

QA Automation - API Automation - All in one course

Course Summary:

- ⇒ In detailed, easy, step by step, real time, practical and well organized Course
- ⇒ Not required to have any prior programming knowledge, as all the prerequisites such as Java will be explained from basics
- ⇒ Training videos and PPT slides will be shared
- ⇒ Downloadable automation frameworks, programming code and other files will be provided
- ⇒ Real time Automation Projects will be explained (**Course Differentiator**)
- ⇒ Single stop course for all your API and Automation concepts learnings
- ⇒ This single course covers API Concepts, Web Services concepts, pre-requisites, basics, Apache HTTP Client Automation Framework, Rest Assured Automation Framework, SoapUI and many other tools related to API. (**Course Differentiator**)

Course Contents:

⇒ **Basics**

- Introduction to API Testing
- Client Server Architecture
 - Aim behind client server architecture
 - Characteristics of client and server
 - Types of architectures
- What is API ?
 - About URL
 - About URI
 - About end point
 - Parts of URL
 - Query parameters
 - Types of URLs

- About API
 - Introduction
 - Types of APIs
 - API Examples
 - Testing
 - What is API Testing?
 - Importance of API testing
 - Challenges of API Testing
 - Difference between API testing and unit testing
 - Best practices of API Testing
 - Types of testing one can do on API
- What are Web Services ?
 - Introduction
 - Components of web services
 - Working of web services
 - Architecture of web services
 - Types of web services
 - Testing
 - Difference between web service testing and Web API testing
 - Web services testing tools
 - Challenges of testing web services
 - Restful vs Restless protocols
- HTTP Protocol
 - HTTP vs HTTPS
 - About HTTP methods
 - Headers
 - Request body
 - payload formats
 - JSON
 - XML
 - WSDL

- Formdata
- Raw
- Etc..
- Authorization
- Status Codes
- Cookies
- URL encoding
- Characteristics

⇒ **Basic Java Concepts**

- Introduction
 - Installation of Java
 - Understanding Simple program
 - JVM, JDK & JRE architecture
 - Variables, Operators, Data types and Flow control statements with examples
 - Modifiers
 - Constructors
 - Arrays with examples
 - Packages
 - Overloading and Overriding
- OOPS concepts
 - Inheritance
 - Abstraction
 - Encapsulation
 - Polymorphism
- Strings
 - Specialties of Strings
 - Strings Immutability
 - StringBuffer and StringBuilder
- Exception handling
 - Specialty of Exceptions
 - Try and Catch

- throw and throws
- Custom exceptions
- Examples
- Collections
 - Types of Collections
 - Comparisons
 - List
 - Set
 - Iterators
- JDBC Connection
 - Creating connection
 - Creating the table
 - Inserting data into table
 - updating the data
- Threads
 - Initializing the thread
 - Purpose of threads
 - Multi Threading

⇒ **Soap and Rest API Concepts**

- Soap API
 - About Soap API
- REST API
 - About REST API
 - REST API Architecture
 - Resources
 - How they work?
- Soap versus REST API

⇒ Postman Tool

- Basics
 - Creating Collection
 - Importing & Exporting Collection
 - Setting up Environments
- Features
 - About Runner
 - Code
 - Environment & global variables
- handling CRUD requests
- Advanced concepts
 - Monitors
 - Request Chaining
 - Handling Runner
 - About Postman console
 - Reporting
- Exercises

⇒ Swagger UI

- About
- Features
- Types
- Postman Vs Swagger UI

⇒ Few browser tools

- Useful extensions
- Browser Console features
 - Viewing requests in Network tab
 - Response and headers
 - Looking towards API related errors

⇒ **Apache HTTPClient (with Real Time Automation projects)**

- Introduction
 - Configuration
 - Understanding syntax
- HttpClient Configuration
 - HTTP Request
 - Creating and sending HTTP GET
 - Creating and sending HTTP POST
 - Creating and sending HTTP PATCH
 - Creating and sending HTTP DELETE
 - Creating and sending HTTP PUT
 - HTTP Response
 - Validation of response body and code
 - Deserialization of responses in the form JSON, XML etc.
 - HTTP Entity
 - Exception Handling
 - Working with custom headers
 - Handling the secure endpoints
 - Terminology
 - Examples
- Multipart body handling
- Response Handling
- Parameterizing tests
- Handling Client authentications
 - Example
- HTTP Mime
 - Examples
- Automation Framework
 - Project structure

⇒ **Rest Assured (with Real Time projects)**

- Introduction
 - Setup
 - Understanding syntax
- Logging Test details
- Automation Framework
 - Project structure
- About Headers and Cookies
- Configuration
 - HTTP Request
 - Creating and sending HTTP GET
 - Creating and sending HTTP POST
 - Creating and sending HTTP PATCH
 - Creating and sending HTTP DELETE
 - Creating and sending HTTP PUT
 - HTTP Response
 - Validation of response body and code
 - Deserialization of responses in the from JSON, XML etc.
 - HTTP Entity
 - Exception Handling
 - Working with custom headers
 - Handling the secure endpoints
 - Terminology
 - Examples
- Handling Client authentications
 - Example
- MIME
 - Examples
- Multi part body handling
- Response Handling
- Parameterizing tests

⇒ **Real time Automation Examples**

- Types of Tests
 - API Tests
 - Acceptance Tests
 - Regression tests

⇒ **JMeter and Fiddler**

- Introduction
 - Installation
 - Steps to create sample request
- parameterizing the test data
- Assertions
- Metrics needs to be set in JMeter
- Components of JMeter
- Jmeter Functions
- Fiddler
 - Introduction
 - Installation
 - Steps to create sample request
 - Working of Fiddler
 - Inspectors
 - Components
 - Web Sessions

⇒ **Soap UI**

- Basics
 - Introduction
 - Installation and Configuration
 - Creating Project, Test suite and test cases
 - Assertions in SoapUI
 - Types of Assertions
 - Understanding Soap response
 - Database interaction
 - Restful Web Services
 - Performance and Load testing
- SoapUI + Scripting
 - Features
 - Setup and configuration
 - Sample Script explanation
 - Different HTTP requests forming
 - Response handling

⇒ **Assignments, Interview Questions, Resume, Exercises and Guidance**

- Assignments
- Interview Questions
- Exercises
- Resume Help
- Guidance

----- The End of the Contents -----