



Learning Objectives of CP-WST

"Knowledge with experience is power; certification is just a by-product"

Certified Professional Web Services Testing (CP-WST)

What is CP-WST?

Certified Professional Web Services Testing program is designed to train and uplift the knowledge of a test professional on Web Services, Web Services Testing techniques and test automation tools that will be utilized for web services testing and validation.

How is it useful?

Most of the organizations today deliver their applications utilizing web services. Web UI, Mobile and Desktop applications are now integrated to web services to do enterprise level functionality where they are able to deliver thin applications to the clients and move the complex functionality to server. Web services are the foundations for the creation of internet of things and Micro services-based enterprise level applications. When testing applications, it is utmost important to test web services or API level before moving up to the UI level testing and automation like Selenium. This program is a all in one program which will create the skill and expertise for any test professional to learn on web services testing and related tool stack. The course tends to look at tools and technologies which are demanded by the industry where the user will get hands on experience and certification to be a top notched web services test engineer.

Am I Eligible?

Any test professional who has the enthusiasm to learn web services testing and who has a basic knowledge in Java can join the program.

Duration

CP-WST is a hands-on training program which will run over 3 full days and completing with an online exam.



CP-WEB SERVICES TESTING

Learning Objectives of CP-WST:

1. Introduction to Web Services / API
 - 1.1. What is a web service?
 - 1.2. Web Services v/s API
 - 1.3. Examples of Web Services
 - 1.4. Advantages of Web Services
 - 1.5. Types of Web Services - SOAP and REST
 - 1.6. Web services architecture
 - 1.7. Web services characteristics
 - 1.8. REST v/s SOAP
 - 1.9. Creation of Demo REST Web Service
 - 1.10. What is API/Web Service Testing
 - 1.11. API Testing Strategy
 - 1.12. Web services components
 - 1.12.1. HTTP Request
 - 1.12.2. HTTP Methods (GET, PUT, POST, DELETE, ...)
 - 1.12.3. Request, Response
 - 1.12.4. Status codes (1xx, 2xx, 3xx, 4xx, 5xx)
2. Testing your web services with **Fiddler**
 - 2.1. Creating web services testing scenarios with fiddler
 - 2.2. Performing CRUD operations on APIs with GET, POST, PUT, DELETE
 - 2.3. Fiddler - Save the session to Archive
 - 2.4. Fiddler - Save the session to Archive
 - 2.5. Web services security
 - 2.6. Authentication and Authorization
 - 2.7. OAuth 1.0 and OAuth 2.0
3. Testing web services with Postman
 - 3.1. API Testing Tools usage
 - 3.2. Introduction to Postman
 - 3.3. Collections in Postman
 - 3.4. Start testing with Postman on GET/POST/DELETE/PUT request
 - 3.5. Environment Variables in Postman
 - 3.6. Test and Collection Runner in Postman
 - 3.7. Monitor Collections
 - 3.8. Assertions in Postman with Chai Assertion Library
 - 3.9. Different types of Asserts in Postman
4. Test Automation with Postman and Newman
 - 4.1. Installation and Pre-requisites.
 - 4.2. Running your test automation scripts in Newman
 - 4.3. Executing you Postman scenarios with Newman
 - 4.4. Integration of Newman with Jenkins
5. Test Automation with RestAssured
 - 5.1. Rest Assured Basics
 - 5.2. Configure Eclipse with Rest-Assured
 - 5.3. REST API Test
 - 5.4. Automate GET/POST/PUT and DELETE request



CP-WEB SERVICES TESTING

- 5.5. Validate Response Status
- 5.6. Validate Response Header
- 5.7. Read JSON Response Body
- 5.8. Data driven with JSON/XML and Excel in Rest Assured

6. Test Automation with UniRest
 - 6.1. Overview
 - 6.2. Maven Setup
 - 6.3. Request Methods
 - 6.4. Response Methods
 - 6.5. Passing Query Params
 - 6.6. Requests with Body
 - 6.7. Handling file Uploads request

7. Test Automation with Karate
 - 7.1. Overview, Installation and Configuration of Karate
 - 7.2. Karate and BDD
 - 7.3. Creating tests in Karate
 - 7.4. Testing the Status Code
 - 7.5. Testing the Response
 - 7.6. Validating Response Values with Markers
 - 7.7. Running Tests

8. Micro services testing
 - 8.1. Introduction to Micro services architecture
 - 8.2. Types of Micro services testing