



# USING GENROCKET FOR BANKING, FINANCIAL SERVICES, AND INSURANCE (BFSI)

GenRocket offers a powerful synthetic data generation platform that enables BSFI firms to generate high-quality, secure, and diverse synthetic data. GenRocket effectively addresses crucial testing and compliance challenges. This article covers the following topics:

- Key Applications of GenRocket in BFSI
- What GenRocket Features are Helpful in BFSI Environments?
- Platform Features Enhancing Financial Services Testing
- Example BFSI Use Case Scenarios
- How to Integrate GenRocket with BFSI Workflows



# Key Applications of GenRocket in BFSI

GenRocket effectively integrates into the Banking, Financial Services, and Insurance (BFSI) sector by addressing challenges in data provisioning, testing, compliance, and AI enablement. Here's how it adds value:

- **Data Privacy and Compliance Support** - Due to handling sensitive customer data, BFSI firms must adhere to strict compliance regulations. GenRocket is capable of generating synthetic data that simulates production data but is non-sensitive. You can also replace sensitive data values with realistic synthetic data equivalents while maintaining referential integrity across complex structures. GenRocket is the only synthetic data platform that requires no direct access to the production data environment. That's because GenRocket uses metadata (database schema, DDL, etc.) to create a data model that guides the synthetic data generation process.
- **Application Test Data for Core BFSI Use Cases** - GenRocket enables the provisioning of synthetic data for various core use cases, including Loan Origination Systems, Payment and Transaction Engines, Customer Onboarding, Risk and Compliance Platforms, and Insurance Policy Systems. GenRocket's **Test Data Cases** and **Real-Time Synthetic Data API** let teams replicate real-world BFSI conditions.
- **Accelerated and Comprehensive Testing** - Generate synthetic test data on-demand with extensive test coverage, reduced testing time, and dynamic workflow testing. With GenRocket, you can create positive, negative, and edge case data for thorough testing and simulate complex, real-time data workflows across systems to ensure robust end-to-end testing.
- **AI/ML Model Development and Testing** - BFSI institutions are increasingly adopting AI/ML for credit scoring, fraud detection, customer churn prediction, and personalization. Synthetic data is ideal for generating large datasets that do not contain sensitive information and are not present in production. GenRocket can help by generating diverse and balanced training datasets, creating rare or edge-case data, and mitigating bias.
- **Legacy System Modernization** - BFSI firms transitioning from mainframe or legacy systems can benefit from GenRocket by generating synthetic data in various formats to align with new system requirements while ensuring consistency and accuracy across complex data relationships during migration.
- **Development/Testing Life Cycle Acceleration** - GenRocket integrates with CI/CD pipelines, DevOps workflows, and Automation Frameworks commonly used in BFSI. You can generate synthetic test data in real-time, and it is compatible with tools such as Jenkins, Selenium, Tosca, HyperExecute, and more.

# What GenRocket Features are Helpful in BFSI Environments?

GenRocket offers several key features for generating test data in a BFSI environment. You can use it to handle complex data structures, address privacy needs, and meet the testing demands of banks, financial institutions, and insurers.

## 1. Quick Setup with Design-Driven Data Modeling and Project Configuration -

GenRocket enables easy onboarding of banking, insurance, and investment data models. You can easily import or create your data model in multiple ways. GenRocket supports 15+ data model inputs, including:

- [XTS Wizard \(Database Schemas\)](#) - Create an XTS file and import your table schema into GenRocket.
- [DDL](#) - Import a single DDL or a file containing multiple DDLs.
- [XSD](#) - Import an XML Schema Definition (XSD) that formerly describes the elements of nesting of an XML Document.
- [Spark Schema](#) - Import a Spark Schema file to create a JSON, Avro, or Parquet output.
- [JSON Schema File](#) - Enables the import of different and ever-growing JSON formats into the GenRocket web platform.
- [JSON](#), [CSV](#), and [Avro](#) files

## 2. Model Complex Relationships and Maintain Referential Integrity - Model complex BFSI workflows, such as loan origination, payment processing, or policy lifecycle management, with full data integrity.

- [Parent, Child, and Sibling Relationships](#) - Govern how complex relationships (Domain models) can be easily represented and manipulated with full referential integrity.
- [Organization Variables](#) - Variables that are global to all Projects within an Organization. Any Generator can reference them within an Attribute of a Domain or the self service module.
- [Organization Attributes](#) - Attributes that are global to all Projects within an Organization. Multiple Projects can reference their generated value without setting up the Attribute each time.
- [Master Projects](#) - Maintain referential integrity for a Domain (e.g., a User Domain) ACROSS applications and databases. They should be used when multiple databases or applications share a common Domain or set of Domains.
- [G-Families](#) - Break down complex data models by organizing them into smaller family units based on defined relationships.
- [G-Map Server](#) - Map values in memory to be maintained and used throughout a complex workflow.
- [CI/CD Pipeline Integration](#) - Integrate test data generation into your CI/CD pipeline for continuous and streamlined testing.

### 3. Rich Library of Intelligent, Pre-Built Generators for Designing Test Data -

GenRocket offers **750+ Generators**, including BFSI-specific ones. Generators create synthetic test data or query enumerated data and can be combined or linked to generate complex synthetic test data. Here are a few available [Generators](#):

- **Credit Card Numbers** (Visa, Mastercard, American Express, etc.)
- **Bank Account Numbers for Different Regions** (Austria, Belgium, Finland, Germany, etc.)
- **Transaction Record Values** (Monetary Values, Currency, etc.)
- **Personal Identifiers** (SSN, Tax ID, Passport)
- **Addresses, Phone Numbers, and Names** (localized for different regions)
- **Date and Time Generators** - Create realistic time and date data or simulate missing data for negative test results (e.g., null, blanks, improperly formatted values).
- **PermutationGen** - Exhaustively tests all value combinations across multiple attributes (e.g., account types × transaction types × status codes).

### 4. Self Service Scenario and Test Data Case Design - Design and control the volume/variety of generated test data for each imported/created data model.

- [Scenarios](#) - Instructions for what synthetic data to generate and how much. Scenarios generate data for one Domain and can be chained together.
- [Test Data Cases](#) - Create many variations and volumes of test data that modify the Scenario without manually making changes to the actual Scenario.
- [Test Data Queries](#) - Query data from a database, CSV file, or Excel file to use in your test data case.
- [G-Questionnaire](#) - Create a Questionnaire from a Test Data Case, which testers can quickly modify to generate the needed test data without having extensive knowledge of GenRocket.

### 5. Rule-Based Test Data Conditioning - GenRocket provides Generators and self service features for generating conditioned test data. Enables high-precision edge case coverage (e.g., expired policies, overdraft charges, unusual cash flows).

- [Conditional Logic Generators](#) - Generate the exact conditioned test data required to run a specific test and verify a particular outcome. Conditional Generators can be assigned to a Domain Attribute.
- [Test Data Rules](#) - Dynamically control how data is generated based on business logic. You can set up and organize rules into a suite. These rules ensure the generated test data complies with the defined logic. Conditional Generators can also be assigned within a rule, depending on the test data requirements.

- 6. Receivers Provide Multi-Output and Multi-Target Support** - Generate data in [formats](#) required by BFSI systems:
- Flat and Nested files (JSON, XML, EDI, Kafka streams, SQL statements).
  - XML and JSON for API Testing
  - [EDI](#) and [HL7](#) for cross-industry messaging
  - Database Insertion - Oracle, SQL Server, MySQL, PostgreSQL, MongoDB, and more.
  - Supports [SWIFT](#), [NACHA](#), BAI2, and custom financial payloads for mainframe and modern APIs.
  - Supports legacy core banking systems *and* modern cloud-native services simultaneously.
- 7. Real-Time Data Generation API** - Easily integrate with your CI/CD pipeline and testing tools by calling the [GenRocket Runtime API](#). Click [here](#) to see an index of API methods.
- Enables on-demand test data provisioning as part of automated pipelines.
  - Enables true shift-left testing for BFSI, where fast and realistic data are critical.
  - Generate data in real time for systems that test live transactions.
  - No stale test data — regenerate fresh data for each test run.
  - Easily integrates with Jenkins, GitHub Actions, [Selenium](#), JMeter, [TOSCA](#), and more.
- 8. Synthetic Data Replacement and File Masking** - Replace sensitive fields in production-like datasets with realistic synthetic values.
- [File Masking](#) - Replace sensitive values in a provided source file with synthetic data in generated files. Available for Delimited, Fixed, JSON, XML, ORC (Hadoop), and EDI files.
  - [G-Migration+](#) - Supports data subsetting and synthetic data masking, ensuring compliance with data privacy regulations.
  - **Partial Replacement** - Keep some non-sensitive columns intact while masking regulated fields.
- Note:** GenRocket does not have access to this information. It remains in your environment behind your corporate firewall. GenRocket generates synthetic test data based on metadata and rules.
- 9. Parallelized, High-Throughput Data Generation** - Scales to generate millions of records per minute. Partitioning and multi-threaded processing support high-speed parallel data generation.
- **High-volume data:** Generate millions of records quickly to simulate real transaction volumes.
  - **Variability:** Use permutations and data pools to avoid repetitive test data.
  - **Streaming to queues:** Pipe data to Kafka or message brokers to test event-driven systems.



## 10. Secure, Compliant, Enterprise-Ready Architecture

- Synthetic data is generated entirely from metadata and rules.
- Supports on-prem, cloud, and hybrid deployments.
- Offers role-based access controls.

## Platform Features Enhancing Financial Services Testing

Beyond Generators and Receivers, GenRocket's platform offers features that significantly benefit financial services testing:

- **[Project Categorization](#)** - Organize projects to make it easier for users to discover and leverage what is already available.
- **[Team Permissions](#)** - Ensure users can only access the appropriate Projects and manage permitted information, such as Domains or Scenarios, for those Projects.
- **[Test Data Stories and Epics \(G-Stories & G-Epics\)](#)** - Structure complex testing workflows by grouping related test data cases.
- **[G-Portal](#)** - Provide self-service capabilities for testers and developers to request test data.
- **[G-Delta](#)** - Automatically detects database schema changes in your environment and updates the test data project so that the generated test data reflects the latest database structure.



# Example BFSI Use Case Scenarios

The following are a few use cases for which GenRocket can be used for testing:

- 1. Core Banking System Testing** - Rolling out a banking upgrade where the QA teams need to test account creation, transactions, overdraft handling, etc. GenRocket can:
  - Generate thousands of customer records with realistic information.
  - Link customers to multiple types of accounts (e.g., checking, savings, credit card).
  - Generate deposits, withdrawals, transfers, fees, and other transactions.
  - Create edge cases such as zero balances and overdrafts.
  - **Benefit** - Allows you to run full automated regression tests nightly with fresh, production-like data, ensuring full test data coverage to verify the core banking platform operates as expected.
- 2. Payment Gateway and Fraud Scenario Simulation** - A payment processor wants to test its fraud detection engine by feeding realistic transactions, including unusual and suspicious patterns.
  - Generate millions of payment transactions with various merchants, currencies, and locations.
  - Include duplicate transactions, suspicious amounts, and rapid, repetitive payments.
  - Flag some transactions as potential fraud.
  - Deliver data in real-time via API or CI/CD pipeline to detect fraud.
  - **Benefit** - Machine learning fraud models can be tested and tuned based on high-risk and low-risk transactions without exposing real data.
- 3. Loan Origination and Underwriting Workflow** - Test a loan processing system from application submission, approval, funding, and repayment.
  - Generate synthetic applications with varying incomes, credit scores, and collateral types.
  - Simulate valid/rejected applications.
  - Simulate regular or late payments.
  - Validate interest calculation logic for loan projects.
  - **Benefit** - End-to-end testing of personal loans, mortgage, and auto loan workflows, with realistic edge conditions such as defaults or early loan payoffs.
- 4. Insurance Policy Lifecycle and Claims** - Test policy issuance, renewals, and claims processing.
  - Generate a large number of policies across different products (life, auto, etc.).
  - Create realistic policyholder data with dependents and beneficiaries.
  - Simulate claims, including small losses, large losses, and multiple claims.
  - Test fraud detection by including suspicious or fraudulent claims.
  - **Benefit** - Helps ensure that high volumes, diverse products, and fraudulent edge cases can be handled.

5. **Regulatory Reporting and Stress Tests** - Prepare for annual stress tests and regulatory submissions (e.g., BCBS 239).
  - Generate transaction records spanning multiple years.
  - Simulate things such as sudden interest rate changes.
  - Validate that data pipelines function correctly and report the correct results.
  - **Benefit** - Ensures that risk models and reporting workflows meet the performance requirements in adverse conditions.
6. **Customer Onboarding** - Testing a digital onboarding application that has automated checks.
  - Create diverse customer profiles for various countries, age groups, and income levels.
  - Include valid and invalid ID documents.
  - See how incomplete or conflicting information is handled during onboarding.
  - **Benefit** - Validate the onboarding process for real-world situations. Doing so ensures a smooth customer experience and compliance.

## How to Integrate GenRocket with BFSI Workflows

1. **Model Data Entities** - Customers, Policies, Loans, Transactions, Accounts, etc.
2. **Design Test Data Cases** - Based on banking and insurance workflows.
3. **Deploy in Multiple Data Outputs** - SWIFT MT, BAI, Blockchain NEXO, ISO 8583, NACHA, and direct database inserts, among others.
4. **Manage the Test Data Lifecycle** - Organize test data projects into distributed repositories where changes are automatically synchronized, and Test Data Cases are available for immediate execution using a self service portal.

