

Efficient generation of customized, complex and consistent synthetic data

It is often not possible or permitted to use production data (anonymized or not) for testing purposes, and this can lead to unpleasant side effects.

Synthetic data is the alternative. In the past, however, creating synthetic data has been too costly, too time consuming, or the process simply impractical.

With iSynth, you can now create synthetic data efficiently and cost-effectively.

Key Features

Support for all test levels: iSynth can supply synthetic data for all test levels. From unit tests to fully integrated system tests on large system land-scapes. iSynth's synthetic data can be used as primary data or to complement existing data and is suitable for generating both small and large test data sets.

Flexible and extensible: iSynth is easy to adapt and extend according to specific requirements in terms of projects and target environments. iSynth users can implement specific functions themselves as extensions in Python.

Teamwork: iSynth supports collaboration across teams during the data definition, generation and usage phases.

Customized test data: iSynth customizes test data directly for specific test cases, making time-consuming searches for suitable test data a thing of the past. The data variance required for the test cases can be managed using rules or simple Excel sheets.

Manage data consumption: iSynth's integrated Data Checkout Tool can be used to mark individual data records as depleted and thus manage data consumption.

Metadata: iSynth allows you to define any metadata you wish to complement the generated data (e.g. you can add a test case ID).

Reusability: One of the basic concepts of iSynth is the definition of reusable components at the level of the data model, the templates for data constellations, as well as the deployment scripts.

Challenges

Regulatory requirements: The requirements for handling personal, health and banking data are becoming even more restrictive. In the age of big data analytics and machine learning, solutions based on anonymizing data will most likely fail to meet data and confidentiality protection requirements.

Agile projects and DevOps testing: In addition to appropriate development methods and tools, frameworks, service simulators, and test utilities, having the right amount of consistent, deterministic, high-quality test data is essential to performing appropriate testing.

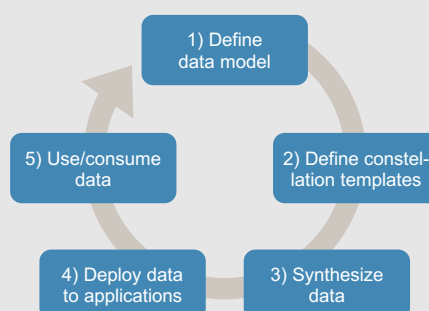
External sourcing: Software components are often developed by external partners. Whether these partners operate onshore, nearshore or offshore is irrelevant. The development partner needs data that accurately reflects the customer's sensitive production data and that it can use without security concerns.

Complexity of data and applications: IT systems are used to process increasingly complex and interconnected data. At the same time, IT landscapes are transformed from large, monolithic applications to fine-grained components (known as "micro services"). As a result, data is far more distributed. Therefore, test data must be provided consistently across all affected components for testing purposes.

Efficiency in testing: Whether in manual or automated testing, efficiency is highly dependent on test data. The search for suitable test data, the handling of varying data after data refreshes and the irreversible consumption of test data lead to inefficiencies in the test process.

Solution overview

iSynth supports the entire lifecycle of test data. Its object-oriented, model-based approach makes it possible to generate the required synthetic data stepwise in a structured, reproducible and yet flexible process.



- 1) The synthetic data model is defined from a business perspective using an object-oriented, Python-based syntax
- 2) Various data constellation templates are shaped using the object types from the data model
- 3) Data constellations are synthesized in the needed quantity and variance
- 4) Synthesized data is transformed into application-specific formats and deployed using existing application interfaces
- 5) Synthetic data is used and consumed during development and testing

Technology and architecture

Technology stack: iSynth is based on technologies and frameworks such as Python, Django, JavaScript, Vue.js and Docker.

GUI: iSynth comes with a **web-based user interface** for direct user interaction. In addition, iSynth artefacts can also be edited using any text editor or IDE with Python support.

Database management systems: iSynth uses a **PostgreSQL, Oracle or SQLite** database to generate and hold synthetic data.

Interfaces to target applications: Synthetic data is deployed to the target applications via any available interface. File or SQL loaders (CSV, XML, JSON etc.), messages via JMS or MQ or web service calls (REST, SOAP) are commonly used to inject data. iSynth also has a mechanism to interact bidirectionally with target applications. When an application assigns business keys to its data objects, they have to be further propagated to other applications in the downstream data flow.

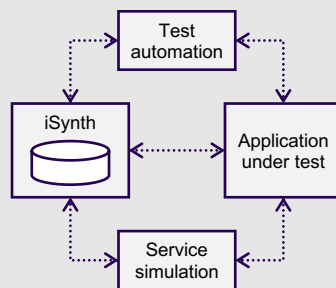
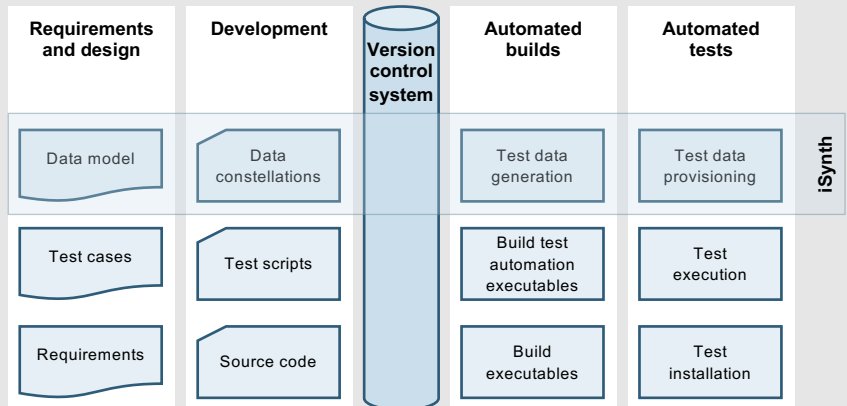
Pipeline integration: iSynth comes with a comprehensive OpenAPI-based REST interface that allows seamless integration into a CI/CD pipeline, enabling an end-to-end automated test process. This includes connecting to test management tools, service simulators or mocks and in order to provide the right data with the a defined variance as requested in the test cases.

Runtime environment: iSynth runs on container platforms (e.g., Kubernetes) as well as on Windows, macOS, Linux or any operating system that supports a recent Python version.

Version Control System: iSynth artefacts are text-based files, typically kept under version control (e.g., using git). The generated data does not have to be versioned since it can be produced any time based on the versioned artefacts.

Integration in DevOps

In order to run appropriate, automated tests in DevOps setups, it is essential to supply data that supports these tests exactly. With code-based data definitions under version control that keep pace with ongoing development, iSynth will automatically generate and deploy this data as part of a CI/CD pipeline so that subsequent test execution becomes a breeze.



iSynth delivers consistent data to both the application under test and the test automation tools and service simulations/virtualizations. So everything fits together.

iSynth can also be used to define test cases and test steps, including input data and expected results. In this way, test execution can be driven directly by synthetic data from iSynth. Input data embedded in test scripts becomes obsolete, which simplifies their creation and maintenance and saves costs.

Benefits

Cost reductions

- GDPR compliant by design, no measures for data protection needed
- Don't lose time searching for suitable test data after a data refresh
- Don't waste time manually creating and maintaining suitable test data
- Cut your execution or processing time with reduced datasets

Facilitators

- Off-shore software development based on rich synthetic test data
- Hassle-free data exchange with external partners (e.g., reproduce defects)
- Requirements enriched with data samples that foster a common understanding
- Test automation in CI/CD pipelines and in end-to-end environments

Quality improvements

- Early verification of interface specifications
- Early, implicit testing of APIs by using them for data deployments
- Automated tests with higher-level business relevance
- Parallel testing without side-effects

Contact

Have we aroused your interest? Learn more about the potential of synthetic data in the entire software lifecycle and about iSynth, our tool to create it.

Your contacts:

Josef Bösze, Partner, josef.boesze@itopia.ch

Ralph Schibli, Managing Partner, ralph.schibli@itopia.ch



corporate information technology

itopia ag - corporate information technology
Technoparkstrasse 1, CH-8005 Zurich
www.itopia.ch

© itopia ag, December 2022