API Testing Workshop-oAPI

- Get a foundational understanding of API Testing
- Gain experience and practice with hands-on exercises
- Learn API Testing best practices
- Sample common tools and techniques used in API Testing

Join us for this workshop-style class where we will examine and practice the benefits of API testing using commercial and open-source software. Students will leave with a better appreciation of the value of API testing, and how it can be applied to their work.

An application programming interface (API) acts as the plumbing for a system. For web applications, developers use a web API to communicate between the client and server. This API is often nothing more than regular text that follows a known pattern that both the client and server understand. Accessing this layer allows us to test differently. In the UI, certain invalid inputs are prevented, but the API will allow any possible ordering of text, which opens up a lot of possibilities for finding defects.

Why is testing at the API level valuable? For one, it lets us follow data-driven testing, so we can be faster and more robust than testing the UI. Also, it can be used to test the security; for example, does your server implicitly trust what the client is sending? Finally, it is an excellent method to confirm that when sending invalid information, the API should respond in a helpful manner. A large portion of the developer's work is debugging and maintenance, and API's that respond with helpful messages go a long way towards that goal.

Who Should Attend

This course is ideal for agile and DevOps teams who need to define and validate specific API use cases and understand the boundaries and expectations of the system. While some of the course requires writing code, the majority can be accomplished with a simple text editor, web browser, and access to the API we're testing.

Course Outline

An introduction to API testing

Reasons for API testing

What software uses APIs

API formats

- RESTful
- Soap
- JSON
- HTML
- Plain text
- YAML
- GRPC
- protobuf
- historical
- Corba
- RPC

API testing techniques

Focus on negative cases

Clarity of error messages

Security concerns

Performance

Avoiding the network

Authorization testing

Query string

Body

Headers

Data-driven testing

Fuzzing

Required inputs

API Contracts

Character encoding

Software testing with APIs
General testing mindset
Agile test pyramid
End-to-end testing
TDD
Functional and Non-Functional
Manual exploratory testing
Min-size/Max-size tests
External API's
Debugging
Documenting
Caching

API testing tools
General programming languages
Postman
SoapUI
Specialized tools for external API's
Swagger
Network sniffers