Basics of Configuration Management

In this lesson we will walk you through the basics of Configuration Management.

WE'LL COVER THE FOLLOWING

- What is configuration management?
- Types of configuration files
- Use of configuration in the framework

What is configuration management?

One of the key features of a framework is ***Configuration management***. A framework can contain so many features, but how easy is it to customize it and change the behavior of the already present features. Therefore, using configuration-driven parameters is an essential feature to have in the framework.

Types of configuration files

The configuration can be picked up from files like `.properties`, `.ini`, `.xml`, `.json`, `.yml`, or `.toml`, or it can be picked up from the database or any other external sources.

Use of configuration in the framework

The framework should be able to read the configurable parameters from any of the sources and be able to change the behaviour of the features provided by the framework.

For example, let's say the framework has utilities to connect to the database. One of the projects using the framework uses database connections and one does not. If the framework always connects to the database irrespective of whether it is used or not, then it's not a good way to utilize the resources. To avoid such a thing from happening, if we have a configuration that the framework reads to turn on/off database connection initialization like

`database.enabled = true` or `database.enabled = false`, we can control the behavior at will with no hard-coding.

This will allow the framework to work independently and free from hard-coding, thus making it more extensible, robust, configurable and easily pluggable with any CI/CD tools.

Now that you’re familiar with Configuration Management, it’s time to learn about building Configuration Hierarchy and Overriding.