Azureをベースとした開発環境、プロジェクト体制について

マジックソフトウェア・ジャパン株式会社

2021・07

Azureを使ったクラウド開発環境のご提案

テレワークを支援するクラウド開発環境のご提案

WebClientで使用する環境

クラウド環境

Microsoft Azure

※ Azure DevOps Services
(xpa ソースコード管理)

Azure SQL Database
(データベース)

GitHub
(Angular ソースコード管理)

Magic xpa 4.x

SQL Server
Management Studio
[SSMS]

Visual Studio Code [VSC]

ローカル環境

※サポート対象外となります。

クラウドベース開発環境の特徴

- サーバーレス環境を実現できる

  サーバー機の導入・管理コストがなくなります。（サービス利用料は逆に発生します）

- 共有資源はインターネット上のサービスで管理できる

  インターネット回線があれば簡単に接続することができます。
  強固なクラウドサービス基盤を利用できる。

- 開発ロケーションの制約がなくなる

  組織を超えた開発環境を提供できます。
  柔軟な開発体制を作ることができます。→　必要な人材の確保

Azure SQLのメリット

- アップグレード、修正プログラムの適用、バックアップ、監視などのほとんどのデータベース管理機能をユーザーの介入なしで実行する、フル マネージドの PaaS (サービスとしてのプラットフォーム) データベース エンジンです。

- 常に最新の安定したバージョンの SQL Server データベース エンジンおよびパッチが適用された OS 上で実行され、99.99% の可用性を備えています。

- Azure SQL Database に組み込まれた PaaS 機能により、お客様は、ビジネスにとって不可欠なドメイン固有のデータベース管理や最適化アクティビティに集中することができます。

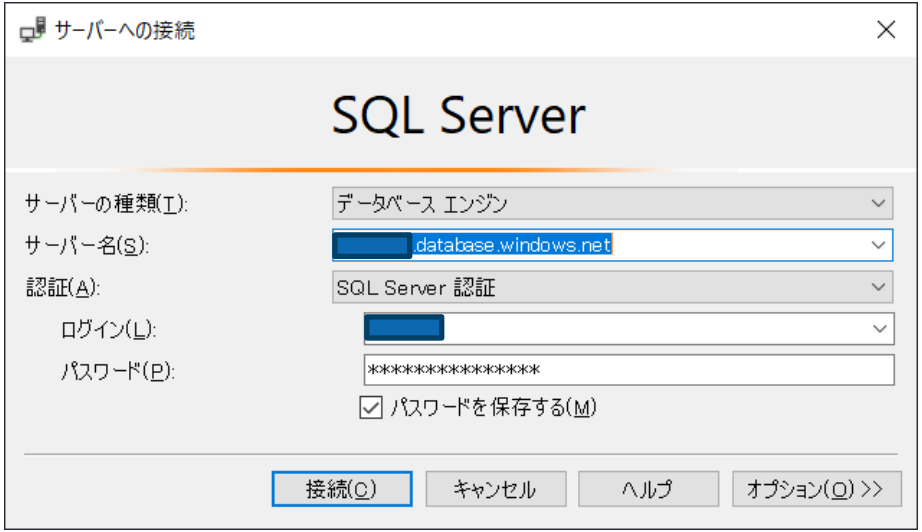
(引用)Azure SQL Database とは
https://docs.microsoft.com/ja-jp/azure/azure-sql/database/sql-database-paas-overview

Azure 環境の利用イメージ(Azure SQL)

xpa4.6以降で正式対応

- サーバーの場所をクラウドのアドレスにするだけ

管理ツールは同じなので、すぐに使いこなせます。



Azure 環境の利用イメージ (Azure DevOps Services)

５アカウントまでは無料

- ソース管理の場所をAzureにするだけ

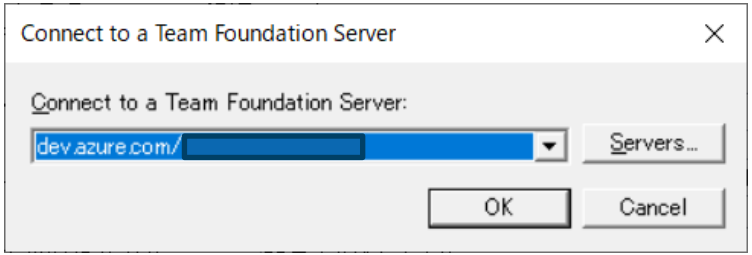
  接続が確立されるといつも通りのソース管理が行えます。

  （チェックイン・チェックアウトの繰り返し）

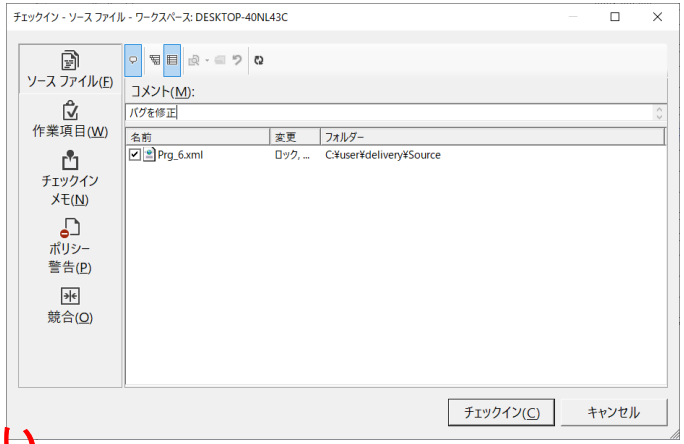
- 既知の問題点

  Studio起動後2回目以降のチェックインでエラーが表示される場合がある。→ Studio 再起動で対応

[サーバーの場所から開く]を選択したときに表示されるサーバー選択ウインドー



チェックイン ウインドー



※サポート対象外となります。自己責任でご利用ください。

クラウドを利用した利用環境の提案ができます

- 実行環境でもAzure SQLの利用が可能になります。
  CS環境でもクライアント実行版から直接接続が可能です。

- 導入が手軽です。
  導入コストがなく、すぐ利用できます。

- 再起動やパッチ適用の管理コストが低減します。

- 冗長化された強固なSQL Server環境を提案できます。

- xpa4への移行時のクラウド提案のネタにして下さい。

WebClient
プロジェクトの注意点

WebClient開発の注意点

- Angular の開発・運用経験者が必要です
  開発環境の構築ができる。(Angular開発環境の作成、DevOps、GitHubの設定)
  Angular のコーディングやデバッグに対処できる。
  Angular のエラーに対処できる。(xpaのプログラム・タスク削除)
  GitHubの運用やエラーに対処できる。(getや pull、 ステージング)
  実行環境のデプロイ・保守ができる。

- Webシステム開発として
  画面デザインができる。(CSS)
  SQL文での処理ができる。→ 処理の高速化

WebClient開発が必要となるスキル(1)

Web Client スキルチェックシート
１．Magic
1.1. データベースのトランザクションの概念を理解している
1.2. Magic xpaを使ったＣ／Ｓシステムの開発経験がある
1.3. Magic xpaを使ったＲＩＡシステムの開発経験がある
1.4. 遅延トランザクションの仕組みを理解している
1.5. Magicのコンテキスト管理の仕組みを理解している
1.6. MagicのWebアプリケーションの仕組みを理解している（MRB、インターネットリクエスタ）
1.7. Magicのプロジェクトをサーバに配置（デプロイ）した経験がある
２．WEBシステム全般
2.1. Webサーバー（IISなど）のインストール／設定ができる
2.2. HTTPリクエスト／レスポンスやWebセッション管理などの基本を理解している
2.3. Webプログラミング言語（Java、PHP、Pythonなど）を経験している
2.4. Webアプリケーションのデバッグを経験している
2.5. IDEをインストールしてWebシステム構築に利用したことがある（Visual Studio Code、Eclipseなど）
2.6. Gitを使ったチーム開発を経験している

WebClient開発が必要となるスキル(2)

Web Client スキルチェックシート	
３． Angular	
3.1.	Angularプロジェクトの基本構成、または主要な設定ファイルを理解している
3.2.	定義済みのAngularプロジェクトからアプリケーションの動きをトレースすることができる
3.3.	実行やビルドなどの基本的なAngularコマンドを理解している
3.4.	Angularを使ったSPAの構築経験がある
3.5.	Angularのルーティングや遅延ロードモジュールなどの基本機能を理解している
3.6.	Angularプロジェクトのデプロイの経験がある
４． HTML5	
4.1.	HTMLのコードを理解できる
4.2.	HTMLを少し触ったことがある
4.3.	HTMLの主要なタグを理解している（<html>、<table>など）
4.4.	margin・paddingの意味と動作を理解し、レイアウトを定義できる
4.5.	HTMLの基礎的な知識があり、ある程度の動くものを作れる
4.6.	自社または個人でデザインテンプレート的なHTMLファイルを保持している

WebClient開発が必要となるスキル(3)

Web Client スキルチェックシート	
5．CSS3	
5.1.	CSSのコードを理解できる
5.2.	CSSを少し触ったことがある
5.3.	CSSを使った開発経験がある
5.4.	CSSを１からコーディングできる
5.5.	CSSの基礎的な知識があり、ある程度の動くものを作れる
5.6.	自社または個人で利用可能なCSSファイルを保持している
6．JavaScript	
6.1.	JavaScriptのコードを理解できる
6.2.	JavaScriptを少し触ったことがある
6.3.	JavaScriptを使った開発経験がある
6.4.	JavaScriptを１からコーディングできる
6.5.	JavaScriptの基礎的な知識があり、ある程度の動くものを作れる
6.6.	自社または個人で利用可能なJavaScriptファイルを保持している

WebClient開発が必要となるスキル(4)

Web Client スキルチェックシート
７．TypeScript
7.1. TypeScriptのコードを理解できる
7.2. TypeScriptを少し触ったことがある
7.3. TypeScriptを使った開発経験がある
7.4. TypeScriptを１からコーディングできる
7.5. TypeScriptの基礎的な知識があり、ある程度の動くものを作れる
８．JS系フレームワーク
8.1. 「Vue.js」「React」などのJS系フレームワークの利用経験がある
8.2. npmを活用した開発経験がある（Node.jsパッケージなど）
8.3. オブジェクト指向ベースのコンポーネントの構築経験がある
９．デザインフレームワーク
9.1. BootstrapやBulmaなどのデザインフレームワークの利用経験がある
9.2. 開発要件に合ったデザインフレームワークの選定ができる
9.3. 既存のデザインフレームワークをカスタマイズした経験がある
１０．UI/UX設計
10.1.使い易いWebデザイン設計をすることができる
10.2.既存プロジェクトのUIデザインを提案・改善することができる
10.3.レスポンシブ対応のアプリケーション開発経験がある

最後に

- Azure DevOps Servicesの利用はサポート対象外となります。
  自己責任でご利用ください。

- Azure やGitHub自体の質問はお受けできません。

- インタネット経由でのAzure SQLのご利用は、レスポンスに低下に
  注意して下さい。