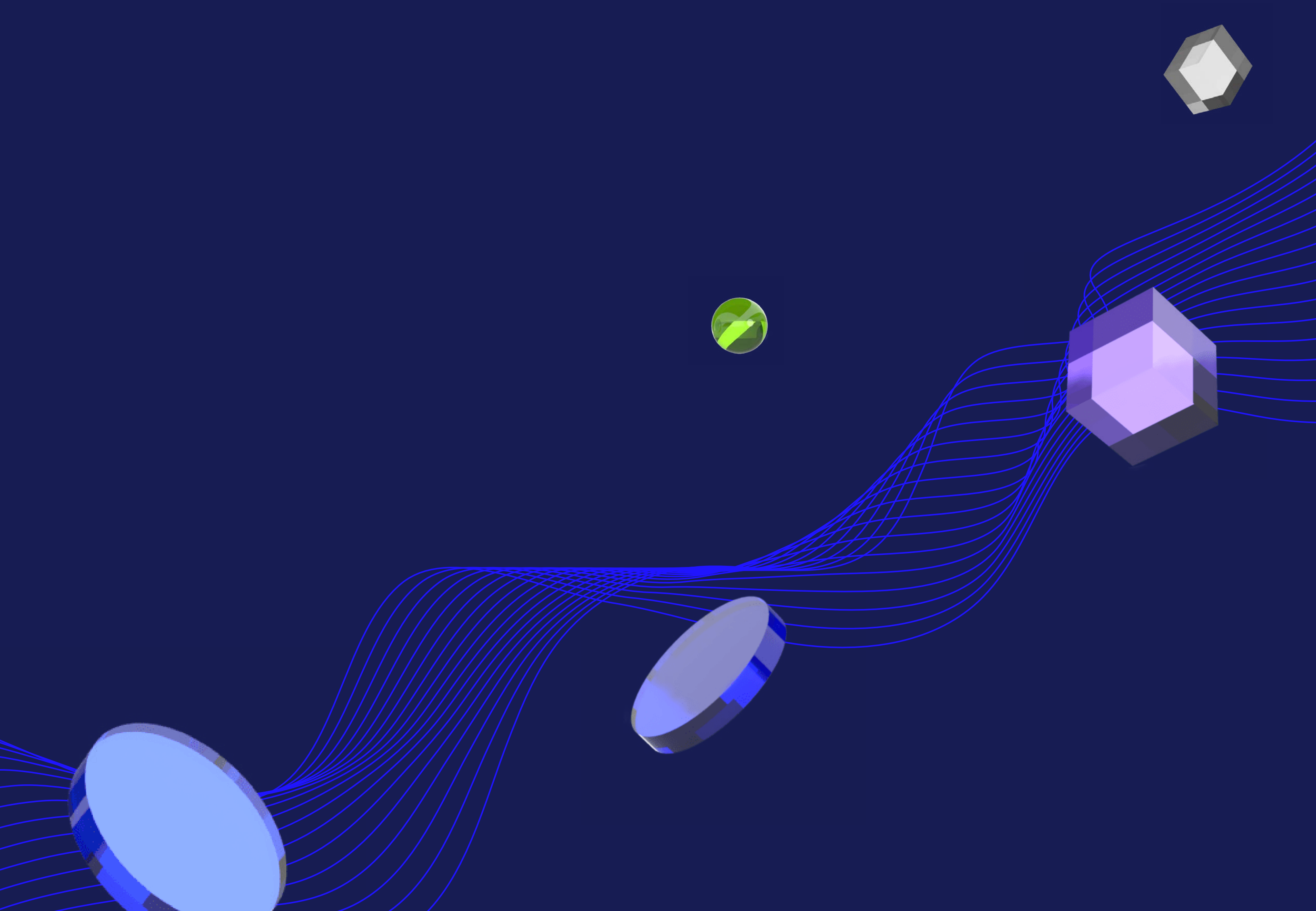
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF DATA SCIENCE

Data Streaming

Nanodegree Program Syllabus



Overview

The Data Streaming Nanodegree program provides learners with the latest skills to process data in real-time by building fluency in modern data engineering tools, such as Apache Spark, Kafka, Spark Streaming, and Kafka Streaming.

Learning Objectives

A graduate of this program will be able to:

- Understand the components of data streaming systems.
- Ingest data in real-time using Apache Kafka and Spark and run analyses.
- Use the Faust Stream Processing Python library to build a real-time stream-based application.
- Compile real-time data and run live analytics, as well as draw insights from reports generated by the streaming console.
- Learn about the Kafka ecosystem, and the types of problems each solution is designed to solve.
- Use the Confluent Kafka Python library for simple topic management, production, and consumption.
- Explain the components of Spark Streaming (architecture and API), integrate Apache Spark Structured Streaming and Apache Kafka, manipulate data using Spark, and understand the statistical report generated by the Structured Streaming console.

Program information

Estimated Time

2 months at 10hrs/week*

Skill Level

Advanced

Prerequisites

Learners should be equipped with intermediate SQL skills, Python knowledge, and experience with ETL. Additionally, it is recommended that they are familiar with traditional batch processing and traditional service architectures.

Required Hardware/Software

Learners need access to the internet and a 64-bit computer.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Foundations of Data Streaming

This course focuses on common data streaming topics and skills including: consumers, producers, Connect Sources, Sinks, Kafka REST Proxy, data schemas, stream processing, KSQL, and Faust.

Course Project

Optimize Chicago Bus & Train Availability Using Kafka

Learners will stream public transit status of trains in real time. They will optimize the availability of buses and trains in Chicago based on streaming data. They will learn how to have their own Python code produce events, use REST Proxy to send events over HTTP, and use Kafka Connect to collect data from a Postgres database to produce streaming data from a number of sources into Kafka. Then, learners will use KSQL to combine related data models into a single topic ready for consumption by the downstream Python applications, and complete a simple Python application that ingests data from the Kafka topics for analysis. Finally, they will use the Faust Python Stream Processing library to further transform train station data into a more streamlined representation. Using stateful processing, this library will show whether passenger volume is increasing, decreasing, or staying steady.

Lesson 1

Introduction to Stream Processing

- Describe and explain streaming data stores and stream processing.
- Describe and explain real-world usages of stream processing.
- Describe and explain append-only logs, events, and how stream processing differs from batch processing.
- Utilize Kafka CLI tools and the Confluent Kafka Python library for topic management, production, and consumption.

Lesson 2

Apache Kafka

- Understand Kafka architecture, topics, and configuration.
- Utilize Confluent Kafka Python to create topics and configuration.
- Understand Kafka producers, consumers, and configuration.
- Utilize Confluent Kafka Python to create producers and configuration.
- Utilize Confluent Kafka Python to create topics, configuration, and manage offsets.
- Describe and explain user privacy considerations.
- Describe and explain performance monitoring for consumers, producers, and the cluster itself.

Lesson 3

Data Schemas & Apache Avro

- Understand what a data schema is and the value it provides.
- Understand what Apache Avro is and what value it provides.
- Utilize AvroProducer and AvroConsumer in Confluent Kafka Python.
- Describe and explain schema evolution and data compatibility types.
- Utilize Schema Registry components in Confluent Kafka Python to manage compatibility.

Lesson 4

Kafka Connect & Rest Proxy

- Describe and explain what problem Kafka Connect solves for and where it would be more appropriate than a traditional consumer.
- Describe and explain common connectors and how they work.
- Utilize Kafka Connect FIleStream and JDBC Source and Sink.
- Describe and explain what problem Kafka REST Proxy solves for and where it would be more appropriate than alternatives.
- Describe, explain, and utilize the REST Proxy metadata and administrative APIs.
- Describe and explain the REST Proxy consumer APIs.
- Utilize the REST Proxy consumer, subscription, and offset APIs.
- Describe, explain, and utilize the REST Proxy producer APIs.

Lesson 5

Stream Processing Fundamentals

- Describe and explain common scenarios for stream processing and where you would use stream versus batch.
- Describe and explain common stream processing strategies.
- Describe and explain how time and windowing works in stream processing.
- Describe and explain what a stream versus a table is in stream processing, and where you would use on over the other.
- Describe and explain how data storage works in stream processing applications and why it is needed.

Lesson 6

Stream Processing with Faust

- Describe and explain the Faust Stream Processing Python library and how it fits into the ecosystem relative to solutions like Kafka Streams.
- Describe and explain Faust stream-based processing.
- Utilize Faust to create a stream-based application.
- Describe and explain how Faust table-based processing works.
- Utilize Faust to create a table-based application.
- Describe and explain Faust processors and function usage.
- Utilize Faust processor and function.
- Describe and explain Faust serialization and deserialization.
- Utilize Faust serialization and deserialization.

Lesson 7

KSQL

- Describe and explain how KSQL fits into the Kafka ecosystem and why you would choose it over a stream processing application built from scratch.
- Describe and explain KSQL architecture.
- Describe and explain how to create KSQL streams and tables from topics.
- Understand the importance of KEY and schema transformations.
- Utilize KSQL to create tables and streams.
- Describe and explain KSQL selection syntax.
- Utilize KSQL syntax to query tables and streams.
- Describe and explain KSQL windowing.
- Utilize KSQL windowing within the context of table analysis.
- Describe and explain KSQL grouping and aggregates.
- Utilize KSQL grouping and aggregates within queries.

Course 2

Streaming API Development & Documentation

The goal of this course is to grow expertise in the components of streaming data systems, and build a real time analytics application. Specifically, learners will be able to identify components of Spark Streaming (architecture and API), build a continuous application with Structured Streaming, consume and process data from Apache Kafka with Spark Structured Streaming (including setting up and running a Spark Cluster), create a DataFrame as an aggregation of source DataFrames, sink a composite DataFrame to Kafka, and visually inspect a data sink for accuracy.

Course Project

Evaluate Human Balance with Spark Streaming

In this project, learners will work with a real-life application called the Step Trending Electronic Data Interface (STEDI). It is a working application used to assess fall risk for seniors. When a senior takes a test, they are scored using an index which reflects the likelihood of falling, and potentially sustaining an injury in the course of walking. STEDI uses a Redis datastore for risk score and other data. The data science team has completed a working graph for population risk at a STEDI clinic. The problem is the data is not populated yet. Learners will work with Kafka Connect Redis Source events and Business Events to create a Kafka topic containing anonymized risk scores of seniors in the clinic.

Lesson 1

Streaming Dataframes

- Start a Spark cluster and deploy a Spark application.
- Create a Spark Streaming DataFrame with a Kafka source.
- Create a Spark view.
- Query a Spark view

Lesson 2

Joins & JSON

- Parse a JSON Payload into separate fields for Analysis.
- Join two streaming DataFrames from different data sources.
- Write a streaming DataFrame to Kafka with aggregated data.

Lesson 3

Redis, Base64 & JSON

- Manually save to Redis and read the same data from a Kafka topic.
- Sink a subset of JSON fields.

Meet your instructors.

Ben Goldberg

Staff Engineer at SpotHero

In his career as an engineer, Ben Goldberg has worked in fields ranging from computer vision to natural language processing. At SpotHero, he founded and built out their data engineering team, using Airflow as one of the key technologies.

David Drummond

VP of Engineering at Insight

David is VP of Engineering at Insight where he enjoys breaking down difficult concepts and helping others learn data engineering. David has a PhD in Physics from UC Riverside.

Judit Lantos

Senior Data Engineer at Netflix

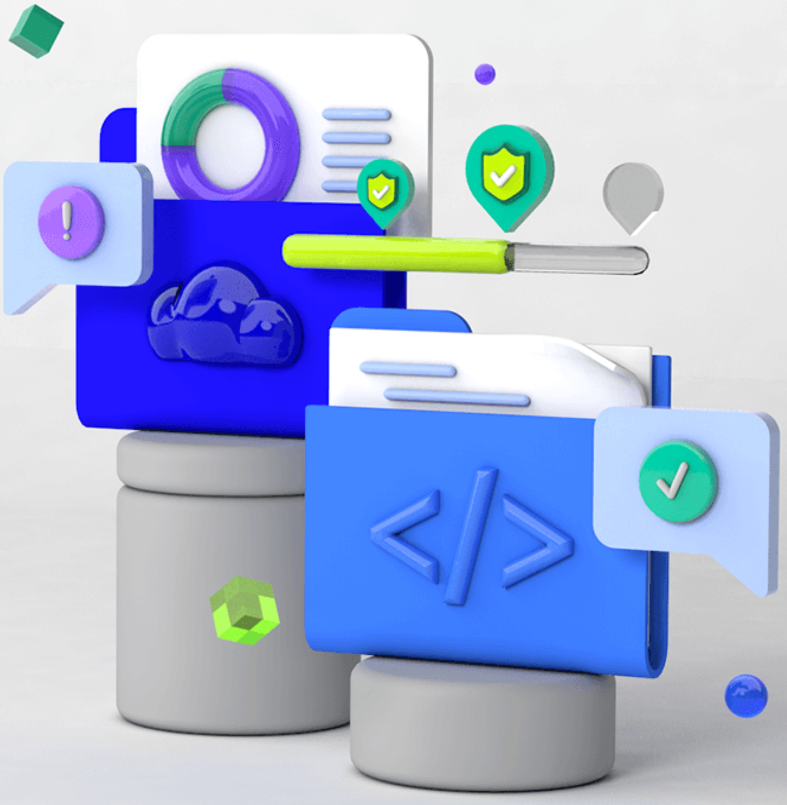
Currently, Judit is a senior data engineer at Netflix. Formerly a data engineer at Split, where she worked on the statistical engine of their full-stack experimentation platform, she has also been an instructor at Insight Data Science, helping software engineers and academic coders transition to DE roles.

Sean Murdock

Faculty at BYU - Idaho

Sean has worked as an architect or software engineer for Columbia Ultimate, Firstsource Global, Intermountain Healthcare, General Motors, The Church of Jesus Christ, Northrup Grumman, Zions Bank, and Ancestry. He currently teaches DevOps and cybersecurity at Brigham Young University - Idaho.

Udacity’s learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren’t just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

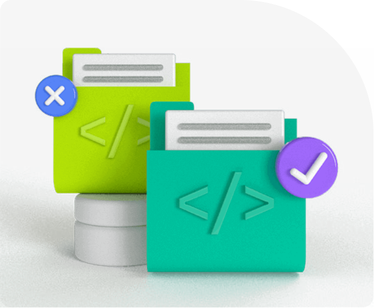
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

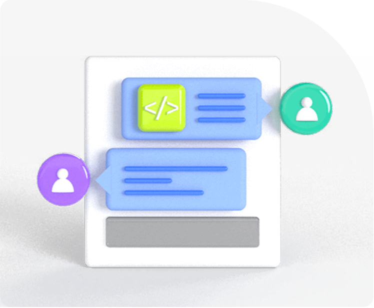
Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

Verify skills mastery.

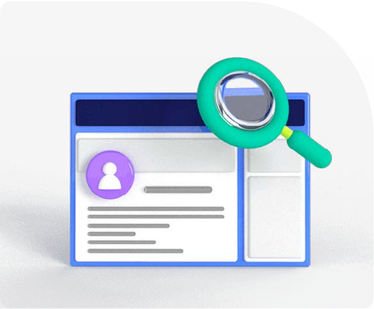
- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.



Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →

12.09.22 | V1.0