UDACITY
FOR ENTERPRISE

THE SCHOOL OF PROGRAMMING AND DEVELOPMENT

Java Web Developer

NANODEGREE SYLLABUS

Overview

Java Web Developer Nanodegree Program

The ultimate goal of the Java Web Developer Nanodegree program is to equip students with the unique skills they need to build enterprise-scale applications with Java. A graduate of this program will be able to:

• Understand the fundamentals of the Spring Boot framework and associated integrations and plugins.

• Describe the differences between web services, APIs, and microservices, develop REST and GraphQL APIs, and learn how to secure, consume, document, and test those APIs and web services.

• Build applications that read and write to relational databases using both the Java Persistence API (JPA) and SQL. Use standard design patterns to make your persistence layer easy to test and integrate with a Spring Boot application.

• Learn about Git, version control, and best practices for authorization and authentication. Use Jenkins to build CI/CD pipeline to deploy code to production.

Program Information

ESTIMATED TIME
4 months
Study 10 hours/week

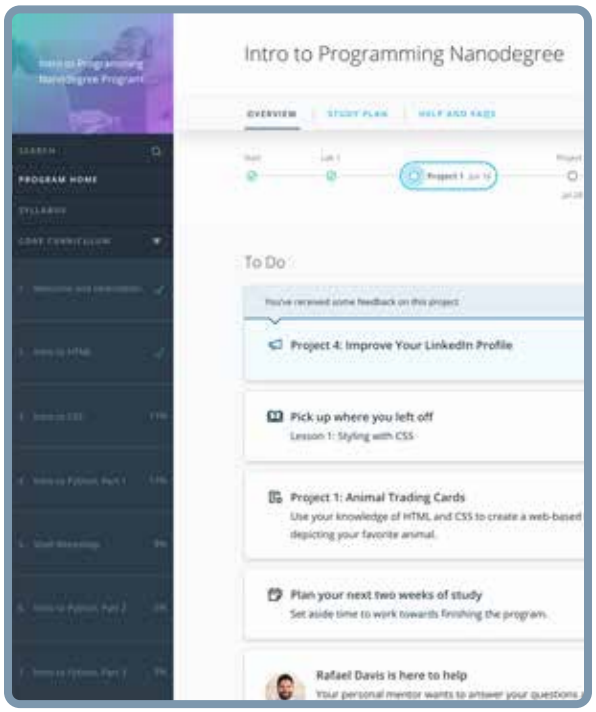
LEVEL
Practitioner

PREREQUISITES
Intermediate knowledge of Java. Some web development experience desirable, but not required. Students should also be able to: Initialize and use primitive Java data types (i.e. float, int, etc), select and use Collections from java.util. Collections, design and create classes and class methods in Java, create interfaces and subclasses in Java, launch Java applications from an IDE, and write basic queries in SQL.

HARDWARE/SOFTWARE REQUIRED
Access to the internet and a 64-bit computer.

LEARN MORE ABOUT THIS NANODEGREE
Contact us at enterpriseNDs@udacity.com

Our Classroom Experience



REAL-WORLD PROJECTS

Learners build new skills through industry-relevant projects and receive personalized feedback from our network of 900+ project reviewers. Our simple user interface makes it easy to submit projects as often as needed and receive unlimited feedback.

KNOWLEDGE

Answers to most questions can be found with Knowledge, our proprietary wiki. Learners can search questions asked by others and discover in real-time how to solve challenges.

LEARNER HUB

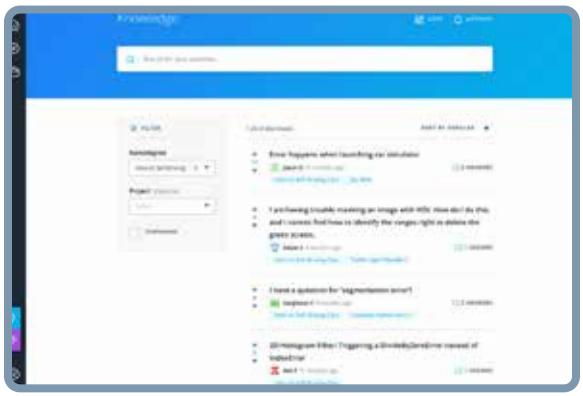
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

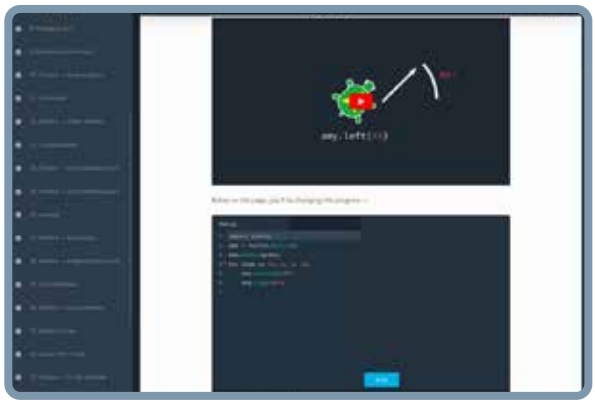
WORKSPACES

Learners can check the output and quality of their code by testing it on interactive workspaces that are integrated into the classroom.

QUIZZES

Understanding concepts learned during lessons is made simple with auto-graded quizzes. Learners can easily go back and brush up on concepts at anytime during the course.





CUSTOM STUDY PLANS

Mentors create a custom study plan tailored to learners' needs. This plan keeps track of progress toward learner goals.

PROGRESS TRACKER

Personalized milestone reminders help learners stay on track and focused as they work to complete their Nanodegree program.

Learn with the Best

Peter Zastoupil

INSTRUCTOR

Peter Zastoupil is an enterprise developer and technical administrator. He has seven years of on-the-job experience building features for massive enterprise Java servers, and over four years of teaching those skills to new developers.

Kesha Williams

INSTRUCTOR

Kesha has over 20 years experience in software development and is a software engineering manager at Chick-fil-A, routinely leading innovation teams in proving out the use of cloud services to solve complex business problems. She was recently named an Alexa Champion by Amazon.

Alex Pritchard

INSTRUCTOR

Alex is a Senior Software Engineer for CPA Global. He is excited to combine his background as a music educator with more than a decade of enterprise Java experience to help create this practical course on Data Stores and Persistence.

Sareeta Panda

INSTRUCTOR

Sareeta is a Java enthusiast and Senior Developer at Walmart e-Commerce. She specializes in Enterprise Application development with Java and Kafka, NoSQL, Spring security, and CI/CD. Sareeta has over a decade of experience, spanning recently acquired startups to top Fortune 500 companies.

Course 1: Spring Boot Basics

Learn the fundamentals of Java while being introduced to a Spring Boot framework and associated integrations and plugins.

Project: Build a Web-based Personal Storage Application

In this project, students will use the skills acquired in the first course to build a web-based personal storage application: SuperDuperDrive! Students will implement user-facing features like file, note, and secure credential storage with industry-standard, full-stack development tools. Building on Spring Boot as a base, students will use Spring MVC and Thymeleaf to build a Java-backed web app. Using H2 as an inmemory database, students will store user data with MyBatis, a deadsimple ORM library, and secure that data from unauthorized access with Spring Security. Finally, to round out the development cycle and verify that the app is feature-complete, students will implement a series of automated user tests with JUnit and Selenium.

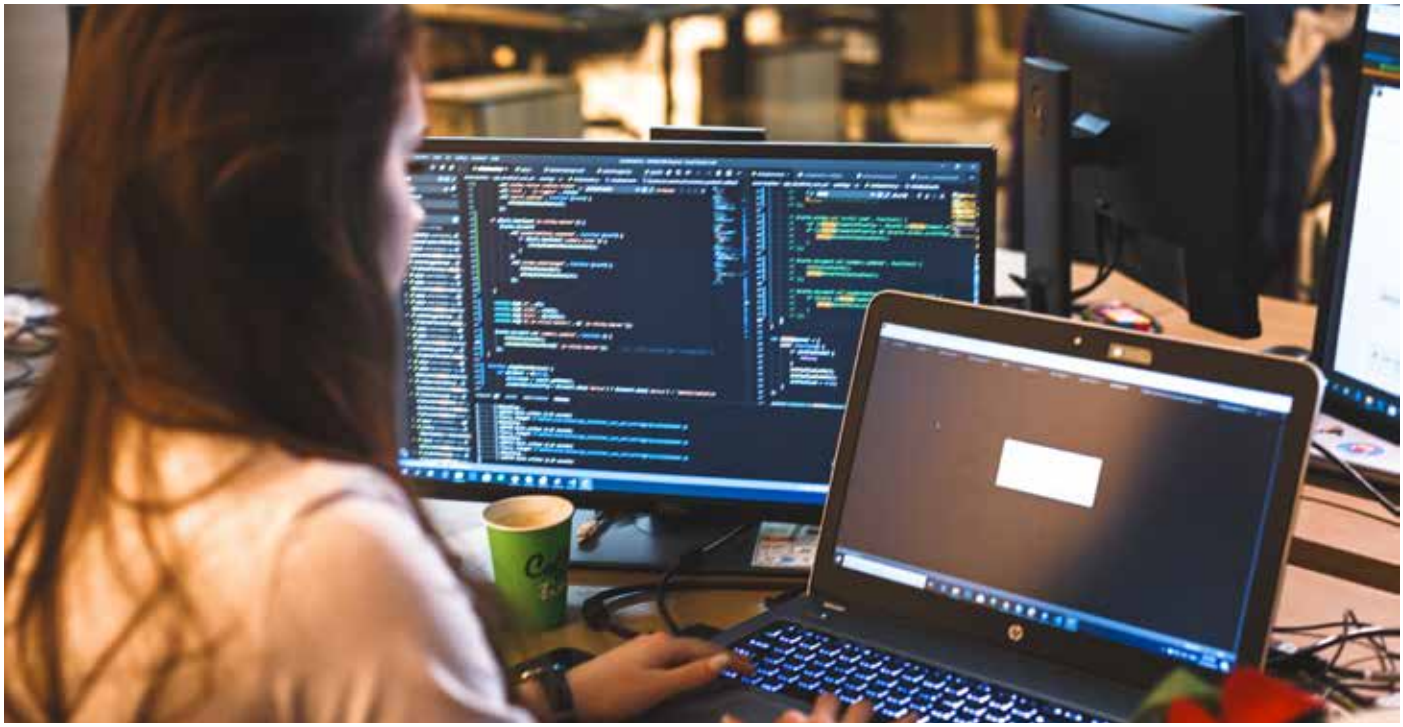
LESSON TITLE	LEARNING OUTCOMES
WEB DEVELOPMENT IN JAVA	• Describe how the Java Application Server facilitates web development. • Identify the role of a Servlet within a Java Application Server. • Identify the role of Spring as a Servlet application tool. • Add and update project dependencies within a Maven POM file. • Choose appropriate starter packs for Spring depending on the application's requirements.

Course 1: Spring Boot Basics, cont.

LESSON TITLE	LEARNING OUTCOMES
SPRING BOOT BASICS FOR WEB DEVELOPMENT	• Set up and configure a working Spring Boot Environment for web development. • Configure a Spring Boot application within Java using annotations and factory methods. • Identify Spring registered components in Java applications based on Spring annotations. • Annotate code to register custom components with a Spring App. • Identify and customize essential properties for the Spring Server.
SPRING MVC AND THYMELEAF	• Design HTML templates with Thymeleaf and populate HTML templates with Spring's MVC data model. • Explain how a controller populates the data model for a given view. • Identify the relationship between controller endpoint return values and the templates that are displayed. • Identify the role of the model object passed to controller endpoint methods. • Write variable resolution expressions within a ThymeLeaf template to access model data.
DATA PERSISTENCE AND SECURITY	• Identify mappings between Java objects and SQL tables and leverage those mappings to connect an application with a data store. • Explain how ORM leverages similarities between Java data types and SQL data types to reduce development time and programmer error. • Write MYBatis SQL template queries using an application's data model. • Explain how the @Mapper annotation functions in the Spring App context as a component annotation. • Securely store user credentials in a database. • Query user information and identify accessible pages based on that information. • Use Spring security to automatically filter web traffic based on that information.

Course 1: Spring Boot Basics, cont.

LESSON TITLE	LEARNING OUTCOMES
TESTING	• Use Selenium/Webdriver to automatically perform user actions in order to test the functionality of web pages. • Define JUnit test classes with the @Test annotation. • Use the JUnit assertion class to test specific success or failure points. • DRun a suite of JUnit tests from their IDE and interpret the results. • Navigate to specific URLs with the Selenium web driver. • Interact with queried elements from Selenium in the manner of a user to test that functionality exists as intended. • Write JUnit tests using these techniques to test individual features of a web app. • Organize tests into Page objects so that the application structure is mirrored by the test structure.



Course 2: Web Services and APIs

Explore the differences between web services, APIs, and microservices. Develop REST and GraphQL APIs, and learn how to secure, consume, document, and test those APIs and web services.

Project: Build the Backend System for a Car Website

In this course, the student will build a backend system for a website of cars. This backend will be composed of vehicles list services, pricing services, and location services as mentioned below: Vehicles API - a REST API to maintain vehicles data (CRUD), Pricing Service - a REST API to retrieve the price of a vehicle, and Location API - a HTTP client to retrieve the location of the vehicle. In the project, students will use Java APIs and frameworks to integrate different services using different communication styles. Students will write the CRUD operations to store and retrieve vehicle data and implement an HTTP client to retrieve the address of the vehicle given the latitude and longitude. Students will also integrate the clients (Vehicle API) with pricing services to retrieve the price. Lastly, students will learn to use Swagger to efficiently create documentation for their APIs. During the development of these steps, the student will be guided to write unit tests, error handling, logging, and other best practices.

LESSON TITLE	LEARNING OUTCOMES
WEB SERVICES & APIS OVERVIEW	• Describe web services and their advantages. • Describe how web services communicate. • Explore the differences between web services, APIs, and microservices.
DEVELOP REST APIS WITH SPRING BOOT	• Describe the REST architectural style and the importance of data formats. • Develop a REST API using Spring Boot and incorporate exception handling. • Use proper HTTP response codes.

Course 2: Web Services and APIs, cont.

LESSON TITLE	LEARNING OUTCOMES
DEVELOP GRAPHQL APIS WITH SPRING BOOT	• Describe GraphQL and its advantages over REST. • Create a GraphQL schema. • Develop a GraphQL server and API using Spring Boot. • Use GraphQL to execute queries and operations on data.
DEVELOP MICROSERVICES WITH SPRING BOOT	• Describe the Microservices Architecture (MSA). • Expose a microservice using Spring Boot. • Register a microservice.
SECURE API ENDPOINTS WITH SPRING SECURITY	• Describe Spring Security. • Explain the differences between authentication vs authorization. • Incorporate Basic Authentication practices to secure an API.
CONSUME WEB SERVICES AND APIs	• Consume a REST API. • Consume a SOAP-based web servicer. • Fetch and process XML and JSON.
DOCUMENT REST APIs	• Describe Swagger, a n open-source software framework to design, build, document, and consume RESTful web services. • Add Swagger annotations to model. • Generate API documentation.
TEST REST APIs	• Describe and explain unit and integration testing. • Incorporate unit and integration testing into a REST API.

Course 3: Data Stores & Persistence

Build applications that read and write to relational databases using both the Java Persistence API (JPA) and SQL. Use standard design patterns to make your persistence layer easy to test and integrate with a Spring Boot application.

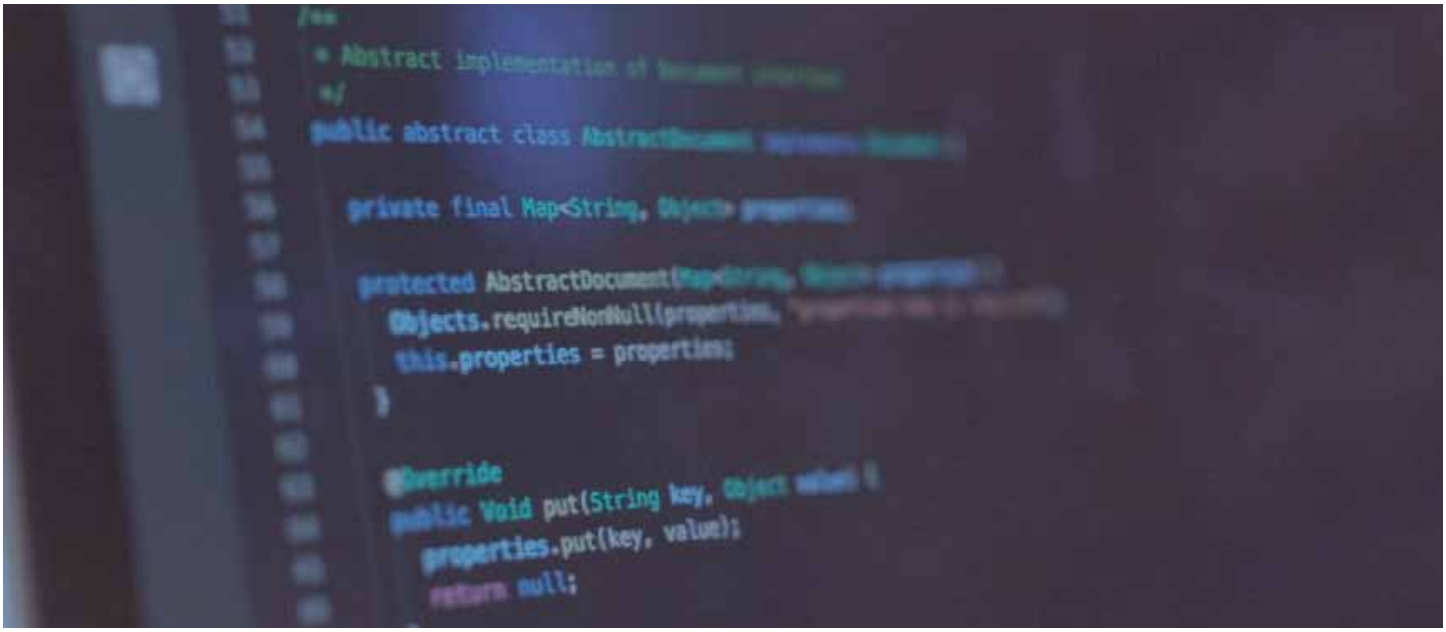
Project: Design the Data Model for a SaaS Application

Students will design and implement the data model for Critter Chronologer, a Software as a Service application that provides a scheduling interface for small businesses that take care of animals. This enterprise project will allow users to create schedules that associate pets, owners, and employees with calendar events. Students will configure their application to connect to an external database and use both JDBC and Hibernate to persist changes to it. Basic CRUD operations will be exposed via a REST controller layer so that students can test their application using Postman.

LESSON TITLE	LEARNING OUTCOMES
DATA IN MULTITIER ARCHITECTURE	• Design Entities that map Java data types to database structures. • Represent complex associations between Entities in persistence. • Identify and select inheritance strategies. • Isolate Entity scope through the use of annotations and DTOs.
JAVA PERSISTENCE API	• Understand and utilize key concepts in Object Relational Mapping (ORM) such as Persistence Context and Entity Manager, and learn about the Repository design pattern. • Propagate retrievals and persists with the help of Lazy Loading and Cascading. • Write and execute object queries in Java using JPQL. • Build implementations for your Repository methods automatically with Spring Data JPA. • Control the execution of queries through Transactions.

Course 3: Data Stores & Persistence, cont.

LESSON TITLE	LEARNING OUTCOMES
CONNECTING TO DATA SOURCES	• Connect Spring Boot to both internal and external data sources. • Customize Spring DataSource construction and injection. • Use Spring and Hibernate to automatically initialize your data.sources. • Configure unit tests to use different data sources.
PERSISTENCE WITHOUT JPA	• Learn about the differences in Data Object design when retrieving data with SQL. • Initialize Data Sources with SQL scripts. • Use the Data Access Object design pattern. • Execute SQL queries with JdbcTemplate and automatically map the results to your Data Objects. • Decide when to use SQL and when to use Hibernate, and learn how to combine them both in the same project.



Course 4: Security and DevOps

Learn about Git, version control, and best practices for authorization and authentication. Use Jenkins to build a CI/CD pipeline to deploy code to production.

Project: Implement Authorization for an eCommerce Application

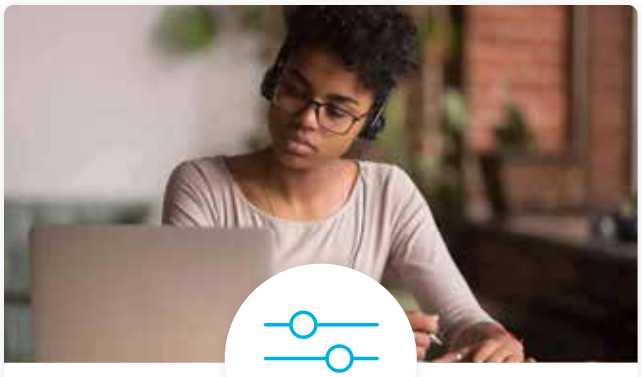
In this project, students will add authorization using Spring Security with OAuth and username/password combinations to an eCommerce web application created in Spring Boot. Proper security and hashing will need to be implemented to store this data as well. Students will identify the right metrics for an effective analytics environment and use either Splunk or ELK to analyze the metrics. Students will also automate the configuration and deployment of these systems and the application. Students will use Jenkins to integrate with their version control and deploy their application to AWS.

LESSON TITLE	LEARNING OUTCOMES
GIT	• Learn the basics of git such as branching, pull requests, and merging. • Describe what version control is and means.
AUTHORIZATION AND AUTHENTICATION	• Identify the need for security in modern day web applications. • Describe best practices for authorization and authentication. • Implement modern authorization and authentication technologies such as password hashing and JWT.
TESTING	• Learn and use testing frameworks such as junit. • Describe the concept of code coverage and its importance. • Implement negative testing as well as happy path testing.

Course 4: Security and DevOps, cont.

LESSON TITLE	LEARNING OUTCOMES
LOGGING AND ANALYTICS	• Identify important application metrics and log them. • Send logs to Splunk. • Create visualizations and dashboards in Splunk to display those metrics.
JENKINS AND CI/CD	• Describe and explain CI/CD. • Create a build pipeline using Jenkins. • Build a Docker Image. • Create a CI pipeline for a Docker Image. • Deploy Docker container in production.

Our Nanodegree Programs Include:



Pre-Assessments

Our in-depth workforce assessments identify your team's current level of knowledge in key areas. Results are used to generate custom learning paths designed to equip your workforce with the most applicable skill sets.

Dashboard & Progress Reports

Our interactive dashboard (enterprise management console) allows administrators to manage employee onboarding, track course progress, perform bulk enrollments and more.

Industry Validation & Reviews

Learners' progress and subject knowledge is tested and validated by industry experts and leaders from our advisory board. These in-depth reviews ensure your teams have achieved competency.

Real World Hands-on Projects

Through a series of rigorous, real-world projects, your employees learn and apply new techniques, analyze results, and produce actionable insights. Project portfolios demonstrate learners' growing proficiency and subject mastery.

Our Review Process

Real-life Reviewers for Real-life Projects

Real-world projects are at the core of our Nanodegree programs because hands-on learning is the best way to master a new skill. Receiving relevant feedback from an industry expert is a critical part of that learning process, and infinitely more useful than that from peers or automated grading systems. Udacity has a network of over 900 experienced project reviewers who provide personalized and timely feedback to help all learners succeed.

Vaibhav
UDACITY LEARNER

"I never felt overwhelmed while pursuing the Nanodegree program due to the valuable support of the reviewers, and now I am more confident in converting my ideas to reality."

now at

CODING VISIONS INFOTECH

All Learners Benefit From:

- Line-by-line feedback for coding projects
- Industry tips and best practices
- Advice on additional resources to research
- Unlimited submissions and feedback loops

How it Works

Real-world projects are integrated within the classroom experience, making for a seamless review process flow.

- Go through the lessons and work on the projects that follow
- Get help from your technical mentor, if needed
- Submit your project work
- Receive personalized feedback from the reviewer
- If the submission is not satisfactory, resubmit your project
- Continue submitting and receiving feedback from the reviewer until you successfully complete your project

About our Project Reviewers

Our expert project reviewers are evaluated against the highest standards and graded based on learners' progress. **Here's how they measure up to ensure your success.**

900+

Expert Project Reviewers

Are hand-picked to provide detailed feedback on your project submissions.

1.8M

Projects Reviewed

Our reviewers have extensive experience in guiding learners through their course projects.

3

Hours Average Turnaround

You can resubmit your project on the same day for additional feedback.

4.85 /5

Average Reviewer Rating

Our learners love the quality of the feedback they receive from our experienced reviewers.

UDACITY
FOR ENTERPRISE

2440 W El Camino Real, #101
Mountain View, CA 94040, USA - HQ

For more information visit: **www.udacity.com/enterprise**