

Лекція 11. Цифровий двійник (Digital Twin)

Тема 1. Загальні відомості про моделі двійників в Azure Digital Twins.

Тема 2. Мова визначення цифрових двійника (DTDL) для моделей.

Тема 3. Приклад створення цифрового двійника будівлі.

Визначення цифрового двійника в Майкрософт - цифрова репліка речей, місць, бізнес-процесів і людей в реальному світі.

Цифровий двійник призначений для розуміння, управління, моделювання, аналізу та поліпшення реальних бізнес-операцій. Azure Digital Twins - це платформа Інтернету речей, що дозволяє створювати рішення Інтернету речей нового покоління, які моделюють реальний світ.

Azure Digital Twins допомагає виконувати такі завдання:

- ✓ Створення комплексних цифрових моделей для цілих середовищ.
- ✓ Відстеження даних за минулі періоди і прогнозування майбутнього
- ✓ будь-якої підключеної середовища.
- ✓ Усунення розрізненості в підключених середовищах.

Ключовою характеристикою Azure Digital двійників є можливість визначення власного словника і створення графа двійника.

Модель схожа на клас в об'єктно-орієнтованій мові програмування, визначаючи фігуру даних для однієї конкретної концепції в реальному робочому середовищі. Моделі мають імена (наприклад, кімнати або датчик температури) і містять елементи.

Моделі цифрових двійників Azure представлені на мові визначення цифрових двійника на основі JSON-LD (DTDL).

DTDL створений на основі JSON-LD і не залежить від мови програмування. DTDL не є ексклюзивним для Azure Digital двійників, але також використовується для представлення даних пристрою в інших службах IoT. У Azure Digital двійників використовується DTDL версії 2 (використання версії 1 з Azure Digital двійників тепер застаріло). Не всі служби, які використовують DTDL, реалізують одні й ті ж функції DTDL. Наприклад, IoT самоналагоджувальний не використовує функції DTDL, призначені для графів, в той час як Azure Digital двійників в даний час не реалізує команди DTDL

Елементи моделі У визначенні моделі елемент коду верхнього рівня є інтерфейсом. Він інкапсулює всю модель, а інша частина моделі визначається через інтерфейс. Інтерфейс моделі DTDL може містити нуль, один або декілька з наступних полів: - Властивості - це поля даних, які представляють стан сутності (наприклад, властивості в багатьох мовах об'єктно-орієнтованого програмування). Властивості мають резервне сховище і можуть бути лічені в будь-який час.

Телеметрії - поля телеметрії представляють вимірювання або події, і часто використовуються для опису зчитування датчика пристрою. На відміну від властивостей, дані телеметрії не зберігається в цифровому двійнику; Це ряд подій даних, прив'язаних до часу, які повинні оброблятися в міру їх виникнення. - Компонент Components - дозволяє при необхідності побудувати інтерфейс моделі як збірку інших інтерфейсів. Прикладом компонента є інтерфейс фронткамера (і інша інтерфейсна Камера), який використовується при визначенні моделі для телефону.

Особливості реалізації DTDL в Azure Digital двійників

Щоб модель DTDL була сумісна з Azure Digital двійників, вона повинна відповідати цим вимогам. - Всі елементи DTDL верхнього рівня в моделі повинні мати тип Interface. Це обумовлено тим, що інтерфейси API моделі Digital двійників можуть отримати об'єкти JSON, що представляють інтерфейс або масив інтерфейсів. В результаті на верхньому рівні не допускаються інші типи елементів DTDL. - DTDL для Azure Digital двійників не повинен визначати які б то не було команди. - Azure Digital двійників допускає тільки один рівень вкладеності компонентів. Це означає, що інтерфейс, використовуваний в якості компонента, не може мати самих компонентів. - Інтерфейси не можуть бути визначені вбудованими в інших інтерфейсах DTDL; вони повинні бути визначені як окремі сутності верхнього рівня з власними ідентифікаторами. Потім, коли інший інтерфейс хоче включити цей інтерфейс як компонент або через успадкування, він може посилатися на його ідентифікатор. Azure Digital двійників також не стежить за writable атрибутами у властивостях і зв'язках. Хоча це значення можна задати у відповідності зі специфікаціями DTDL, воно не використовується в Azure Digital двійників. Замість цього вони завжди вважаються доступними для запису зовнішніми клієнтами, які мають загальні дозволи на запис в службу Digital двійників Azure.

Словник рішення Azure Digital двійників визначається за допомогою моделей, що описують типи сутностей, що існують у вашому середовищі. Іноді, якщо рішення прив'язане до певної галузі, воно може бути простим і ефективним для початку роботи з набором моделей, які вже існують, замість створення власного набору моделей з нуля. Ці набори моделей, що існували раніше називаються онтологією. Як правило, онтологія - це набір моделей для даного домену, таких як структура будівлі, система IoT, смарт-місто, сітка енергії, веб-вміст і т. д. Онтології часто використовуються в якості схем для діаграм знань, так як вони можуть включати : - Упорядкування програмних компонентів, документація, бібліотеки запитів і т. д. - Скорочені інвестиції в концептуальне моделювання та розробку систем - Спрощення взаємодії з даними на семантичному рівні - Рекомендоване повторне використання, а не починати з нуля або "переносити коліщатко"

Використання онтологій для Azure Digital двійників

Онтологія надає відмінну відправну точку для цифрових рішень двійника. Вони охоплюють набір моделей для конкретних доменів і зв'язку між сутностями для проектування, створення і аналізу цифрового двійника графа. Онтології дозволяють розробникам рішень почати цифрове рішення двійників з перевіреної відправної точки і зосередитися на вирішенні бізнесзавдань. Онтології створені компанією Майкрософт, також спроектовані так, щоб бути легко розширюваними, щоб їх можна було налаштувати для вирішення. Крім того, використання цих онтологій в рішеннях дозволяє налаштувати їх для більш тісної інтеграції між різними партнерами та постачальниками, так як онтологія може надати загальний словник для рішень. Оскільки моделі в Azure Digital двійників представлені в форматі Digital двійників Definition Language (DTDL), онтології для використання з Azure Digital двійників також пишуться в DTDL.

Використання стратегій онтології на шляху розробки моделі Незалежно від стратегії, обраної для інтеграції онтології в Azure Digital двійників, ви можете виконати інструкції по створенню та передачі онтології в якості моделей DTDL. 1. Почніть з вивчення і розуміння DTDL моделювання в Azure Digital двійників. 2. Продовжуйте роботу до обраної стратегії інтеграції онтології з наведених вище: Впровадження, Перетворення, Створення моделей на основі онтології. 3. Перевірте моделі, щоб переконатися, що вони працюють з DTDL документами. 4. Надішліть готові моделі в Azure Digital двійників за допомогою API або прикладу, подібного відправці Azure Digital двійників Model.

Існує три можливі стратегії інтеграції стандартних галузевих онтологій з DTDL ви можете вибрати найбільш підходящий варіант в залежності від ваших потреб: Впровадження. Ви можете запустити рішення з допомогою DTDL онтології з відкритим кодом, створеного на основі широко прийнятих галузевих стандартів. Можна, можливо або використовувати ці набори моделей, або розширити їх з допомогою власних доповнень для налаштованого рішення. Перетворення. Якщо у вас вже є моделі, представлені в іншому стандартному форматі, їх можна перетворити в DTDL, щоб використовувати їх з Azure Digital двійників. Створення. Ви завжди можете розробляти власні призначені для користувача моделі DTDL з нуля, використовуючи будь-які відповідні галузеві стандарти в якості ідей.

Розглянемо спрощений сценарій — створення цифрових двійників будівлі, поверхів і кімнат на основі готових моделей.

Семантичні моделі будівлі

Щоб створювати цифрові двійники, спочатку потрібні описові моделі (model definitions), які визначають:

- ✓ поняття (Building, Floor, Room),
- ✓ їх властивості (назва, тип, номер поверху тощо),
- ✓ їхні зв'язки (будівля — включає поверхи, поверх — містить кімнати).

Об’єкт	Приклад властивостей	Приклад зв’язку
Building	buildingId, name	містить Floors
Floor	floorId, level	розташований у Building
Room	roomId, name	розташована на Floor

Створення цифрових двійників

Після визначення моделей систему можна наповнювати екземплярами — цифровими двійниками реальних об’єктів.

У нашому прикладі створюються:
 один двійник будівлі,
 два двійники поверхів,
 кілька двійників кімнат.

Наприклад:

Будівля Building_01

Поверхи Floor_01 і Floor_02

Кімнати Room_101, Room_102, Room_201, Room_202

Кожен цифровий двійник має унікальний ідентифікатор і набір властивостей.

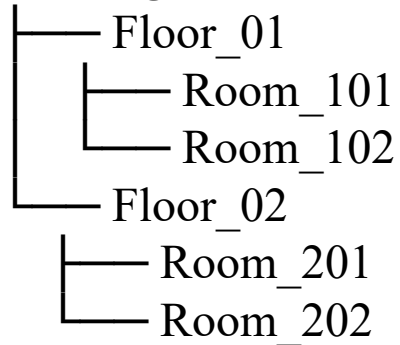
Цифрові двійники — це не просто окремі об'єкти. Важливою частиною архітектури є **взаємозв'язки**, що формують **граф двійників** (twin graph).

Граф показує:

- яке приміщення знаходиться на якому поверсі,
- які поверхи належать певній будівлі.

Структура графа - Цей граф — віртуальна модель фізичної будівлі.

Building



Переваги:

- **Гнучкість**: можна моделювати будь-яку будівлю — від офісу до фабрики.
- **Масштабування**: додаються нові рівні — будівлі, поверхи, кімнати, обладнання, сенсори.
- **Аналітика і моніторинг**: інтеграція IoT-даних, BIM-моделей, систем диспетчеризації.
- **Життєвий цикл будівлі**: проектування → будівництво → експлуатація → ремонт.

Azure Digital Twins Explorer

Home >

soloveidt
Azure Digital Twins



List the endpoints configured for this Azure Digital Twins instance.

How do I troubleshoot issues with this Azure Digital Twins instance?

Show me the properties of this Azure Digital Twins instance.

Search



Open Azure Digital Twins Explorer (preview)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

> Settings

> Connect outputs

> Monitoring

> Automation

> Help

Essentials

Resource group [\(move\)](#) : [demo](#)

Location : West Europe

Subscription [\(move\)](#) : [Azure subscription 1](#)

Subscription ID : fb666b7d-7778-420f-a041-cac5d45406b2

Tags [\(edit\)](#) : [Add tags](#)

Provisioning state : Active

Host name : soloveidt.api.weu.digitaltwins.azure.net

Metrics

Get Started

Learn

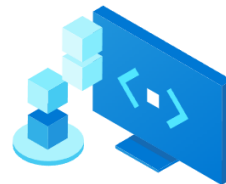
Access

Build

Integrate

Build your digital twins graph

Learn about Azure Digital Twins by using a sample app, and then start customizing your twins graph.



1. Get familiar with a sample

The Azure Digital Twins Explorer Quickstart will help you leverage existing models to visualize a sample modeled environment and explore the twins graph through the property and query tooling.

[Azure Digital Twins Explorer Quickstart](#)

2. Customize models or explore ontologies

Once you've gotten familiar with the twins graph, investigate custom models using Digital Twins Definition Language (DTDL) or start building off of



Add or remove favorites by pressing **Ctrl+Shift+F**

Моделі — це “словник” або понятійна база системи, яка описує все, що може бути цифровим двійником: будівлі, поверхи, кімнати, обладнання, сенсори тощо.

Модель визначає:

Тип об’єкта (наприклад, Building / Floor / Room)

Властивості (properties) — наприклад, ім’я, номер, площа, поверховість

Зв’язки (relationships) — “містить приміщення”, “розташований на поверсі”

Компоненти (components) — складні елементи, що можуть повторюватися

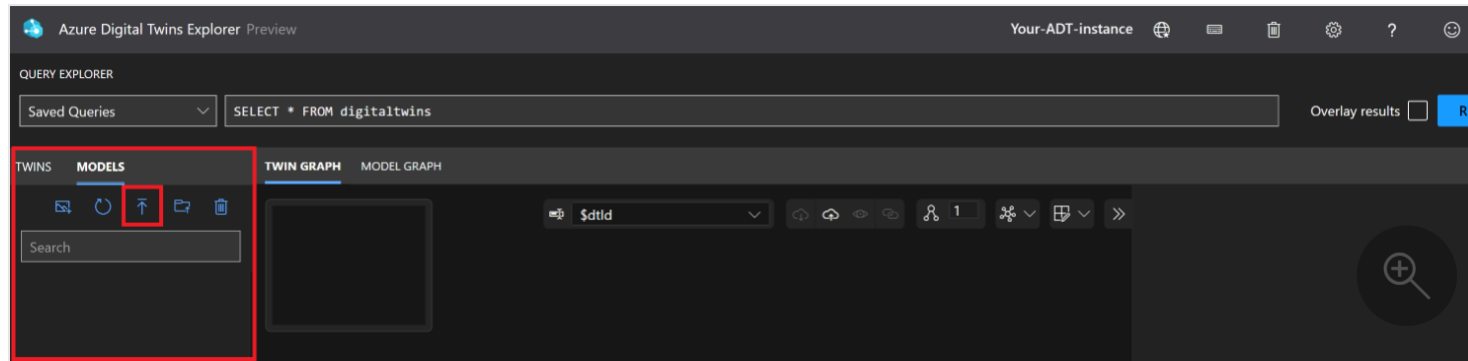
Моделі задають структуру, на основі якої потім створюються конкретні цифрові двійники.

Моделі Azure Digital Twins пишуться мовою **DTDL (Digital Twin Definition Language)**.

Особливості DTDL:

- ґрунтується на JSON-LD
- використовується для структурування IoT-об’єктів
- підтримує властивості, зв’язки, компоненти, телеметрію

```
{ "@id": "dtmi:example:Building;1",  
  "@type": "Interface", "displayName": "Building",  
  "@context": "dtmi:dtdl:context;2",  
  "contents": [ { "@type": "Property",  
    "name": "Date", "schema": "integer" },  
    { "@type": "Relationship", "name":  
    "contains", "displayName": "contains",  
    "target": "dtmi:example:Floor;1" } ] }
```



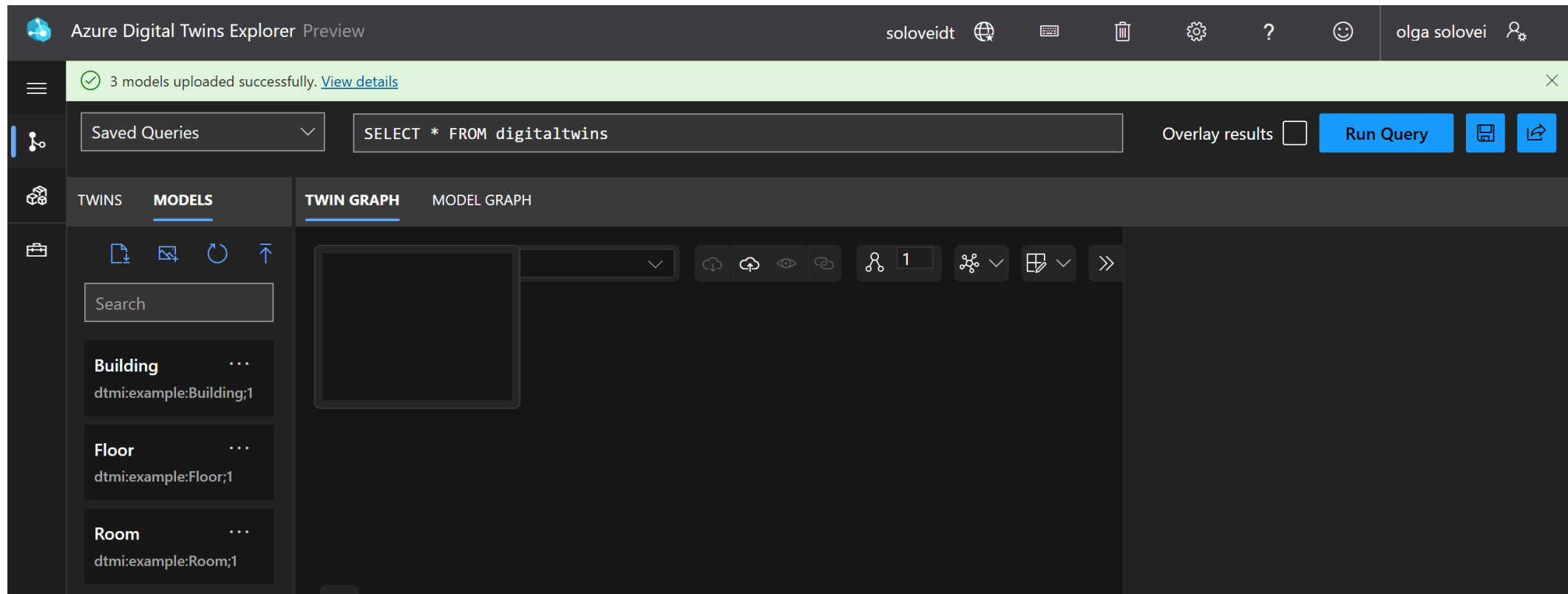
Azure Digital Twins Explorer (Preview) — вебінструмент для роботи з Digital Twins.

У верхній частині видно повідомлення “3 models uploaded successfully” — це означає, що DTDL-моделі (Digital Twins Definition Language) були успішно завантажені. У лівій панелі MODELS відображаються моделі: Building, Floor, Room - Це опис структури цифрового двійника.

Моделі визначають:

які властивості мають сутності (наприклад, Temperature, Humidity);

які зв'язки можливі (Building → Floor → Room).



Вкладка TWIN GRAPH показує екземпляри цифрових двійників, створені на основі моделей.

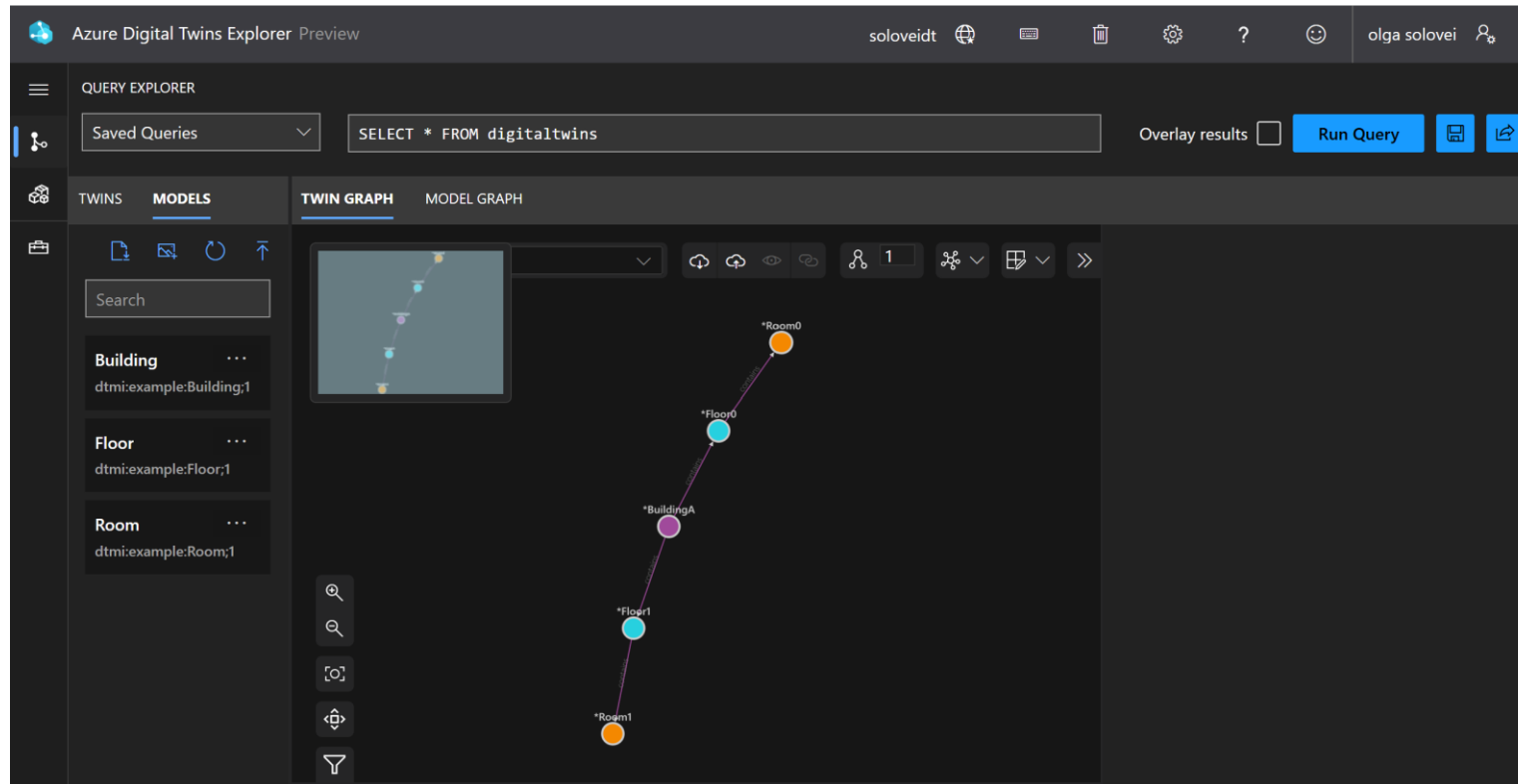
Видно ланцюжок: BuildingA->Floor0, Floor1->Room0, Room1

Кольори вузлів:

- Building — будівля
- Floor — поверх
- Room — кімната

Зв'язки між ними:

Лінії між вузлами — це relationships (наприклад, contains, hasFloor, hasRoom) показують ієрархію фізичного об'єкта у цифровому вигляді.



Перегляд властивостей конкретного Digital Twin

\$dtId: Room1 — унікальний ідентифікатор двійника. Властивості:

Humidity: 60

Temperature: 80

\$etag — версія об'єкта (використовується для конкурентних оновлень)

\$metadata — службова інформація

Що це означає: Це живі дані цифрового двійника, які: можуть оновлюватися з IoT-пристроїв; використовуються в аналітиці, правилах і логіці.

