UDACITY

NANODEGREE PROGRAM SYLLABUS

Data Streaming

Overview

The ultimate goal of the Data Streaming Nanodegree program is to provide students 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. A graduate of this program will be able to:

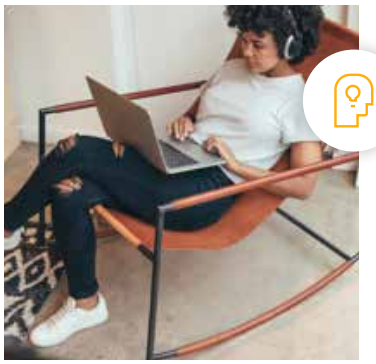
- Understand the components of data streaming systems. Ingest data in real-time using Apache Kafka and Spark and run analysis
- Use the Faust Stream Processing Python library to build a real-time stream-based application. Compile real-time data and run live analytics, as well as draw insights from reports 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 read DataFrames in the Spark Streaming Console.

This program is comprised of 2 courses and 2 projects. Each project you build will be an opportunity to demonstrate what you've learned in the course, and will demonstrate to potential employers that you have skills in these areas.

Prerequisite Knowledge: Intermediate SQL, and Python. And experience with ETL. Basic familiarity with traditional batch processing and traditional service architectures is desired, but not required.

Estimated Time: 2 Months at 5-10 hrs/week

Prerequisites: Intermediate SQL, Python, and experience with ETL



Flexible Learning: Self-paced, so you can learn on the schedule that works best for you.

Need Help?
udacity.com/advisor
Discuss this program with an enrollment advisor.

UDACITY

Course 1: Foundations of Data Streaming, and SQL & Data Modeling for the Web

The goal of this course is to demonstrate knowledge of the tools taught throughout, including Kafka Consumers, Producers, & Topics; Kafka Connect Sources and Sinks, Kafka REST Proxy for producing data over REST, Data Schemas with JSON and Apache Avro/Schema Registry, Stream Processing with the Faust Python Library, and Stream Processing with KSQL.

Course Project
Optimize Chicago Bus and Train Availability Using Kafka

For your first project, you'll be streaming public transit status using Kafka and the Kafka ecosystem to build a stream processing application that shows the status of trains in real-time. Based on the skills you learn, you will be able to optimize the availability of buses and trains in Chicago based on streaming data. You will learn how to have your 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, you 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, you 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.

	LEARNING OUTCOMES	
LESSON ONE	**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

	LEARNING OUTCOMES	
LESSON TWO	**Apache Kafka**	• Describe and explain Kafka architecture • Describe and explain Kafka topics and configuration • Utilize Confluent Kafka Python to create topics and configuration • Describe and explain 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 THREE	**Data Schemas and Apache Avro**	• Describe and explain what a data schema is and what value it provides • Describe and explain 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 FOUR	**Kafka Connect and 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 Source and Sink • Utilize Kafka Connect JDBC Source and Sink • Describe and explain what problem Kafka REST Proxy solves for and where it would be more appropriate than alternatives • Describe and explain the REST Proxy metadata and administrative APIs • Utilize the REST Proxy administrative and metadata APIs • Describe and explain the REST Proxy consumer APIs • Utilize the REST Proxy consumer, subscription, and offset APIs • Describe and explain the REST Proxy producer APIs • Utilize the REST Proxy producer APIs

		LEARNING OUTCOMES
LESSON SIX	**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 SEVEN	**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 EIGHT	**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 and Documentation

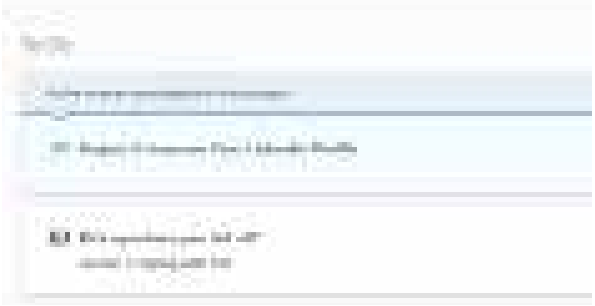
The goal of this course is to grow your expertise in the components of streaming data systems, and build a real time analytics application. Specifically, you will be able to identify components of Spark Streaming (architecture and API), build a continuous application with Structured Streaming, consume 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.

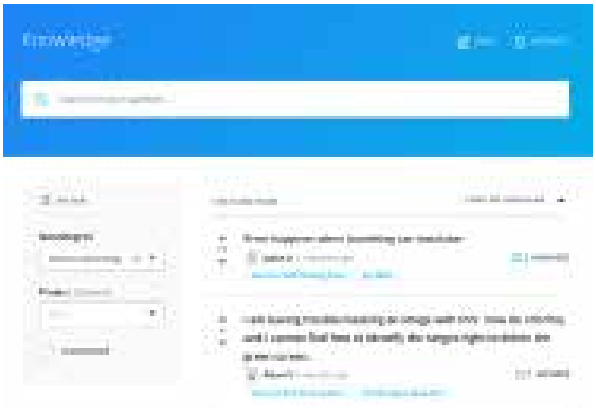
Course Project
Evaluate Human Balance with Spark Streaming

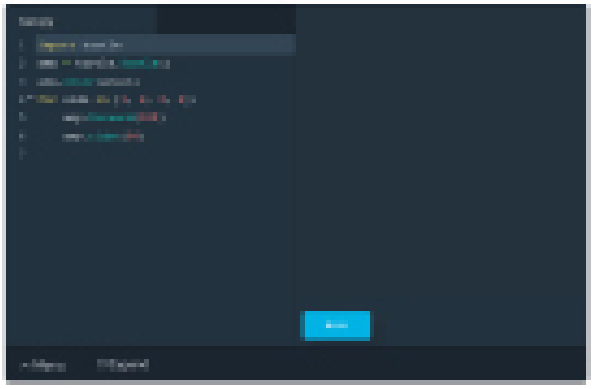
In this project, you will work with a real-life application called the Step Trending Electronic Data Interface (STEDI). It is a working application used to assess fall risk for seniors. When a senior takes a test, they are scored using an index which reflects the likelihood of falling, and potentially 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. You 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.

	LEARNING OUTCOMES	
LESSON ONE	**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 TWO	**Joins and 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 THREE	**Redis, Base64 and JSON**	• Manually Save to Redis and Read the Same Data from a Kafka Topic • Parse Base64 Encoded Information • Sink a Subset of JSON Fields

Our Classroom Experience







REAL-WORLD PROJECTS
Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

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

STUDENT HUB
Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with fellow students in your program as you support and learn from each other.

WORKSPACES
See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES
Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS
Preschedule your study times and save them to your personal calendar to create a custom study plan. Program regular reminders to keep track of your progress toward your goals and completion of your program.

PROGRESS TRACKER
Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

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, 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.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW

WHY SHOULD I ENROLL?

As businesses increasingly rely on applications that produce and process data in real-time, data streaming is an increasingly in-demand skill for data engineers. The Data Streaming Nanodegree program will prepare you for the cutting edge of data engineering as more and more companies look to derive live insights from data at scale.

Students will learn how to process data in real-time by building fluency in modern data engineering tools, such as Apache Spark, Kafka, Spark Streaming, and Kafka Streaming.

You'll start by understanding the components of data streaming systems. You'll then build a real-time analytics application. You will also compile data and run analytics, as well as draw insights from reports generated by the streaming console.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

This program is designed to upskill experienced Software Engineers and Data Engineers to learn the latest advancements in data processing, sending data records continuously to support live updating.

The projects in the Data Streaming Nanodegree program will prepare you to develop systems and applications capable of interpreting data in real-time, and position you for roles in all industries that require live data processing for functions including big data, cloud computing, web personalization, fraud detection, sensor monitoring, anomaly detection, supply chain maintenance, location-based services, and much more.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

This program is intended for software engineers looking to build real-time data processing proficiency, as well as data engineers looking to enhance their existing skill set with the next advancement in data engineering.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

Prerequisite Knowledge: Intermediate SQL, and Python. And experience with ETL. Basic familiarity with traditional batch processing and traditional service architectures is desired, but not required.

FAQs Continued

NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

Udacity's **Programming for Data Science with Python** Nanodegree program is great preparation for the Data Engineer Nanodegree program. You'll learn to code with Python and SQL.

Similarly, the **Data Engineering** Nanodegree program is great preparation for the Data Streaming Nanodegree program.

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Data Streaming Nanodegree program is comprised of content and curriculum to support two projects. We estimate that students can complete the program in two months, working five to ten hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided, and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card on the Nanodegree program overview page. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** for other policies around the terms of access to our Nanodegree programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

There are no software and version requirements to complete this Nanodegree program. All coursework and projects can be completed via Student Workspaces in the Udacity online classroom.