

Study with Me

(Rest Assured Module) -1

Revisit of ConfigManager and Dynamic AuthToken

1. How do you handle the Configuration in your Automation Framework?

For Managing the multiple environments information in our project we have created multiple properties file like

1. config.dev.properties
2. config.qa.properties
3. config.uat.properties

These properties files are stored in src/test/resources folder of our Project.

I have created a Utility Class called ConfigManager whose job is to load the properties into the memory during the execution of the program.

```
public class ConfigManager {

    private static Properties prop = new Properties();
    private static String path = "config/config.properties";
    private static String env;

    private ConfigManager() {
        // Private Constructor!!!
    }

    static {

        env = System.getProperty("env", "qa");
        env = env.toLowerCase().trim();
        System.out.println("Running Tests in Env" + env);
        switch (env) {
            case "dev" -> path = "config/config.dev.properties";

            case "qa" -> path = "config/config.qa.properties";

            case "uat" -> path = "config/config.uat.properties";

            default -> path = "config/config.qa.properties";
        }

        InputStream input = Thread.currentThread().getContextClassLoader().getResourceAsStream(path);

        if (input == null) {
            throw new RuntimeException("Cannot Find the file at the path" + path);
        }

        try {
            prop.load(input);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }


    public static String getProperty(String key) {

        return prop.getProperty(key);
    }
}
```

2. Why the ConfigManager Class has a private Constructor?

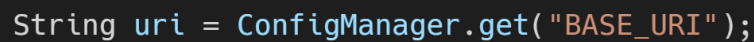
Because ConfigManager is a **utility class** (you never want to create multiple objects).

Making the constructor private prevents:



```
ConfigManager cm = new ConfigManager(); // Not allowed
```

Instead, everything is accessed statically:



```
String uri = ConfigManager.get("BASE_URI");
```

3. What is the purpose of the Static Block in Java?

- Executes **once** at class-load time.
- Loads the properties file into memory only once.
- Saves I/O and ensures fail-fast if file not found.

```
static {  
  
    env = System.getProperty("env", "qa");  
    env = env.toLowerCase().trim();  
    System.out.println("Running Tests in Env" + env);  
    switch (env) {  
        case "dev" -> path = "config/config.dev.properties";  
  
        case "qa" -> path = "config/config.qa.properties";  
  
        case "uat" -> path = "config/config.uat.properties";  
  
        default -> path = "config/config.qa.properties";  
    }  
  
    InputStream input = Thread.currentThread().getContextClassLoader().getResourceAsStream(path);  
  
    if (input == null) {  
        throw new RuntimeException("Cannot Find the file at the path" + path);  
    }  
  
    try {  
        prop.load(input);  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

4. How do you pass the value from the terminal to your automation framework?

How to pass Run time Arguments to your Automation Framework?

- Because we are using Maven Project and our code is majorly kept in src/test/java.
- We use the mvn test life cycle to run the tests from the terminal
- To pass the value from the terminal we use the -D attribute from maven
 - -Denv=qa where env is the name of the variable and qa is the value
- To access this value in the code we use the **System.getProperty("env");**

Correct way to use System.getProperty?

```
String env = System.getProperty("env", "local"); // default = local
```

If nothing passed, it uses "local".

This avoids NullPointerException and makes the framework flexible.

5. What is the purpose of :

Thread.currentThread().getContextClassLoader().getResourceAsStream() ?

- Loads a resource file (like config.properties) from classpath (src/test/resources).
- Returns an InputStream to read it, independent of OS path.
- PS: When you use getResourceAsStream() you don't need to give the explicit path of the File in your code

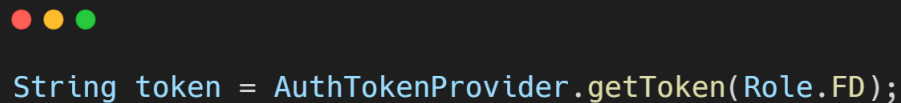
```
InputStream input = Thread
    .currentThread( )
    .getContextClassLoader( )
    .getResourceAsStream( "config/config.properties" );
```

6. What design pattern is used in ConfigManager?

- **Singleton pattern (Utility Singleton).**
- Ensures a single copy of properties is loaded.
- Static methods + static block = Singleton utility.

7. How do you pass Authorization Token in your tests?

Instead of passing it from one tests to another test which becomes a dependency and will affect parallel execution I have created a Dynamic Authorization Token Class which generates the token for a specified role.



```
String token = AuthTokenProvider.getToken(Role.FD);
```

8. What is an enum in Java, and why use it for roles?

Enum = set of **fixed constants**.

Safer than strings ("fd" vs "FD" typo risk).

In our Project all the predefined roles are stored in Role ENUM and we use it in AuthTokenProvider Class to decides which credentials to pick for Model/POJO object creation.

9. How do you run specific Maven tests with a specific environment?



```
mvn test -Dtest=LoginAPITest -Denv=qa
```