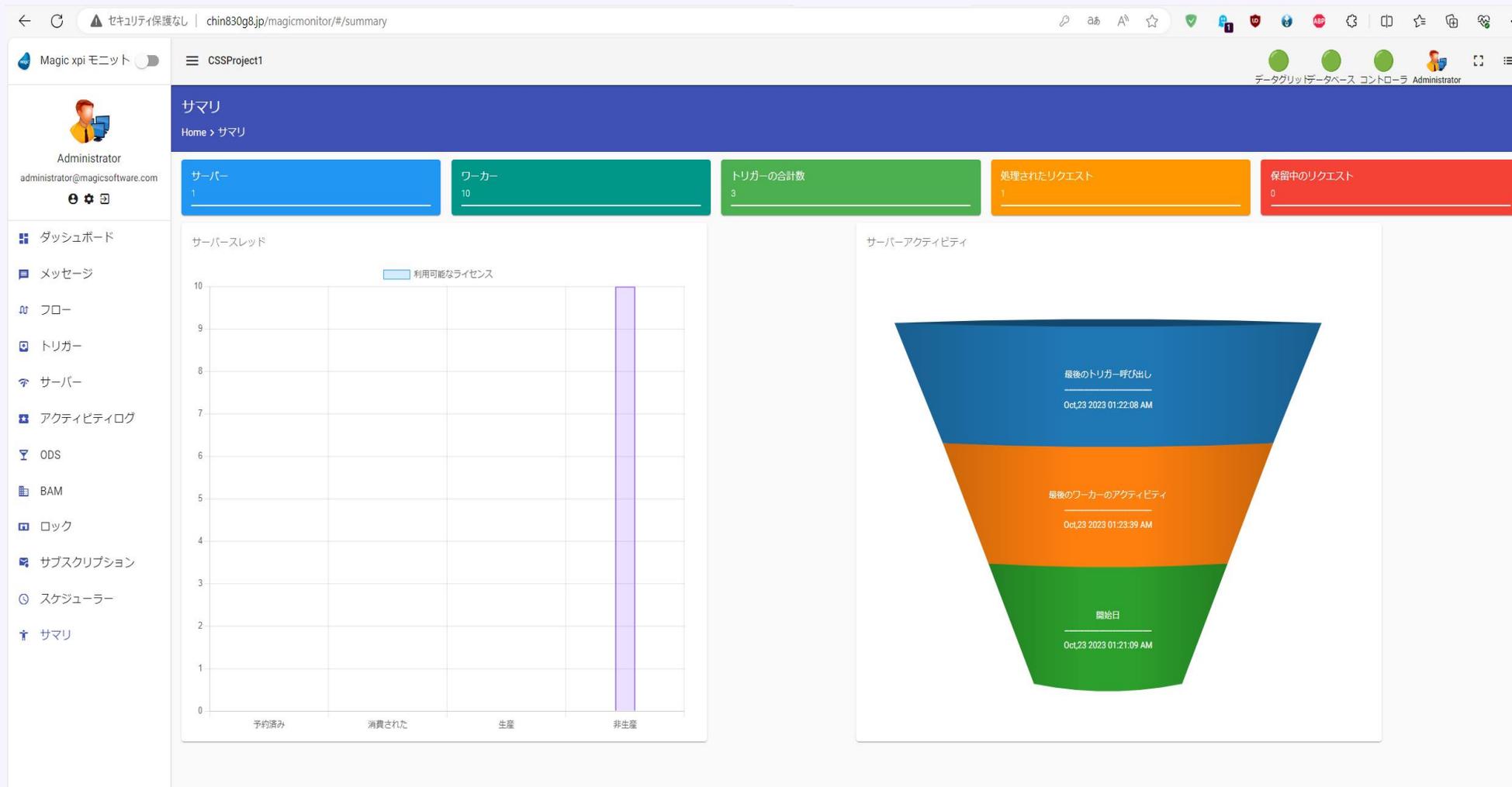
日付時刻	メッセージの種類	メッセージ文字列	サーバー名	ビジネスプロセス	フロー名	ステップ名	ルートFSID	フローリクエストID	FSID	Blob
23-10-2023 03:22:08 PM	Flow completed	Execution Time: 00:00:00:113	Server_1	Business Process-1	Flow-1		1	1	1	
23-10-2023 03:22:08 PM	Flow component completed	Execution Time: 00:00:00:002	Server_1	Business Process-1	Flow-1	NOP	1	1	1	
23-10-2023 03:22:08 PM	Flow component started	Linear	Server_1	Business Process-1	Flow-1	NOP	1	1	1	
23-10-2023 03:22:08 PM	Flow component completed	Execution Time: 00:00:00:063	Server_1	Business Process-1	Flow-1	Flow Data	1	1	1	
23-10-2023 03:22:08 PM	Flow component started	Linear	Server_1	Business Process-1	Flow-1	Flow Data	1	1	1	
23-10-2023 03:22:08 PM	Flow started		Server_1	Business Process-1	Flow-1		1	1	1	
23-10-2023 03:22:08 PM	Trigger Started		Server_1	Business Process-1	Flow-1		0	1	0	
23-10-2023 03:21:13 PM	Server started	- Instance number 1, The server was started with a non-production license, and will shut down in 24 hours.	Server_1				0	0	0	
22-10-2023 10:11:16 AM	Server stopped		Server_1				0	0	0	
22-10-2023 10:08:34 AM	Flow completed	Execution Time: 00:00:00:112	Server_1	Business Process-1	Flow-1		1	1	1	
22-10-2023 10:08:34 AM	Flow component completed	Execution Time: 00:00:00:003	Server_1	Business Process-1	Flow-1	NOP	1	1	1	
22-10-2023 10:08:34 AM	Flow component started	Linear	Server_1	Business Process-1	Flow-1	NOP	1	1	1	
22-10-2023 10:08:34 AM	Flow component completed	Execution Time: 00:00:00:062	Server_1	Business Process-1	Flow-1	Flow Data	1	1	1	
22-10-2023 10:08:34 AM	Flow component started	Linear	Server_1	Business Process-1	Flow-1	Flow Data	1	1	1	
22-10-2023 10:08:34 AM	Flow started		Server_1	Business Process-1	Flow-1		1	1	1	
22-10-2023 10:08:34 AM	Trigger Started		Server_1	Business Process-1	Flow-1		0	1	0	
22-10-2023 10:07:46 AM	Server started	- Instance number 1, The server was started with a non-production license, and will shut down in 24 hours.	Server_1				0	0	0	
21-10-2023 02:21:19 AM	Flow completed	Execution Time: 00:00:00:110	Server_3	Business Process-1	Flow-1		1	1	1	
21-10-2023 02:21:19 AM	Flow component completed	Execution Time: 00:00:00:001	Server_3	Business Process-1	Flow-1	NOP	1	1	1	
21-10-2023 02:21:19 AM	Flow component started	Linear	Server_3	Business Process-1	Flow-1	NOP	1	1	1	
21-10-2023 02:21:19 AM	Flow component completed	Execution Time: 00:00:00:062	Server_3	Business Process-1	Flow-1	Flow Data	1	1	1	
21-10-2023 02:21:19 AM	Flow component started	Linear	Server_3	Business Process-1	Flow-1	Flow Data	1	1	1	

ページ毎のアイテム数: 50 1 - 27 of 27

新Magicモニタ



【サマリ】

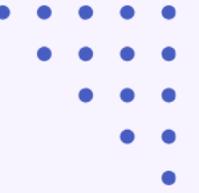




新規/機能追加 コンポーネント



新規/機能追加コンポーネント



- 新Excelコンポーネント
 - Apache POI採用
 - バックグラウンドで動作
 - 実行時Excelは不要（開発時は必要）
 - ユーザーインターフェイス刷新
- REST Client(機能追加)
 - 認証サポート(OAuth、Basic認証)
 - 様々なWebAPIを呼出
- REST APIコンポーネント
 - RESTトリガ機能
 - OAuth、Basic認証サポート
 - OpenAPI定義とりこみ&フロー生成機能

新Excelコンポーネント

Excel設定

接続
リソース名: Excel_1

オペレーション
Excelファイル: 'C:\Users\chin\Documents\Magic\Project\...
範囲: ""ワークシート-オブジェクト (見慣れ) ""\$B\$23:\$A\$A
オペレーション: Read

列

#	列名	データタイプ
1	Item	String
2	ListPrice	Numeric
3	unitPrice	Numeric
4	quantity	Numeric
5	offerPrice	Numeric

ウィザードでExcel展開

項目取り込み

オプション
モード: Overwrite
行から開始:

戻り値
データBlob: Variable C.UserBlob
データフォーマット: XML
更新された範囲: ...

プロパティ
Excel ステップ特性

General	
ステップ名	Excel
説明	
ステップID	5
コンポーネントID	283157797
コンポーネントタイプ	Excel
条件	
処理モード	リニア
完了を待つ	No
インタフェース	XML
アドバンスド	
セーブポイント	No
コールドジョックフロー	No
ロギング	
ステップ名	コンポーネント名またはユーティリティ名。そのステップが何をするかを説明...

新Excelコンポーネント

Project414Test - Magic xpi Studio (管理者)

ツールボックス

- マッパー スキーマ
 - ポインター
 - Call Flow
 - DataBase
 - Flat File
 - JSON
 - ODS
 - Template
 - UDS
 - Variable
 - XML
- データコネクター
 - ポインター
 - JDBC
- 全般

このグループには使用可能なコントロールがありません。項目をこのテキスト上にドラッグして、ツールボックスに追加してください。

データマッパー (NewExcel) x WriteExcel (NewExcel) NewExcel (NewExcel) Flow-1 (Business Process-1)

送り元

名前	データ型
row	()
Item	()
ListPrice	()
unitPrice	()
quantity	()
offerPrice	()

送り先

名前	データ型
Name1	()
Name2	()
Name3	()
Name4	()
Name5	()

ソリューション エクスプローラー

- PSSトピック
- UDS
- エラー
- グローバル変数
- コンテキスト変数
- データコネクター
- デフォルトデータフォーマット
- Business Process-1
 - BP変数
- Flow-1
 - フロー変数
 - エラーポリシー
- NewExcel
 - BP変数
 - NewExcel
 - フロー変数
 - エラーポリシー
 - WriteExcel
 - BP変数

プロパティ

FlatFile_2 フラットファイルスキーマ

プロパティ	値
スキーマ情報	
ID	2
一般	
送り先タイプ	File
データ送り先エンコーディング	ANSI
ファイルパス	'c:\temp\excel.csv'
データ追加	No
タイプ	区切文字
列区切文字	,
レコード区切文字	
区切文字をデータに含める	No
ヘッダー行を作成	No
詳細	(コレクション)
名	FlatFile_2

送り先タイプ

送り元：送り元データがどこにあるか。
送り先：送り先データが保存される場所

チェッカー結果:

0 エラー 0 警告 0 メッセージ

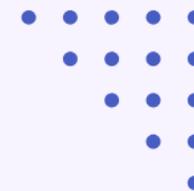
#	重大度	BP/名称	フロー	ステップ/トリガー	コントロール名	説明
---	-----	-------	-----	-----------	---------	----

追加情報:

チェッカー結果: 出力

準備完了

REST クライアントコンポーネント(認証サポート)



The screenshot shows the '設定' (Settings) window for a REST client. The left sidebar shows a tree view with 'リソース' (Resources) expanded, and 'REST' selected. Under 'REST', 'RESTBasic' is highlighted. The main area displays a table of settings for the selected component.

#	名	タイプ	フォーマット	値
1	ローカルエージェント	論理	1	False
2	API定義メソッド	文字	30	Custom
3	サーバURL	文字	2500	http://localhost:8080
4	セキュリティ	文字	30	HTTP-Basic
5	ユーザ	文字	100	
6	パスワード	文字	30	
7	HttpMultipartMode	文字	100	
8	クライアント証明書要求	文字	1	

The 'セキュリティ' (Security) dropdown menu is open, showing the following options: None, HTTP-Basic, HTTP-Digest, OAuth2, and No. The 'HTTP-Basic' option is selected and highlighted. A purple box highlights the dropdown menu, and a purple label '認証方法の選択' (Authentication method selection) points to it.

リソース >> REST Client >> RESTBasic (REST Clientを使用すると、OAuth認証、multipart/form-data、send/receive、添付ファイルなどのRestful APIの機能を利用できます。)

Username : admin | Password : nhk

REST Serverによって公開される構成対象のセキュリティタイプを選択してください

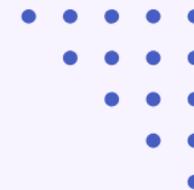
アクセストークン パス OpenAPIのロード

追加 削除 ロード 適用 OK キャンセル

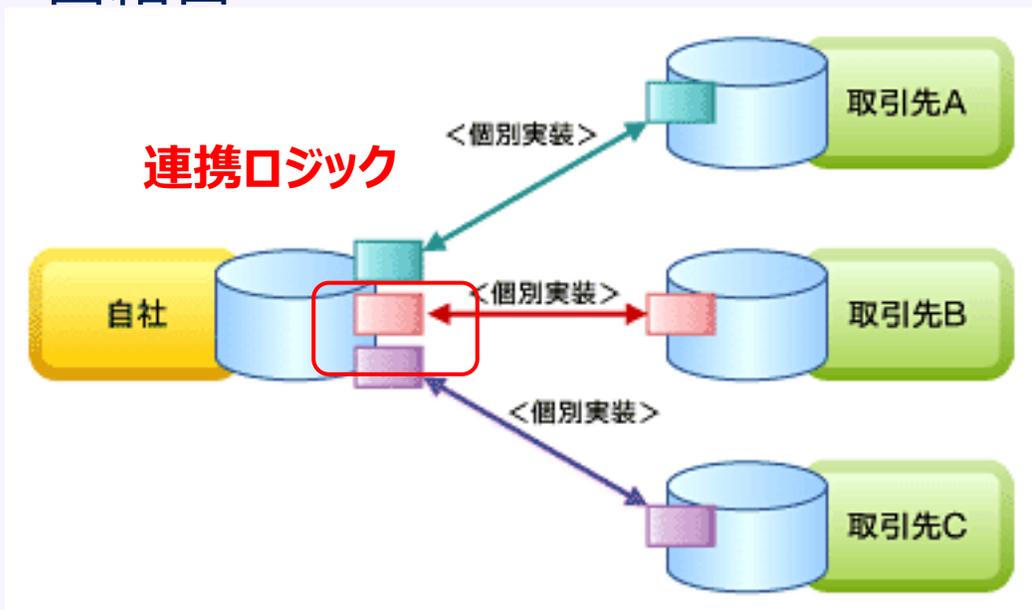
REST APIコンポーネント (RESTトリガ) で何が実現できるのか



密結合と疎結合



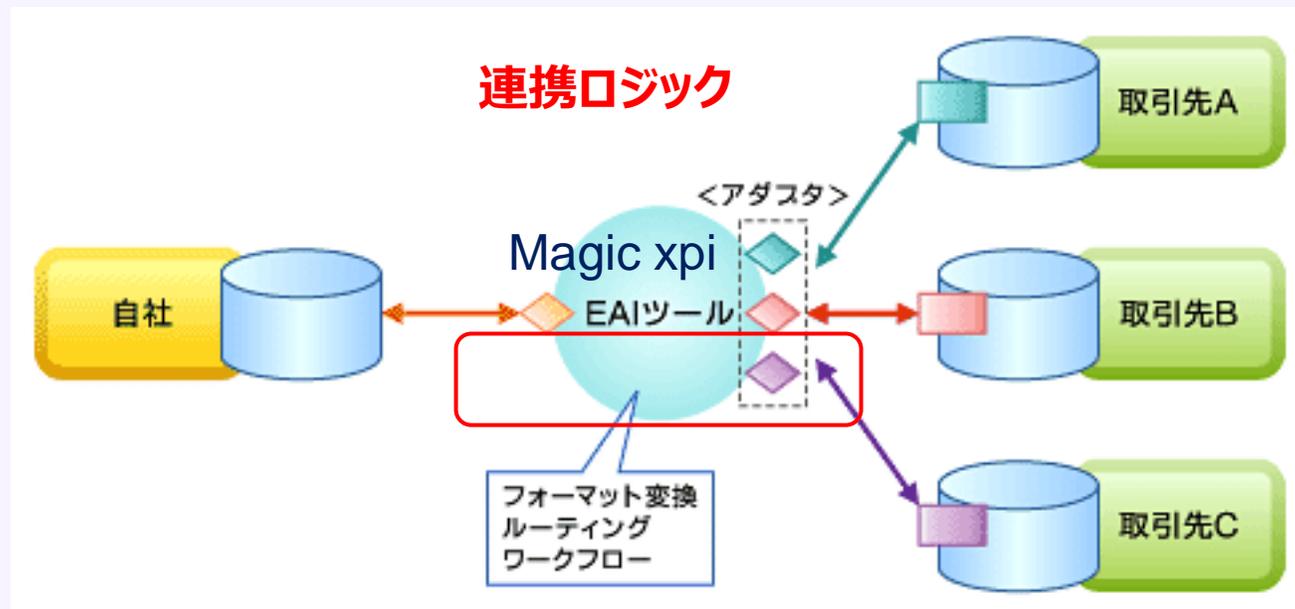
■ 密結合



■ 密結合のデメリット

- 連携先毎に個別に実装する必要がある
- 個々にメンテナンス⇒大変

■ 疎結合



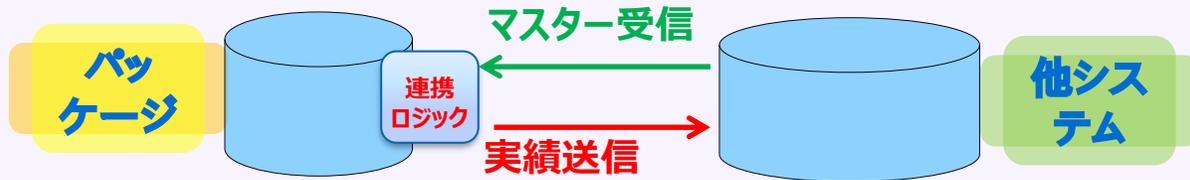
■ 疎結合のメリット

- 連携ロジックを切り離すことができる⇒身軽
- EAIツール側に連携ロジックを集中

実際のシステムでは・・・

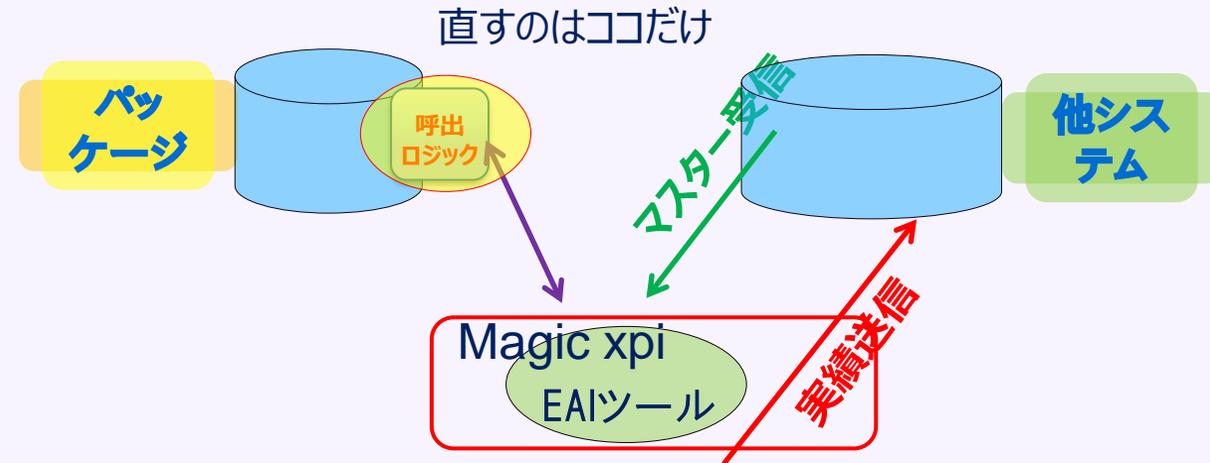
パッケージ

■ 密結合



- 導入先毎に他システムが異なる
 - 環境(DBMS)が導入先毎に異なる
 - マスター受信も毎回作り直し
 - 実績送信も毎回作り直し
 - 連携先毎に個別に実装する必要がある

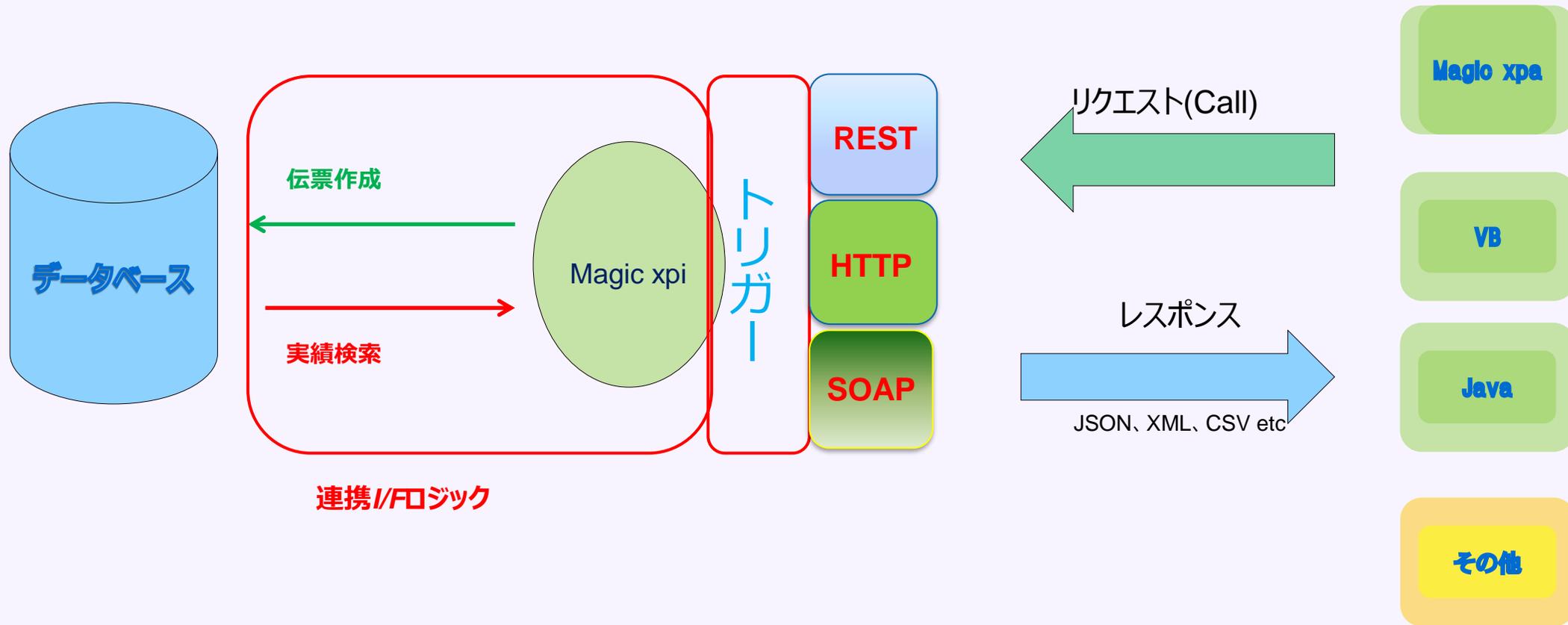
■ 疎結合



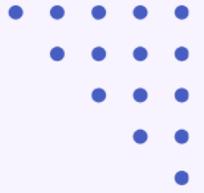
連携ロジック

- 連携ロジックを集中
 - ロジックは全てEAI(Magic xpi)側に集中
 - 環境(DBMS)の違いはEAI(Magic xpi)が吸収
 - パッケージ側は呼出のみ⇒身軽

構成イメージ



Magic xpiで連携用APIを作成する

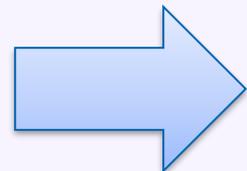


Magic xpiで作成した機能(フロー)を、連携のAPIとすることができる

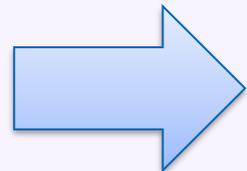
WebServiceトリガを使うことで、WebService(SOAP) APIとなる

HTTPトリガーを使うことでHTTP APIとなる

RESTトリガーを使うことで、最新技術トレンドのREST API(JSON)となる



連携I/Fを元のソフトウェアから切り離せる



疎結合、部品化の実現、メンテナンス性の向上

REST APIコンポーネント(RESTトリガ)

The screenshot displays the IBM Business Process Manager (BPM) interface for configuring a REST API trigger. The main window shows a flowchart with a 'Response' activity. A dialog box titled 'RESTful APIトリガ構成' (RESTful API Trigger Configuration) is open, showing the following details:

- 接続 (Connection):** サービス名: REST
- オペレーション (Operation):** パス: /getCustomer (Get), オペレーション: Get
- パラメータ詳細 (Parameter Details):**

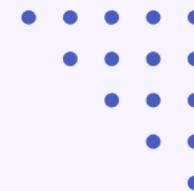
#	パラメータタイプ	パラメータ名	パラメータ値
1	リクエスト	body	
- 出力 (Output):** レスポンスボディ: CUserBlob, レスポンスヘッダ: (empty), レスポンスステータス: (empty)

The 'プロパティ (Properties)' pane on the right shows the configuration for the 'RESTful API トリガ-特性' (RESTful API Trigger Characteristics):

Property	Value
トリガ-名 (Trigger Name)	RESTful API
トリガ-ID (Trigger ID)	5
コンポーネントID (Component ID)	1753821964
コンポーネントタイプ (Component Type)	RESTful API
ロギング (Logging)	
ロギングスコープ (Logging Scope)	No
メッセージ (Message)	
Blob	
ファイル拡張子 (File Extension)	
設定 (Settings)	
サービス名 (Service Name)	REST
サービス説明 (Service Description)	

The bottom status bar shows '準備完了' (Ready) and a 'チェック-結果' (Check Results) section with 0 errors, 0 warnings, and 0 messages.

REST APIコンポーネント(RESTトリガ)



設定

リソース
サービス
REST
RESTBasic
RESTOpenAPI
カレンダー
プロジェクト環境
一般環境

#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	http://localhost:8080/xpiREST
2	API定義メソッド	文字	30	Custom
3	セキュリティ	文字	30	None

サービス >> RESTful API >> REST (REST APIトリガはOpenAPIドキュメントをサポートします)

パス OpenAPIのロード フロー生成

追加 削除 ロード 適用 OK キャンセル

REST APIコンポーネント(RESTトリガ)

RESTFull APIトリガ構成

接続
サービス名: RESTBasic

オペレーション
パス: /postCustomer (Post)
オペレーション: Post

#	パラメータタイプ	パラメータ名	パラメータ値
1	クエリ	id	F.id
2	クエリ	name	F.name
3	クエリ	country	F.country
4	ヘッダ	Content-Type	F.ContentType
5	ヘッダ	xpi-tech	F.xpitech
6	リクエスト	body	C.HTTP_Body
7	リクエスト	Userid	C.UserString

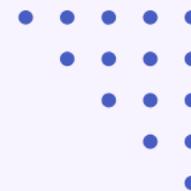
出力
レスポンスボディ: C.UserBlob
レスポンスヘッダ: ...
レスポンスステータス: ...

プロパティ
GETトリガ-特性

General	
トリガ-名	GET
説明	
トリガ-ID	6
コンポーネントID	1753821964
コンポーネントタイプ	RESTful API
ロギング	
ロギングスコープ	No
メッセージ	
Blob	
ファイル拡張子	
設定	
サービス名	RESTBasic
サービス説明	

トリガ-名
コンポーネント名。別の名前を入力して、トリガ-の動作を説明することが...

REST APIコンポーネント(RESTトリガ : 認証サポート)



#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	http://localhost:8080/xpiREST
2	API定義メソッド	文字	30	Custom
3	セキュリティ	文字	30	Basic

認証
Basic
OAuth2

サービス >> RESTful API >> RESTBasic (REST APIトリガーはOpenAPIドキュメントをサポートします)

本RESTサービスのセキュリティタイプを設定します。重要な注意点 : この変更を有効にするには、「パス」ウィンドウを開いて「OK」をクリックしてください。

追加 削除 ロード 適用 OK キャンセル

REST APIコンポーネント(RESTトリガ : OpenAPI取込)



設定

リソース
サービス
REST
RESTBasic
RESTOpenAPI
カレンダー
プロジェクト環境
一般環境

#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	https://api.example.com
2	API定義メソッド	文字	30	Open Api
3	OpenAPIドキュメント	文字	260	C:\Magicxpi414\Runtime\projects\NHK_POcv1\Files\OpenAPIDoc\FlowerShop.json
4	OpenAPIドキュメントのバージョン	文字	30	OpenApi3_0
5	セキュリティ	文字	30	None

サービス >> RESTful API >> RESTOpenAPI (REST APIトリガーはOpenAPIドキュメントをサポートします)

OpenAPIドキュメントのバージョン

パス OpenAPIのロード フロー生成

追加 削除 ロード 適用 OK キャンセル

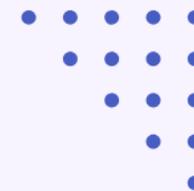
REST APIコンポーネント(RESTトリガ : OpenAPIスキーマ)



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <service xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="RESTOpenAPI" project="TEST_POCV1" security="0">
3 <securityDefinitions/>
4 <endpoint path="/flowers">
5 <get>
6 <produces>application/xml;application/json</produces>
7 <parameters>
8 <body name="body" required="true"/>
9 </parameters>
10 </get>
11 <post>
12 <produces>application/xml;application/json</produces>
13 <parameters>
14 <body name="body" required="true"/>
15 </parameters>
16 </post>
17 </endpoint>
18 <endpoint path="/flowers/{flowerId}">
19 <get>
20 <produces>application/xml;application/json</produces>
21 <parameters>
22 <path name="flowerId" required="true"/>
23 <body name="body" required="true"/>
24 </parameters>
25 </get>
26 <put>
27 <produces>application/xml;application/json</produces>
28 <parameters>
29 <path name="flowerId" required="true"/>
30 <body name="body" required="true"/>
31 </parameters>
32 </put>
33 <delete>
34 <produces>application/xml;application/json</produces>
35 <parameters>
36 <path name="flowerId" required="true"/>
37 <body name="body" required="true"/>
38 </parameters>
39 </delete>
40 </endpoint>
41 </service>[EOF]
```

```
C:\Users\chin\Documents\Magic\Projects\414Projects\NHK_POCV1\RESTful API\RESTOpenAPI\OpenApiSchema\Flower.json - 秀丸
ファイル(F) 編集(E) 表示(V) 検索(S) ウィンドウ(W) マクロ(M) その他(O)
1 {
2   "type": "object",
3   "properties": {
4     "id": {
5       "type": "string"
6     },
7     "name": {
8       "type": "string"
9     },
10    "price": {
11      "type": "number"
12    },
13    "description": {
14      "type": "string"
15    },
16    "image": {
17      "type": "string"
18    }
19  }
20 }[EOF]
```

REST APIコンポーネント(RESTトリガ : フロー自動生成)



RESTFull APIトリガ構成

接続
サービス名: RESTOpenAPI

オペレーション
パス: flowers/{flowerId} (Get)
オペレーション: Get

パラメータ詳細

#	パラメータタイプ	パラメータ名	パラメータ値
1	パス	flowerId	F.ID11_パス_flowerId
2	リクエスト	body	F.ID11_リクエスト_ボディ

出力
レスポンスボディ: F.ID11_出カ_レスポンスボディ
レスポンスヘッダ: F.ID11_出カ_レスポンスヘッダ
レスポンスステータス: F.ID11_出カ_レスポンスステータス

トリガ (1)

ソリューション エクスプローラー

- BasicAuth
 - フロー変数
 - エラーポリシー
- RESTClient
 - BP変数
 - NoAuth
 - BasicAuthGet
 - フロー変数
 - エラーポリシー
 - BasicAuthPost
 - Digest
- AG3_RESTOpenAPI_flowers
 - BP変数
 - AG8_flowers (Get)
 - AG9_flowers (Post)
 - AG10_flowers_{flowerId} (Delete)
 - AG11_flowers_{flowerId} (Get)
 - AG12_flowers_{flowerId} (Put)

プロパティ

AG11_flowers_{flowerId} (Get) トリガ特性

General	
トリガ名	AG11_flowers_{flowerId} (Get)
説明	
トリガID	11
コンポーネントID	1753821964
コンポーネントタイプ	RESTful API

ロギング

ロギングスコープ	No
メッセージ	
Blob	
ファイル拡張子	

設定

サービス名	RESTOpenAPI
サービス説明	

トリガ名
コンポーネント名。別の名前を入力して、トリガの動作を説明することが...

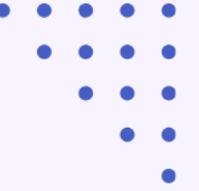
成フロー



今後の計画



今後の計画

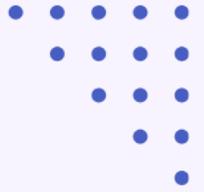


【今後の計画】

- Linux Magic xpiサーバ
- データマージコンポーネント
 - 異なるデータソースをJOINする機能
- データベース更新の高速化
 - Multiレコード INSERTサポート
- Studioのブラウザ化

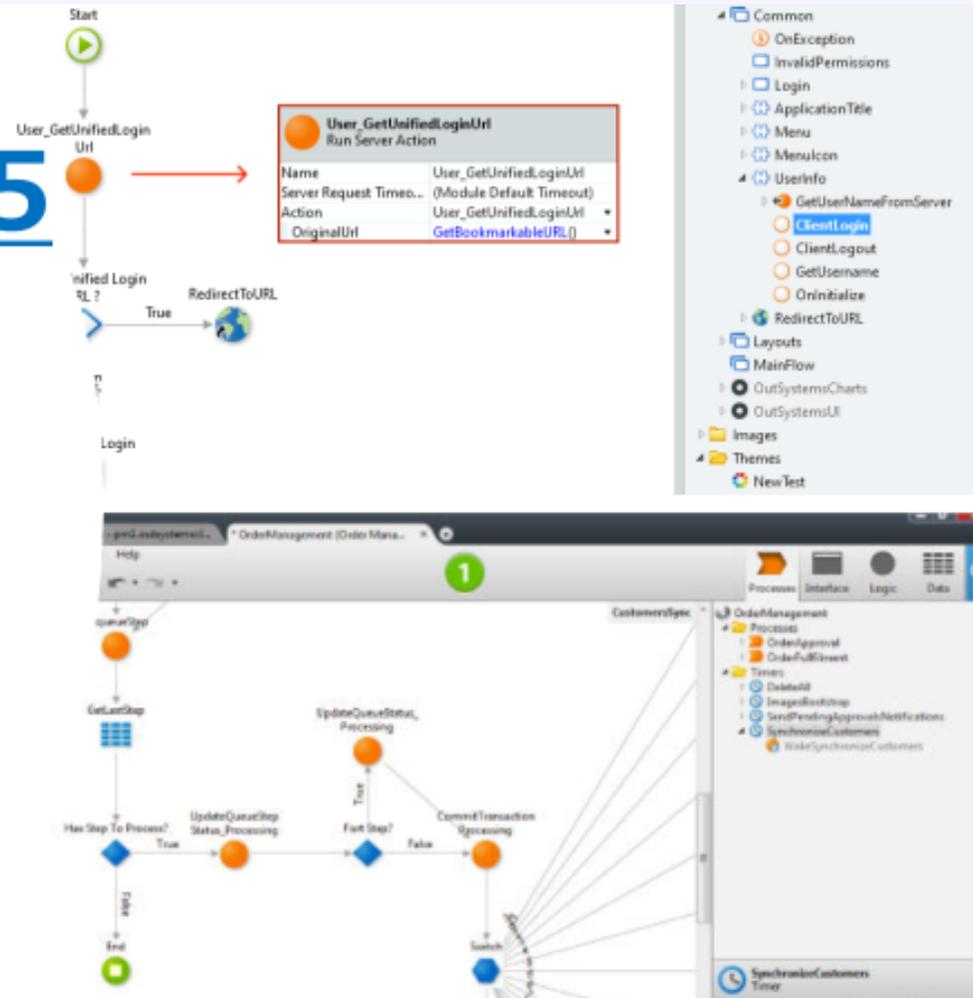
※ 提供時期は未定です
計画が変更になり、提供されなくなる場合もあります

今後の計画



2025 - XPI 5

XPI new Browser based studio



※ 提供時期は未定です
計画が変更になり、提供されなくなる場合もあります

ご清聴ありがとうございました。

