**MODUL 5**

**RESULT AND REPORTING**

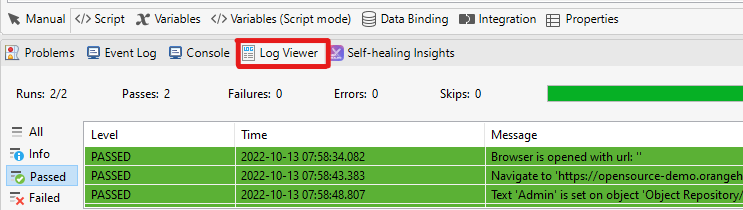
1. **TEST EXECUTION REPORTS**

Didalam katalon studio terdapat dua fitur yang menyajikan hasil eksekusi dari setiap test yang dijalankan.

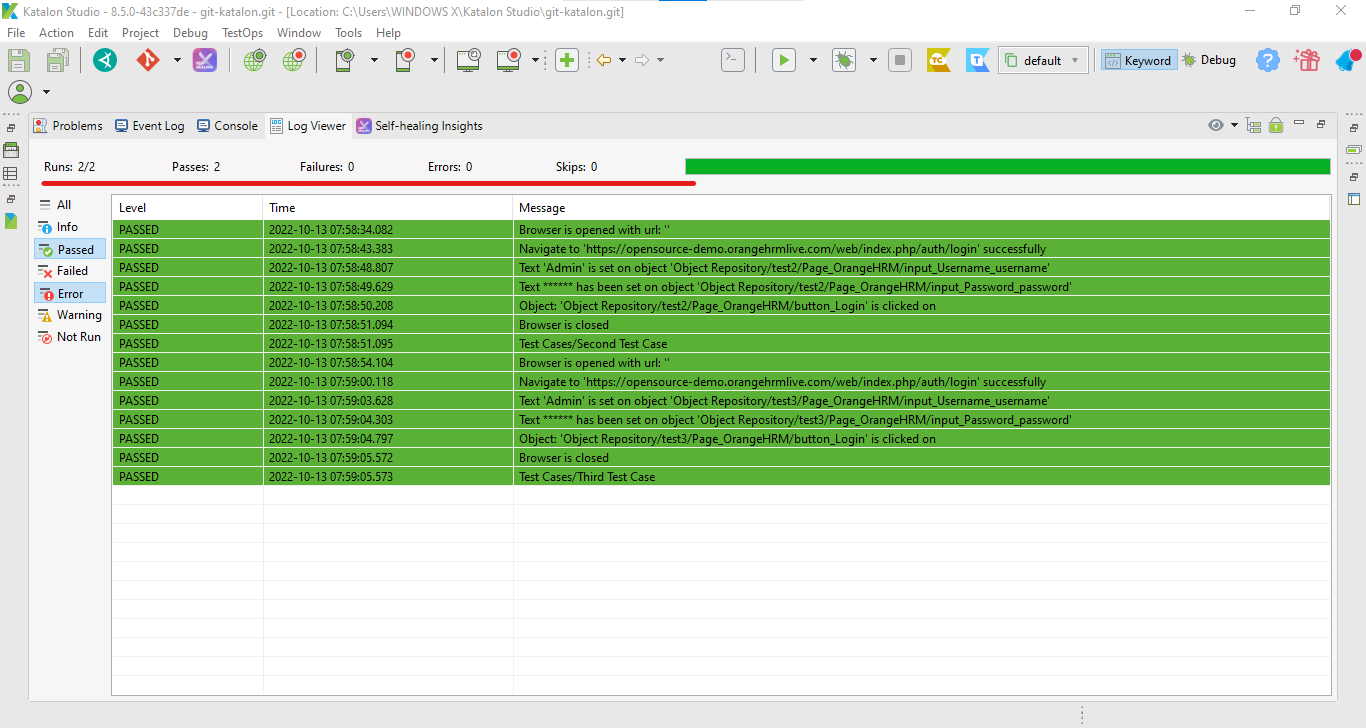
* Log : untuk menyajikan hasil eksekusi dari test case, test suite, dan test suite collection
* Report : untuk menyajikan hasil eksekusi dari test suite dan test suite collection

1. **Log Viewer**

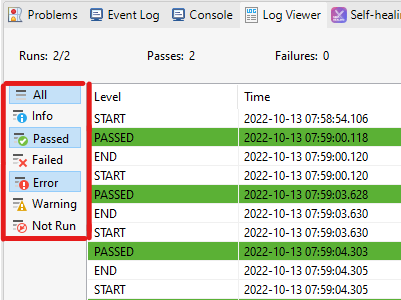
Kita dapat menggunakan tools log viewer untuk melihat log dari setiap test case.



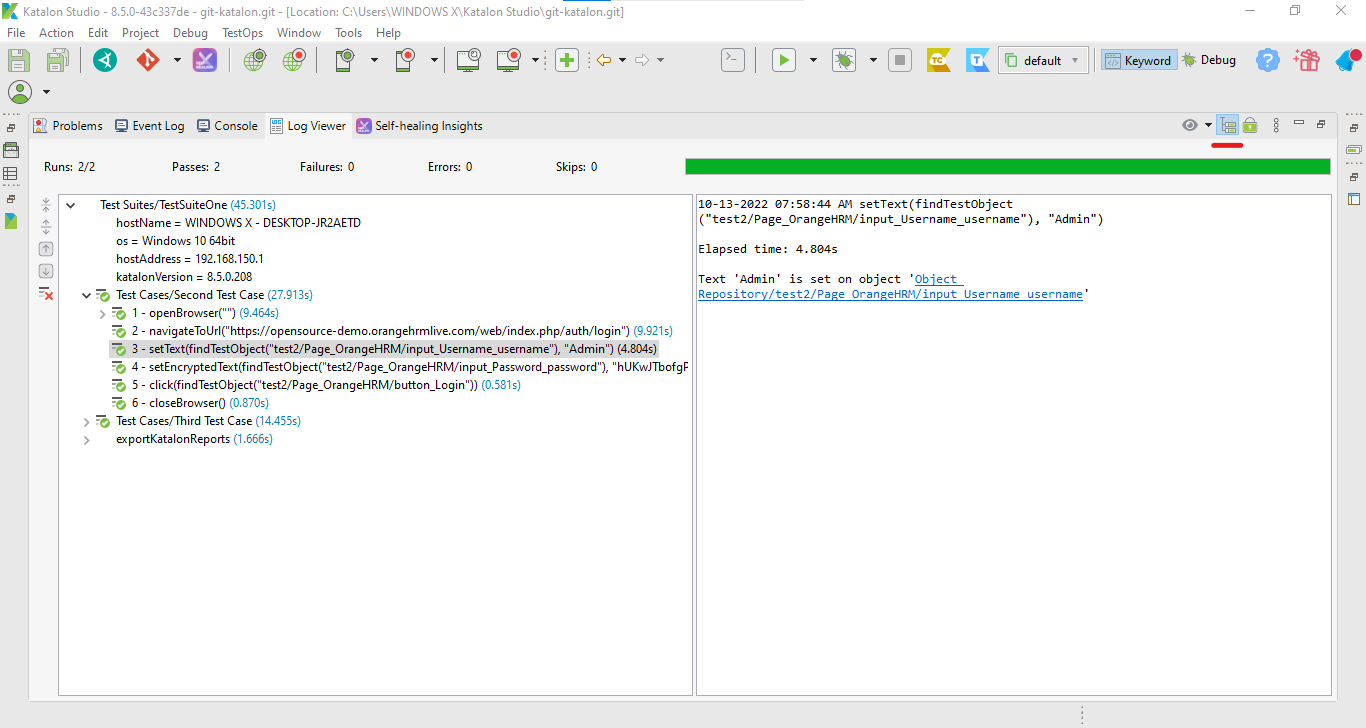
Dalam jendela log viewer kita dapat melihat tes yang dijalankan, tes yang lulus uji, kesalahan, error, dan test yang dilewati.



Jika ingin memfilter log berdasarkan hasil kita dapat menggunakan menu pada jendela log sebelah kiri



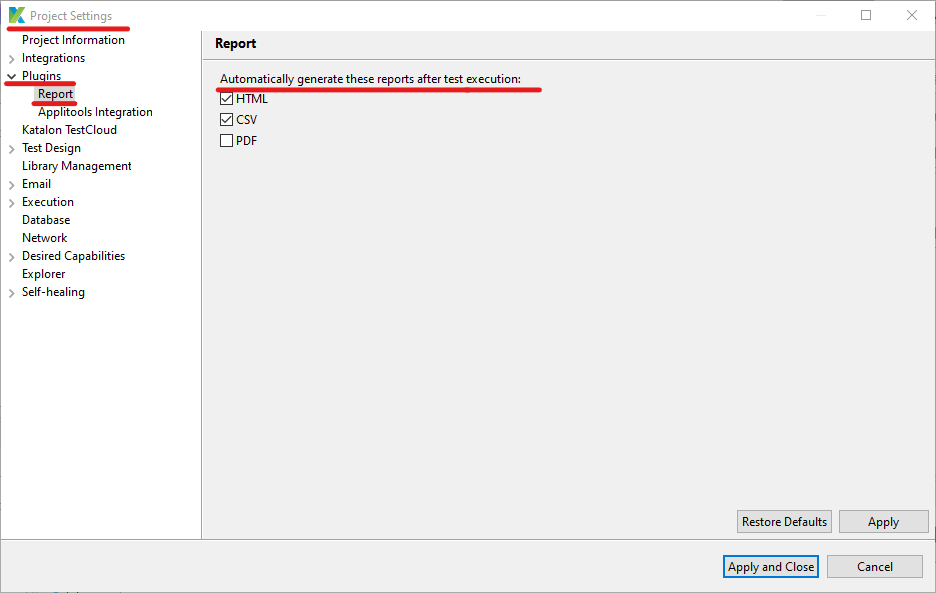
Katalon juga menyediakan tampilan hirarki yang didalamnya terdapat jendela log dan jendela detail dari setiap tahapan pengujian



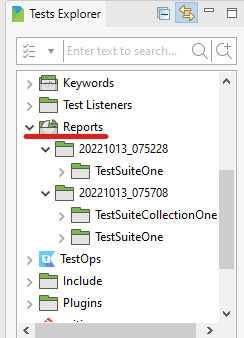
1. **Report**

Dalam katalon studio kita dapat mengatur format report yang digunakan untuk menyajikan report melalui menu project > setting > plugins > report

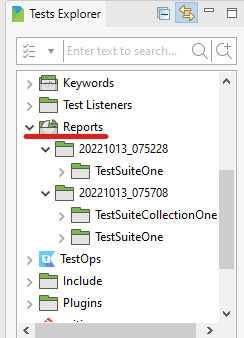
Ada tiga jenis format report yang disajikan yaitu html, csv, dan pdf.



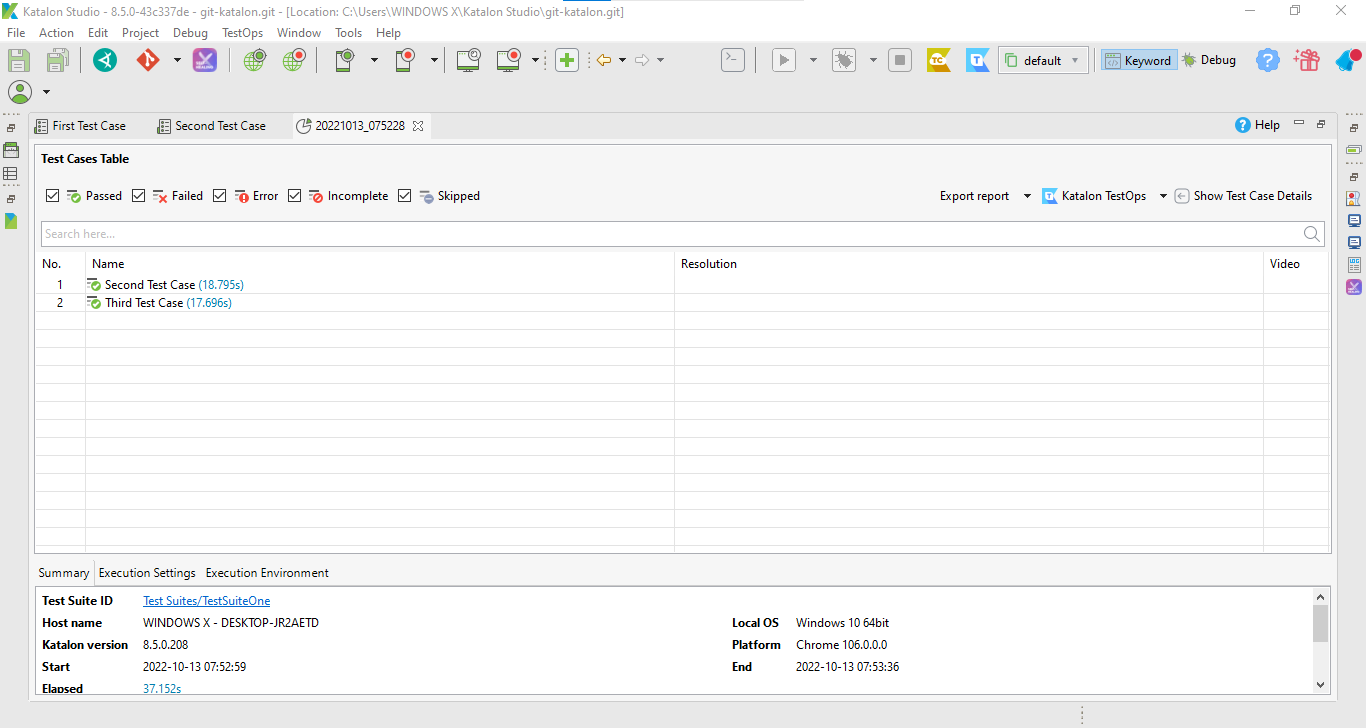
Untuk melihat report kita dapat menuju jendela explorer dan memilih menu report



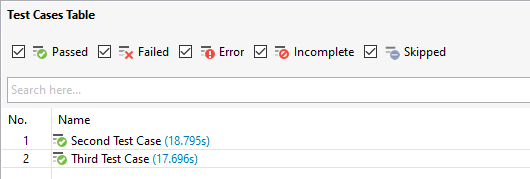
Report yang disajikan sesuai dengan jumlah eksekusi yang dilakukan terhadap test suite ataupun test suite collection. Penamaan report secara default diawali oleh tahun, bulan, dan tanggal atau sesuai dengan waktu eksekusi



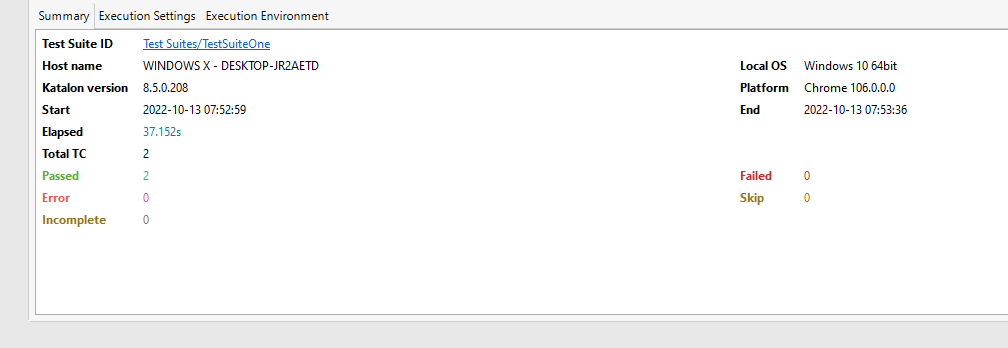
Berikut ini adalah tampilan jendela report secara keseluruhan



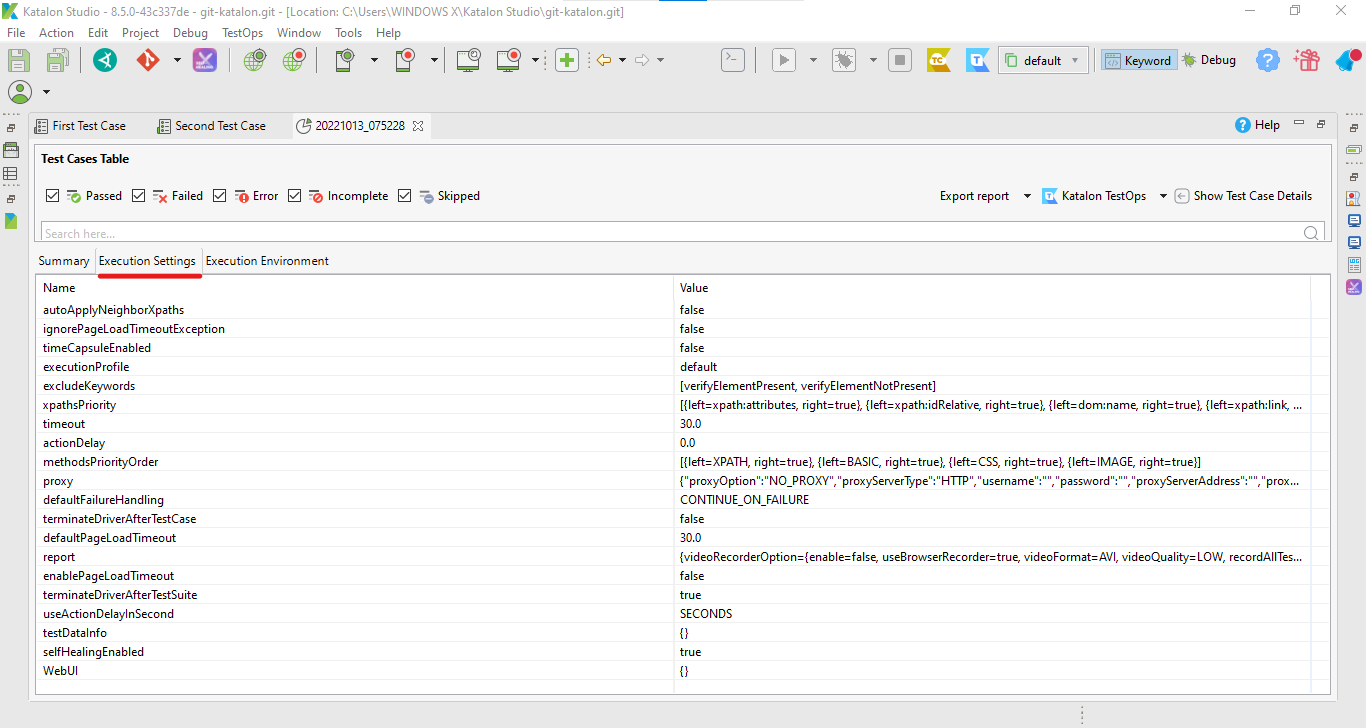
Terdapat test case table yang akan menampilkan test case yang diuji beserta dengan status kelulusan, kita juga dapat memfilter tampilan dengan menggunakan checkbox di bagian atas jendela test case table.



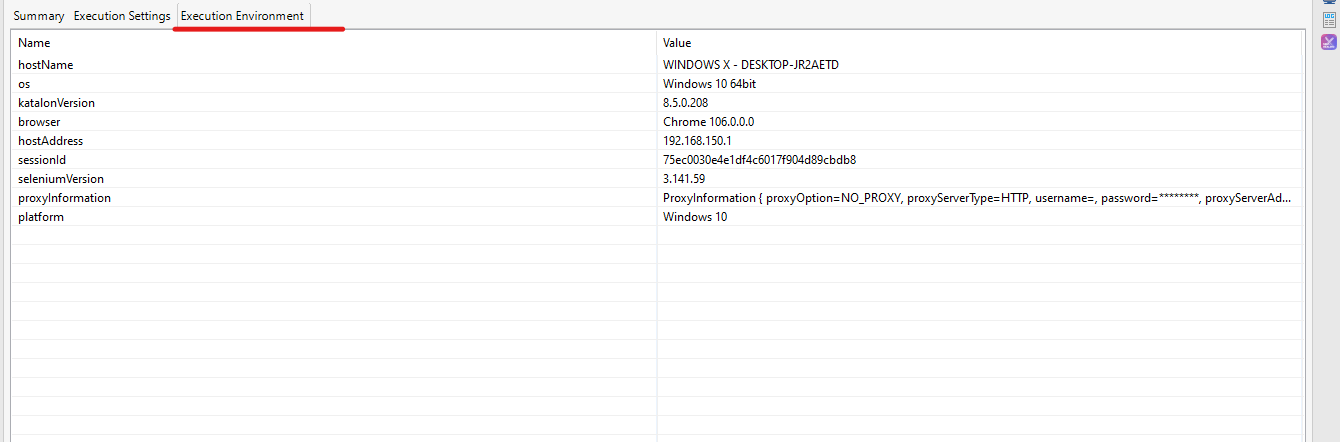
Di bagian bawah jendela report juga terdapat ringkasan pengujian yang terletak pada tab summary



Kita juga dapat melihat seluruh setting saat eksekusi melalui tab Execution setting



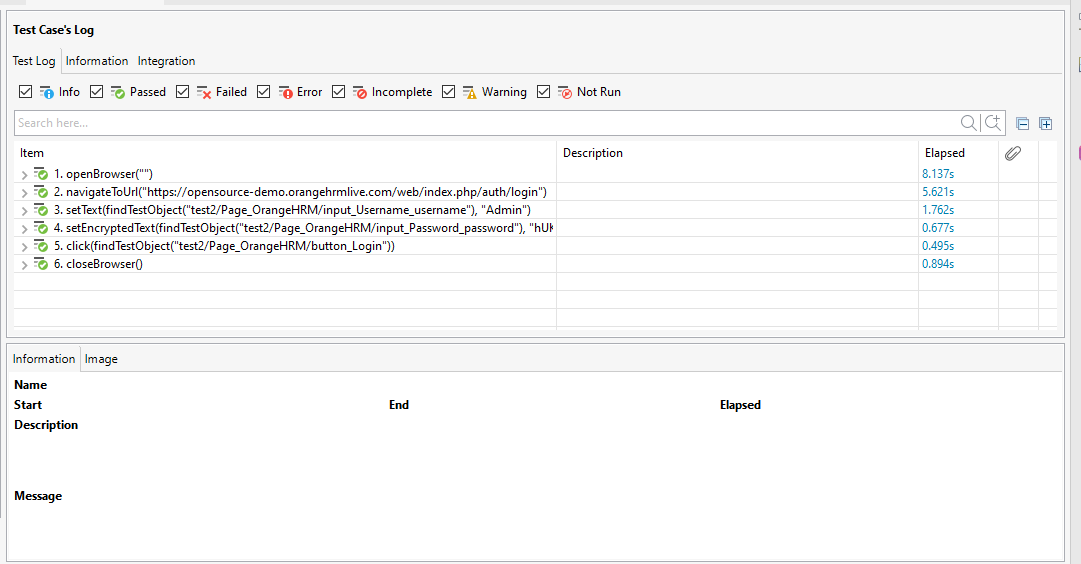
Kita juga dapat melihat lingkungan pengujian melalui tab Execution Environtment



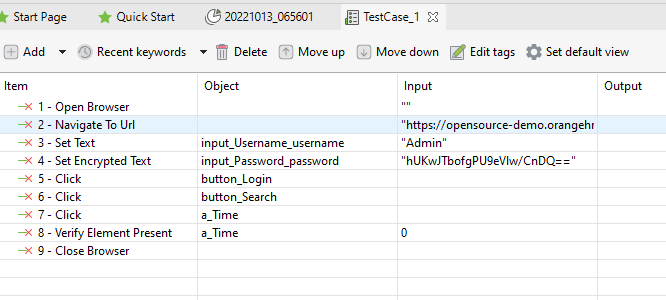
Untuk melihat tampilan detail dari pengujian kita dapat menggunakan menu test case detail

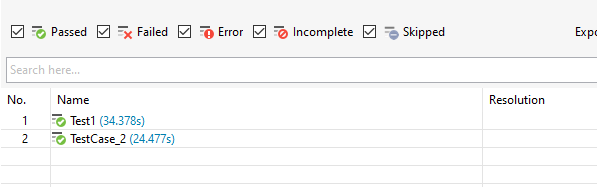


Berikut adalah tampilan dari detail setiap pengujian, yang didalamnya terdapat tahapan pengujian, deskripsi, dan waktu pengujian.

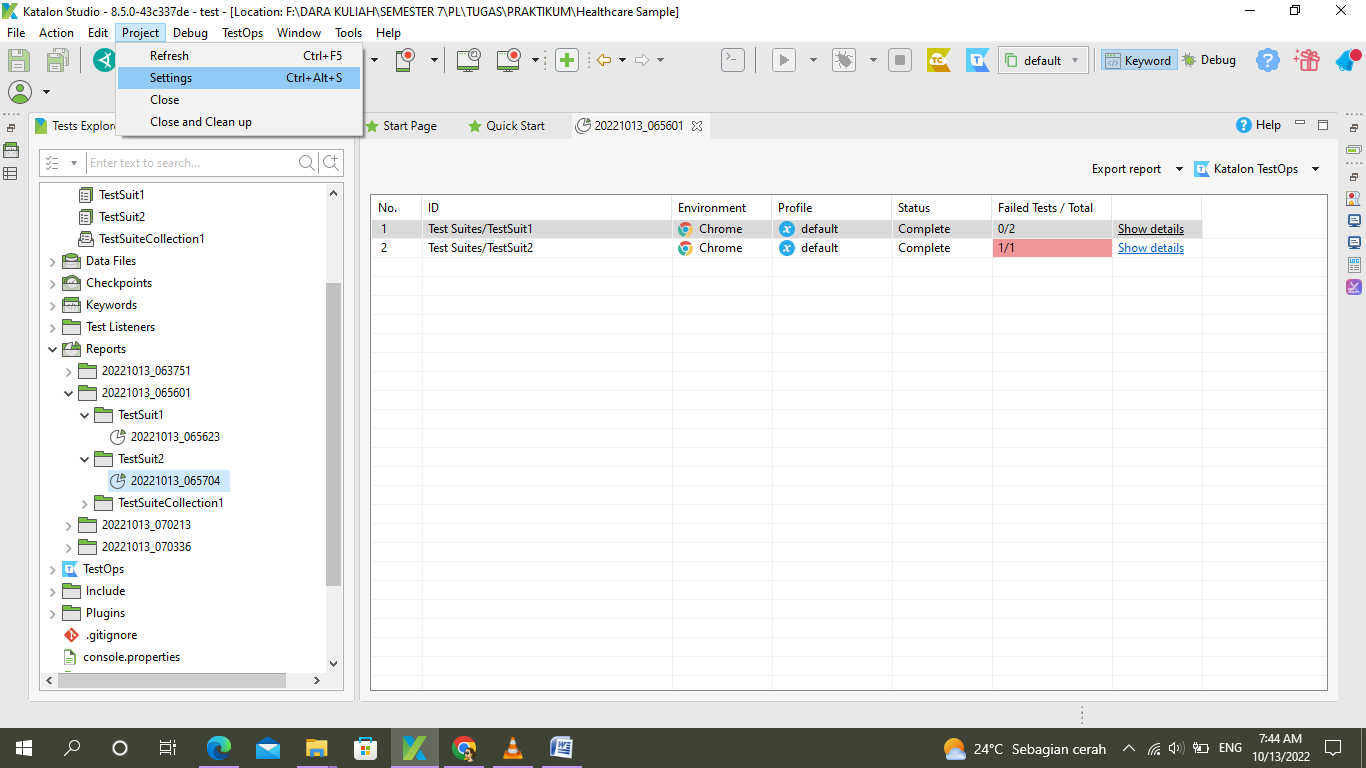


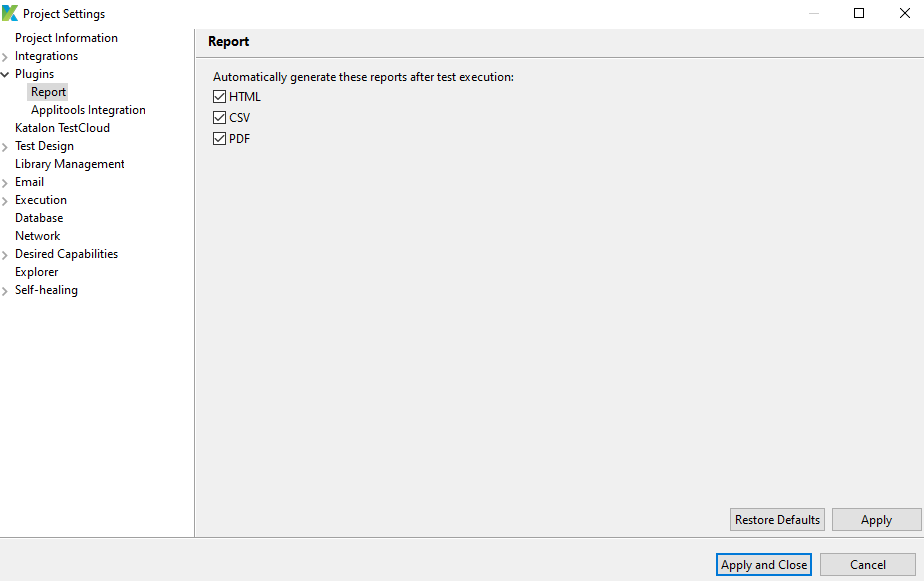
1. **BASIC REPORTS PLUGINS**
2. Pada tahapan sebelumnya kita sudah banyak membuat test case, test suit, maupun test suit collection pada katalon. Salah satunya test case seperti yang terlihat pada gambar di bawah ini.



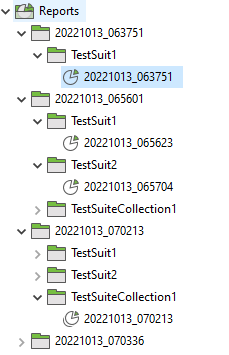


1. Sebelum kita mulai mengeksport laporan hasil pengujian dalam format HTML, CSV, atay PDF, terlebih dahulu kita melakukan setting pada menu projek, dimana untuk semua plugins report ceklis untuk semua format pilihan yang ada.

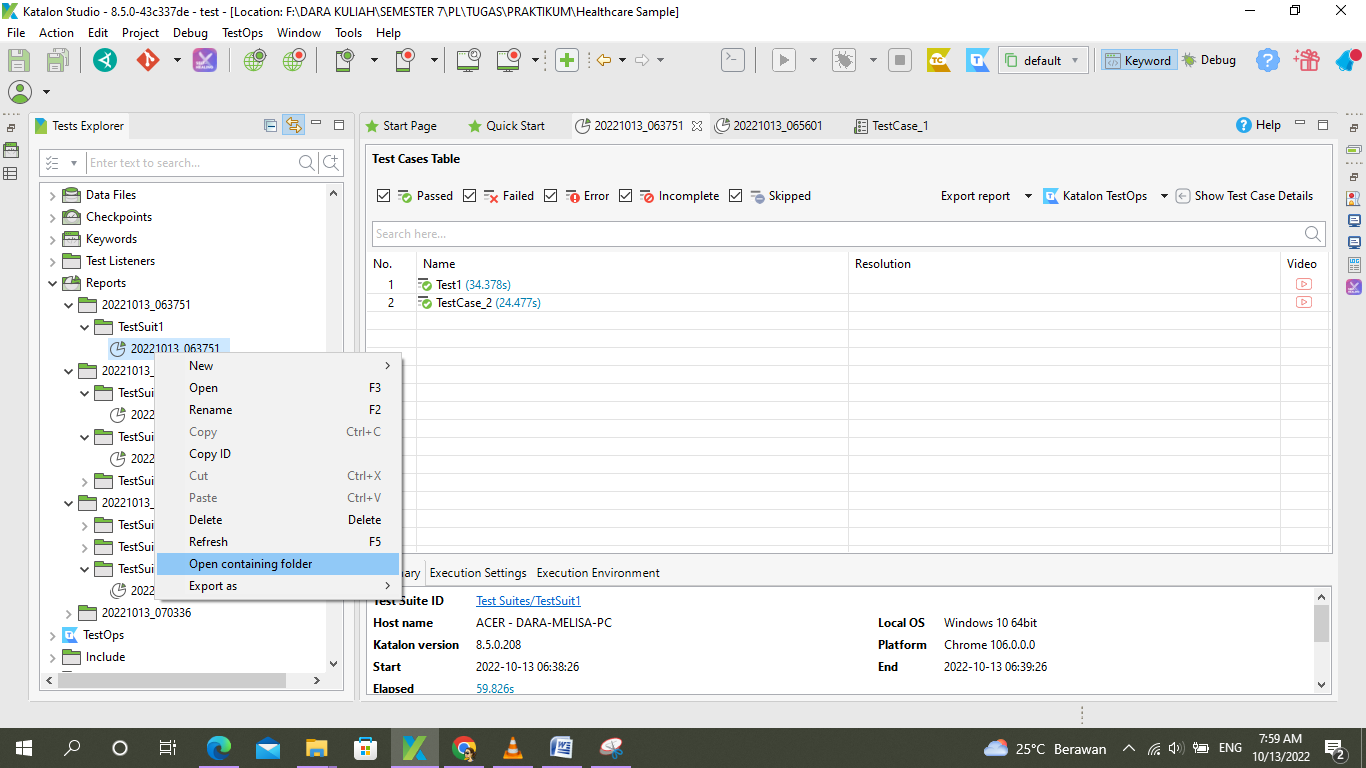




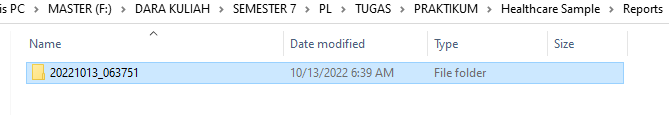
1. Setelah itu pada menu reports, dapat kita lihat berbagai pengujian yang telah kita buat. Untuk kasus ini kita melakukan export terhadap pengujian test case yang sudah dilakukan penambahan pada test suit1.



1. Untuk melihat hasil report dalam format HTML, klik kanan pada report kemudian pilih **open containing folder**.



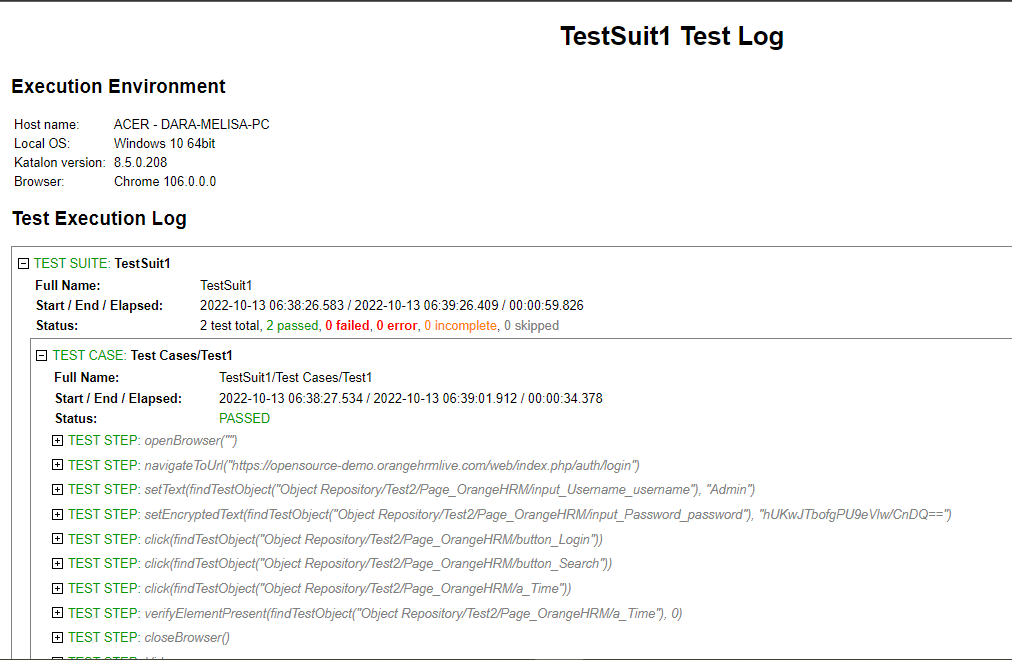
1. Selanjutnya user akan di arahkan ke halaman lokasi tempat penyimpanan folder.

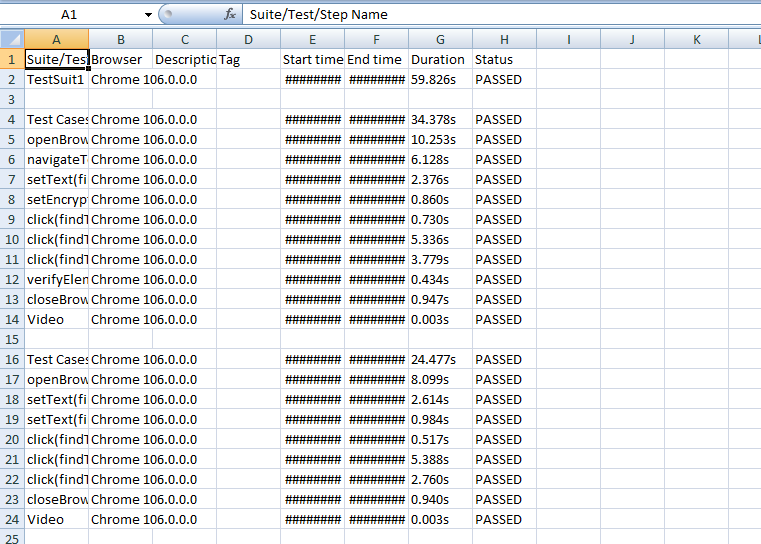


1. Ketika folder dibuka, user dapat melihat hasil report pengujian sudah ada dalam format HTML dan CSV, ini dapat terjadi karena sebelumnya kita telah melakukan setting terhadap report plugin untuk mengaktifkan ceklis terhadap berbagai pilihan format plugin yang tersedia.

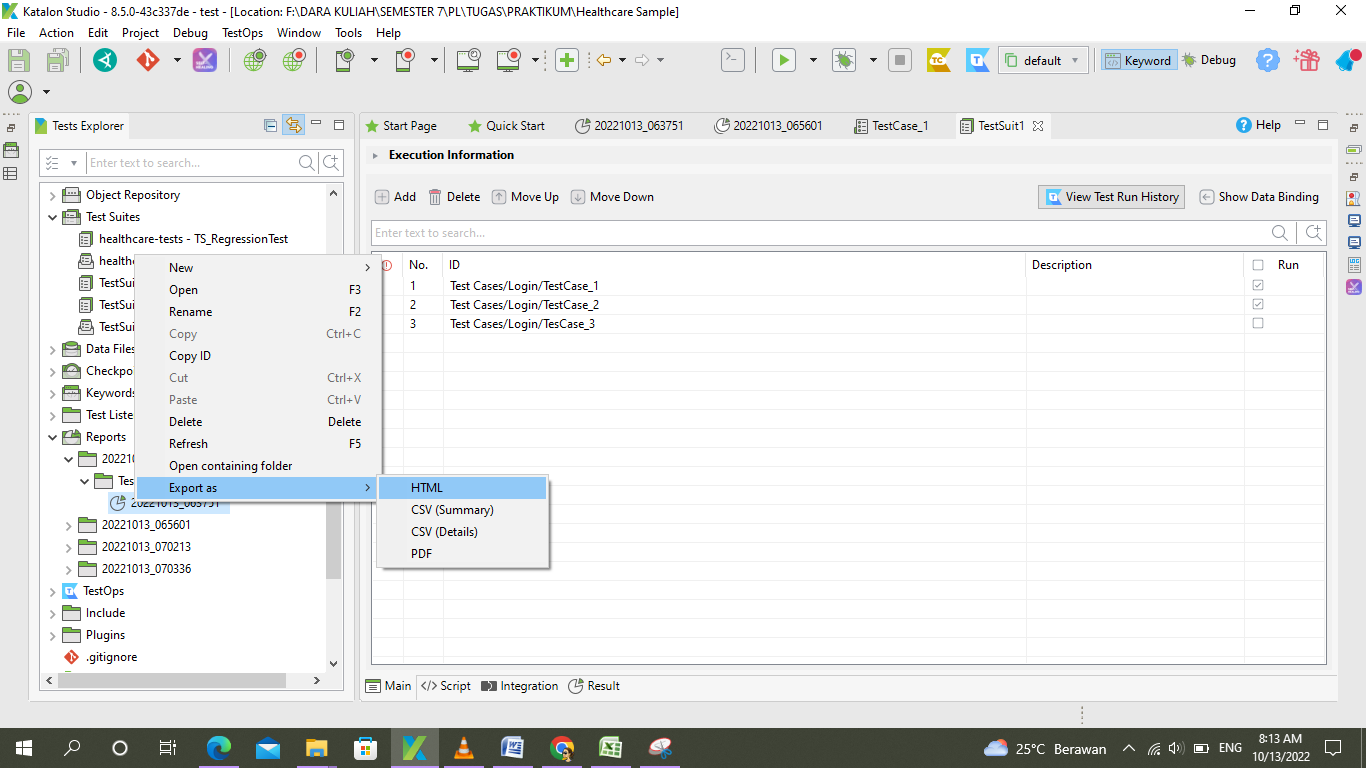


1. Berikut report hasil pengujian dalam format HTML dan CSV.

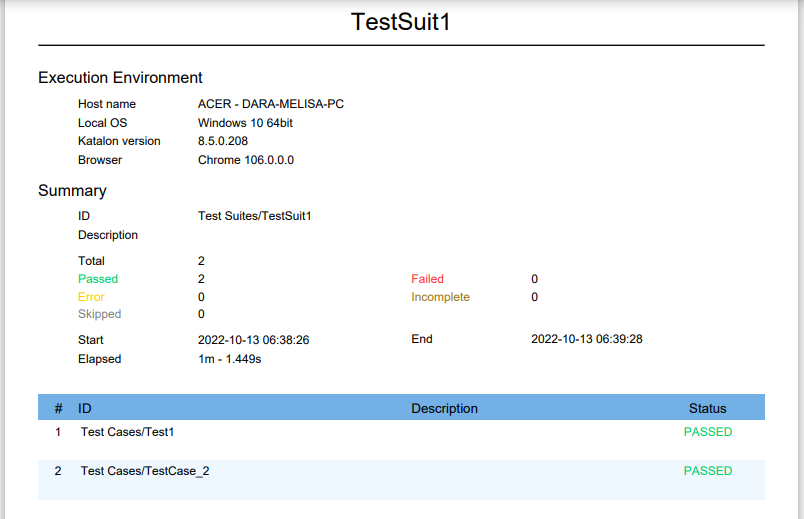




1. Untuk melihat report hasil pengujian dalam format PDF, anda bisa klik kanan pada report, kemudian pilih **exports as**, lalu pilih format pdf.



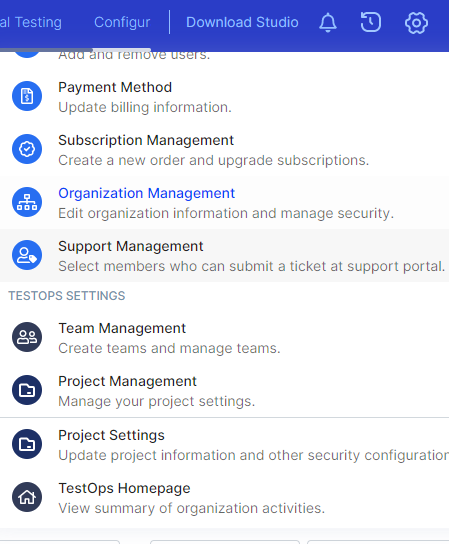
Pilih tempat penyimpanan file, kemudian save. Berikut reprt hasil pengujian dalam format PDF.



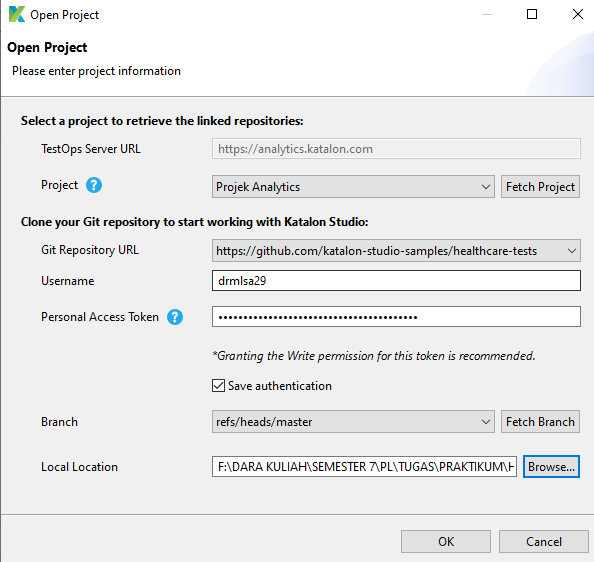
1. **HOW TO EMAIL RESULTS**
2. **HOW TO USE KATALON ANALYTICS STEP BY STEP**

Pada modul ini kita akan mempelajari bagaimana cara menggunakan katalon analytics. Katalon analytic merupakan suatu proses yang berfungsi untuk melihat, menyimpan, dan menganalisis hasil atau laporan dari pengujian yang dilakukan pada katalon. Berikut ini merupakan tahapan menggunakan katalon analytics.

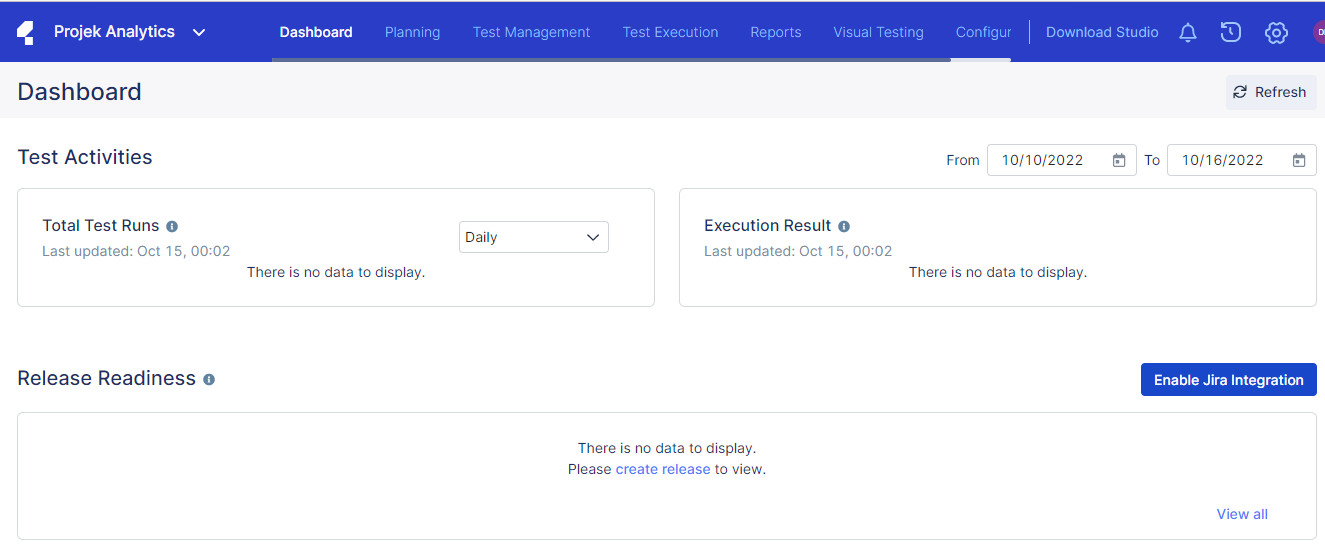
1. Langkah pertama yang kita lakukan adalah, kita bisa membuat projek baru untuk katalon. Projek baru dapat dibuat dengan masuk ke web katalon, kemudian klik setting, lalu pilih new projek, dan projek baru pun telah dibuat.



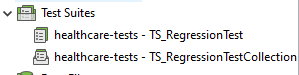
1. Selanjutnya, buka aplikasi katalon kemudian klik file lalu open projek. Masukkan username dan token dari akun github anda, lalu pada bagian projek pilih nama projek yang sudah dibuat pada web katalon analytics sebelumnya, pada kasus ini saya membuat projek dengan nama projek analytics. Pada proses ini kita sudah berhasil menghubungkan langsung test projek yang kita lakukan di aplikasi katalon ke web katalon analytics.



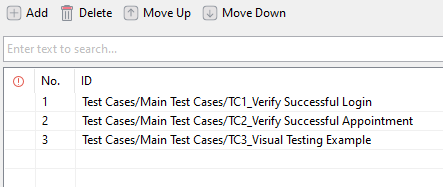
1. Jika kita lihat pada halaman test activities pada web katalon analytics, belum ada test activities apapun yang terekam karena memang belum ada pengujian yang dilakukan.



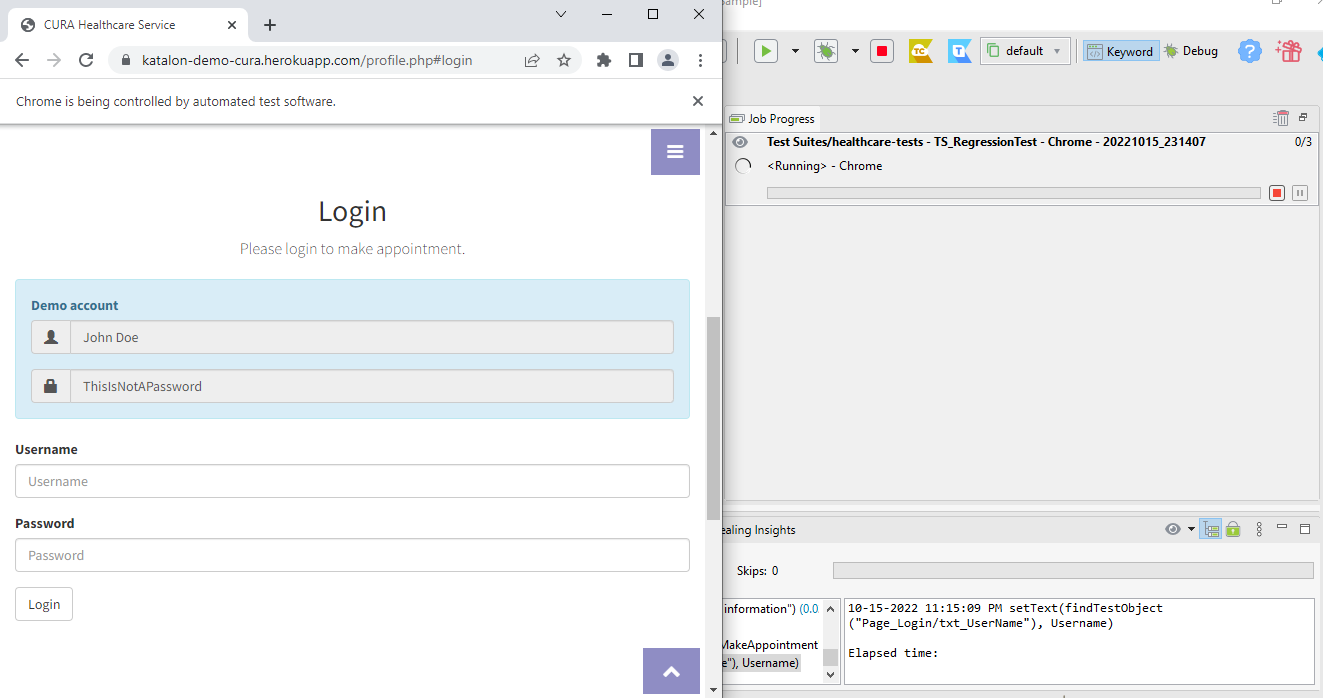
1. Selanjutnya kita akan menjalankan test suit sebagai bentuk percobaannya. Test suit ini merupakan test case yang sudah disedikan oleh katalon sebagi sampel pengujian. Untuk kasus ini saya memilih test case TR\_RegressioonTest.

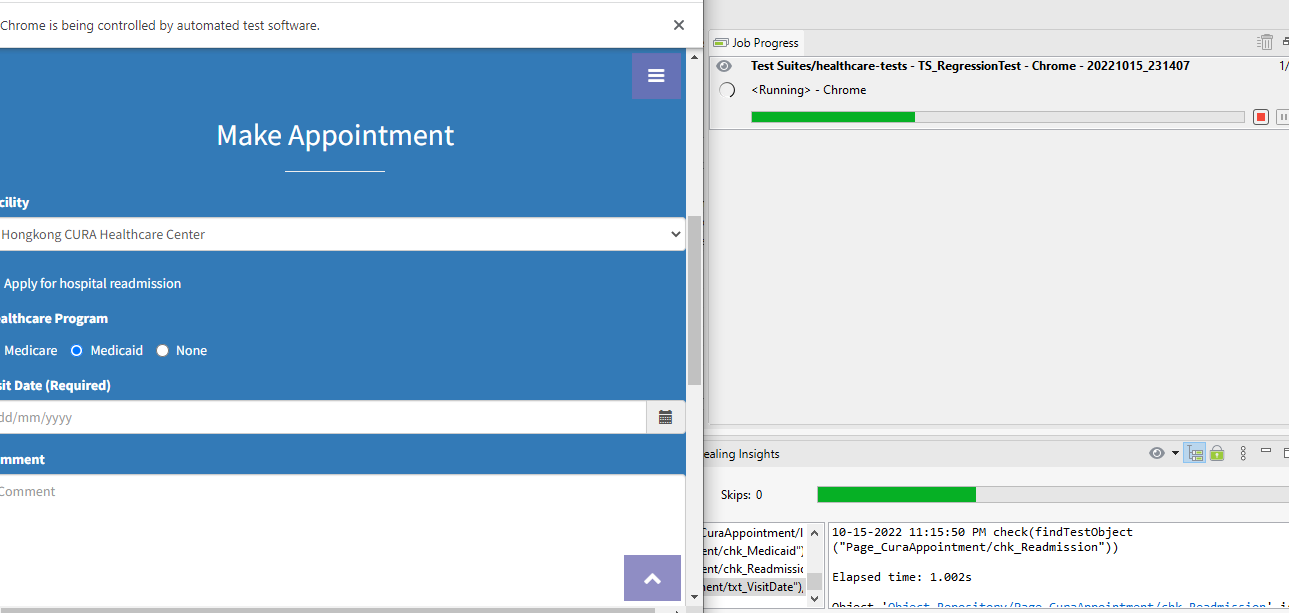


Seperti yang dapat dilihat pada gambar di bawah ini, pada test suit tersebut sudah ditambahkan beberapa test case.

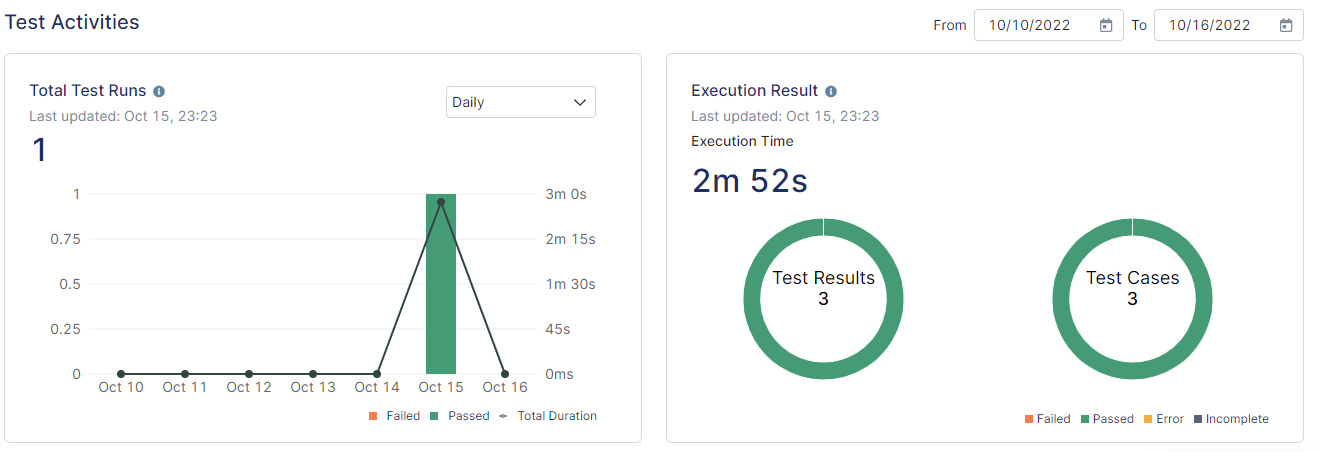


1. Jalankan test suit dengan bwowser pilihan anda, dan tunggu sampai proses eksekusi selesai. Berikut ini gambar tampilan test suit yang sedang dieksekusi.





1. Setelah proses eksekusi selesai, kita bisa melihat hasil report pengujian pada web alytics katalon. Terlihat pada test activities sudah ada 1 proses pengujian yang sudah silakukan.



Dapat dilihat seperti pada gambar di atas, hasil detail report menunjukkan 1 passed, dan 0 error untuk pengujian yang dilakukan.

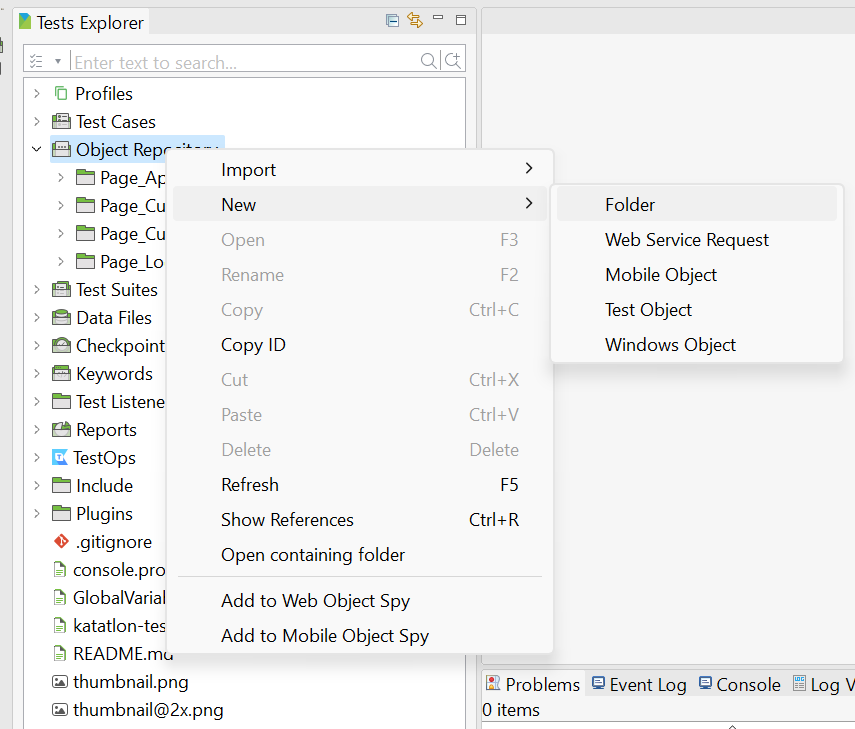
**MODUL 9**

**API TESTING**

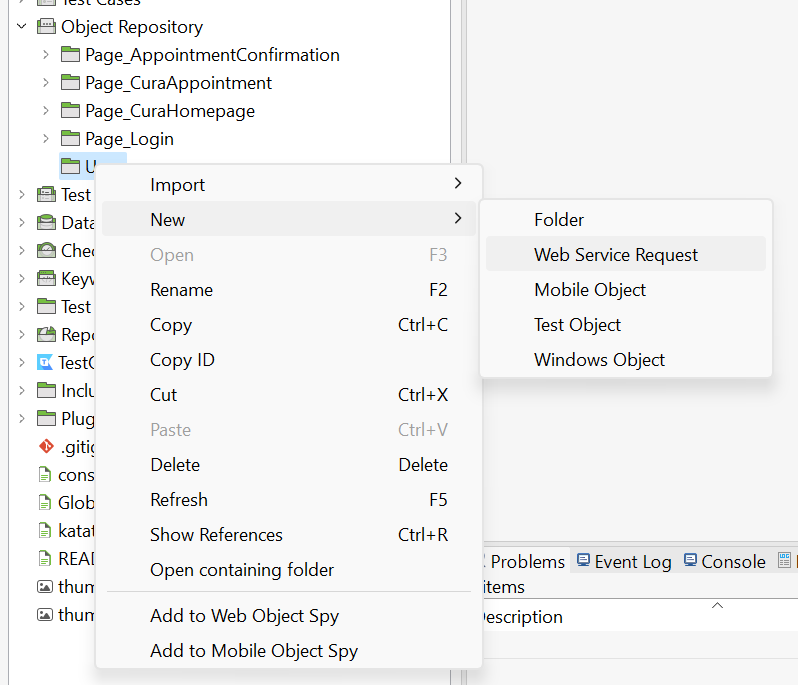
* 1. **HOW TO DO API TESTING WITH KATALON**

Katalon juga dapat digunakan sebagai aplikasi untuk melakukan pengujian terhadap API. Berikut langkah-langkah yang dapat dilakukan dalam menguji API pada katalon.

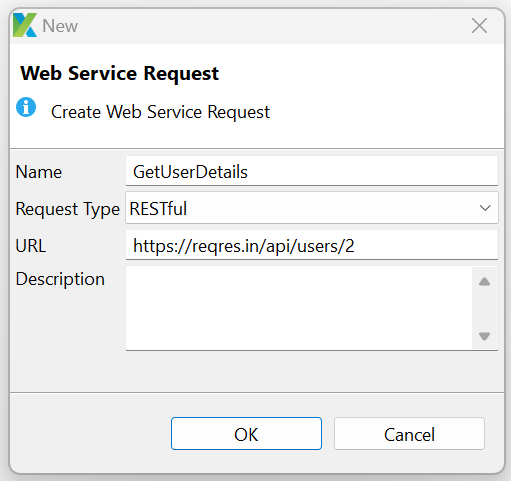
1. Buka project pada katalon, pada bagian Test Explorer klik kanan pada folder Object Respository untuk membuat folder baru.

****

1. Setelah folder berhasil dibuat klik kanan, pilih web service Request

****

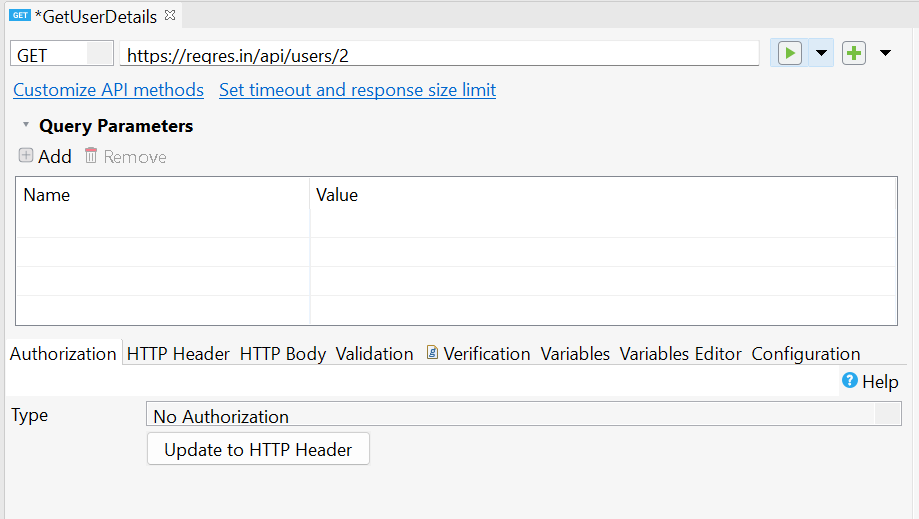
1. Kemudian isikan name untuk file yang akan dibuat, pilih RESTful sebagai request type, dan masukkan URL API, klik Ok.



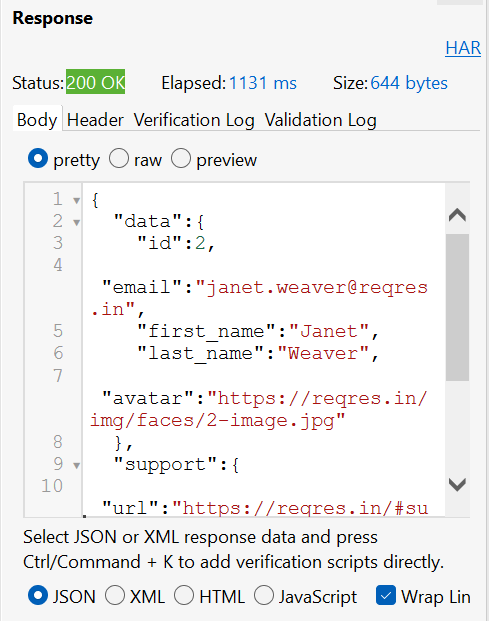
1. URL API tersebut jika dibuka melalui web akan tampil seperti berikut



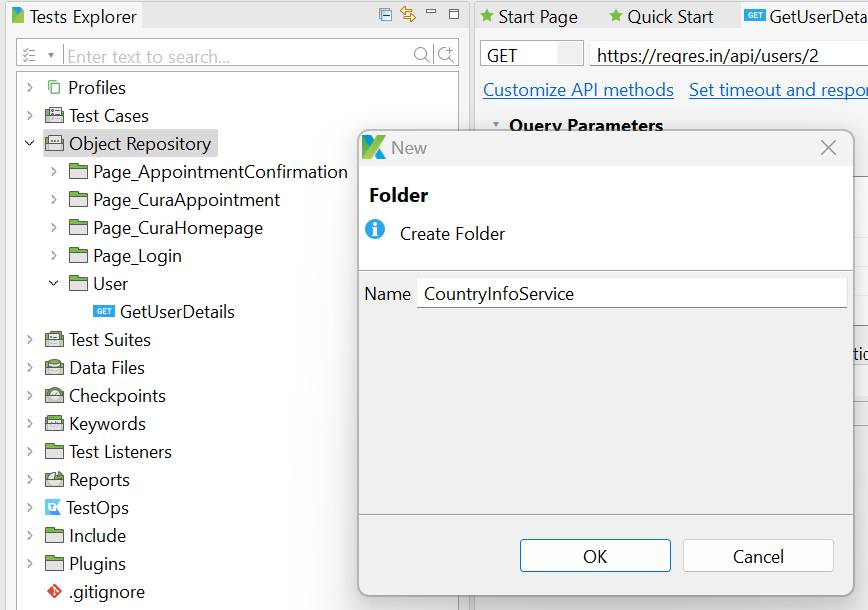
1. Berikut tampilan file yang dibuat, selanjutnya klik icon play.



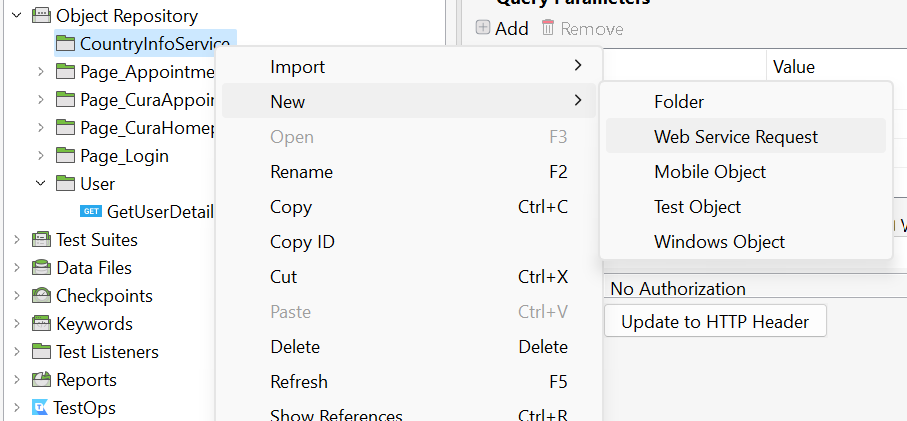
1. Berikut tampilaan hasilnya, yaitu GET user dengan id 2 berhasil dilakukan, ditandai dari Responsenya yaitu status 200 OK



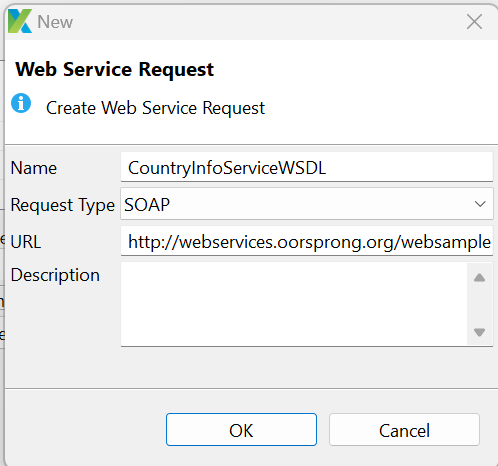
1. Selanjutnya buat folder baru dengan nama “CountryInfoService” pada Object Repository



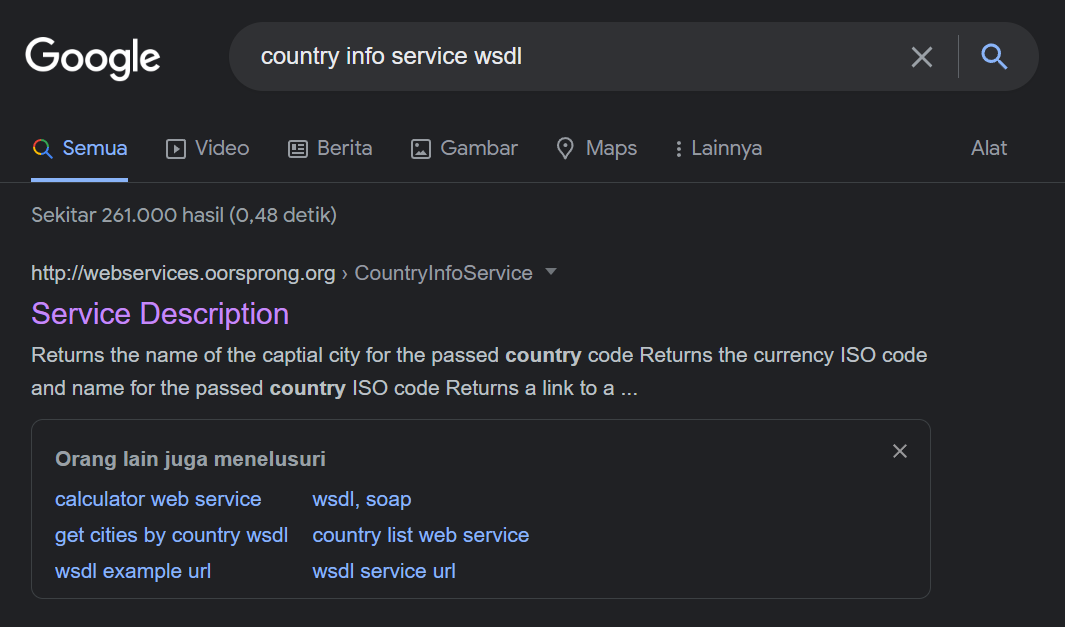
1. Klik kanan pada folder “CountryInfoService”, pilih new dan klik Web Service Request



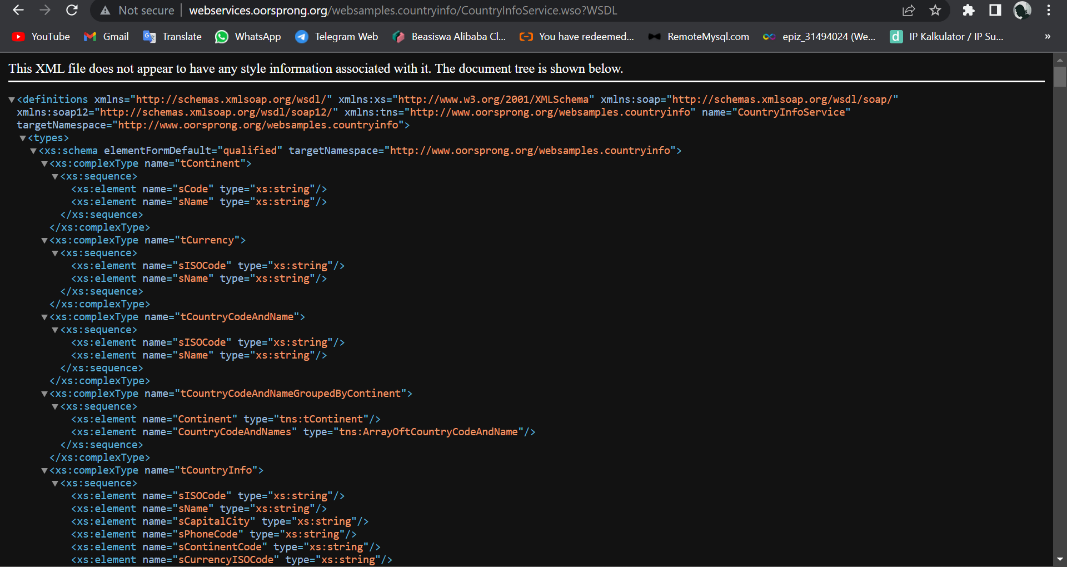
1. Kemudian isikan name sebagai berikut, pilih SOAP sebagai Request Type dan tempelkan url berikut.



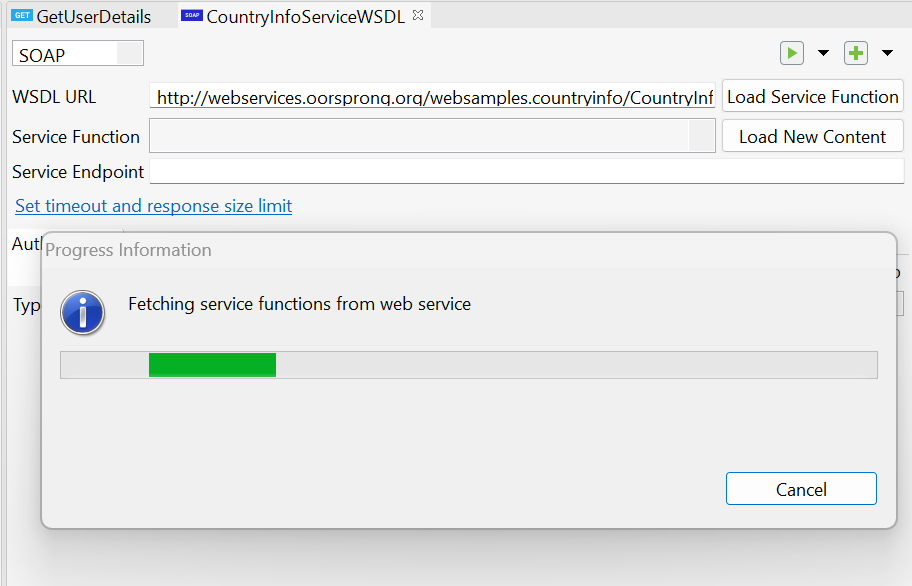
1. Untuk mendapatkan URL diatas dapat di searching di google, buka website berikut.



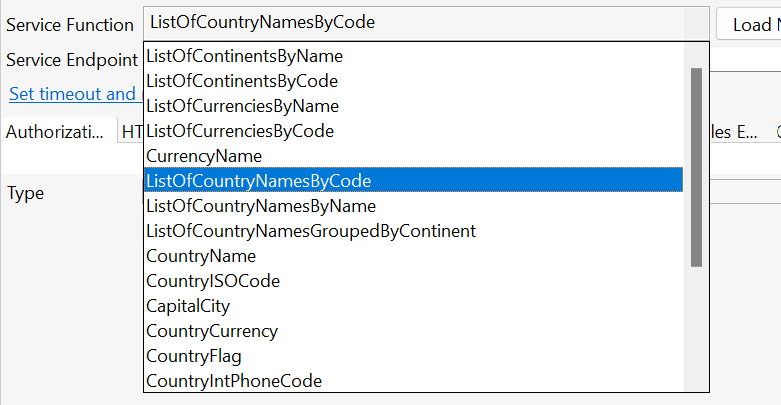
1. Berikut tampilan website, lalu copy link addressnya.



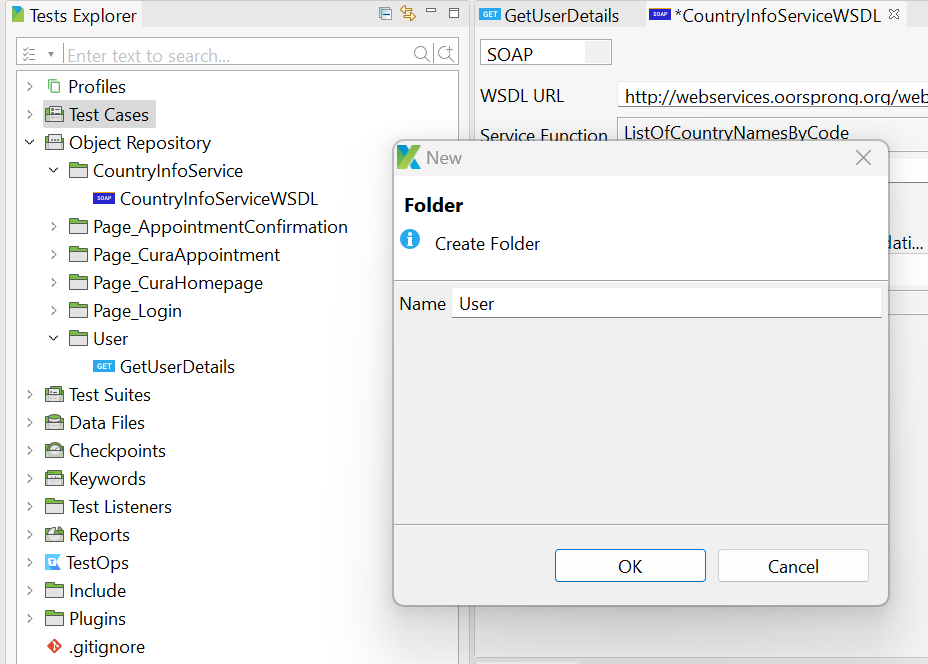
1. Berikut Tampilan File yang telah kita buat, klik “Load Service Function”.



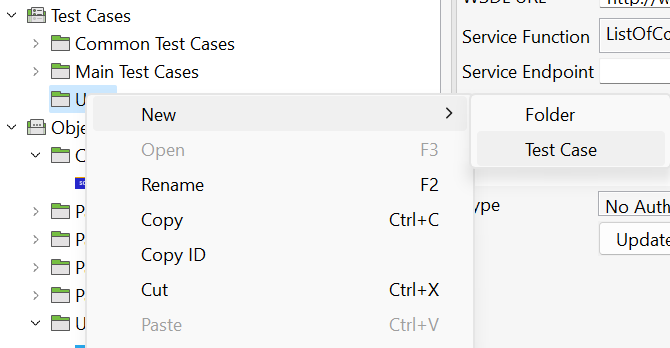
1. Pada Service Function pilih “ListOfCountryNamesByCode”



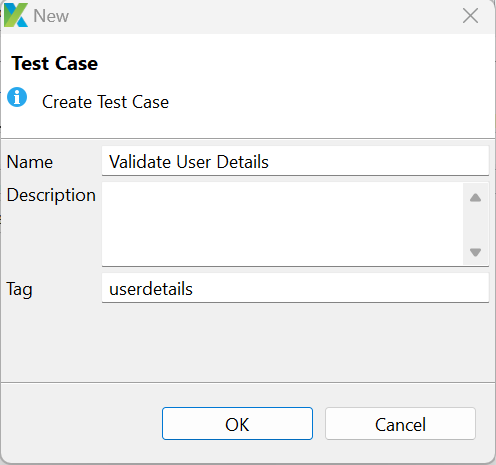
1. Langkah selanjutnya pada Test Explorer, buat folder “User” didalam Test Case



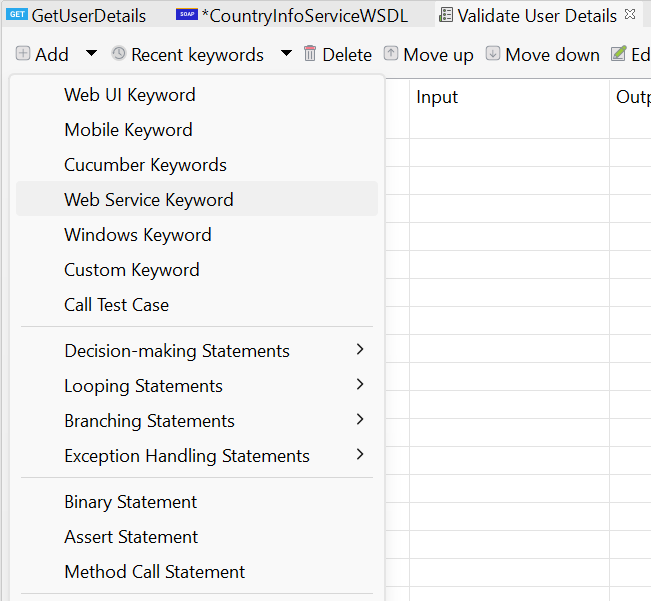
1. Pada folder “User” klik kanan pilih New dan klik Test Case



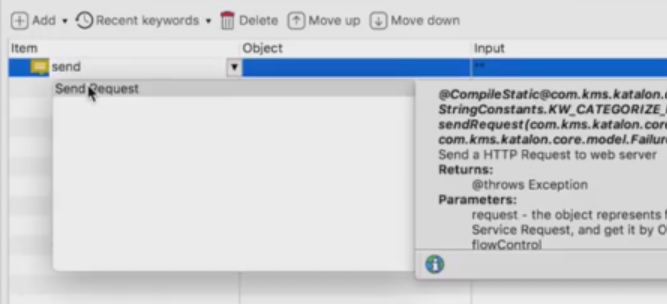
1. Isikan form seperti berikut, klik Ok



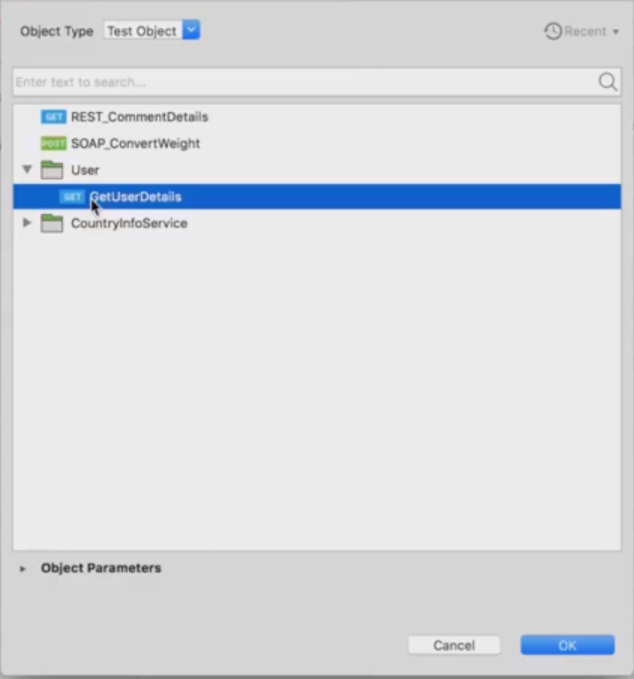
1. Selanjutnya klik dropdown Add pilih Web Service Keyword



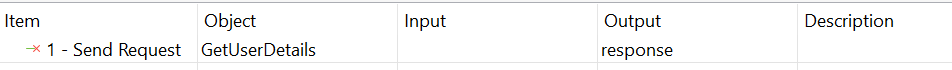
1. Gantikan Namanya menjadi “Send Request”



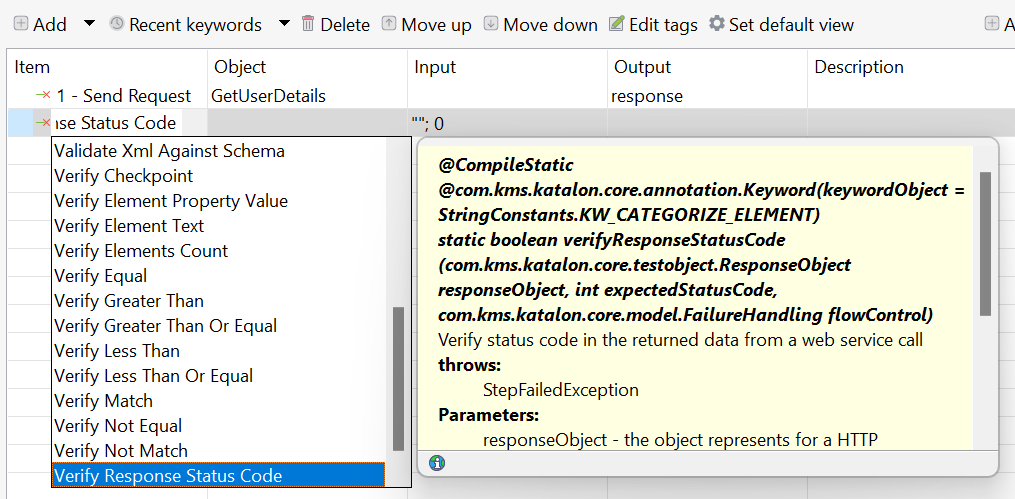
1. Pada Object yang berisi null, klik lalu pilih “GetUserDetails”



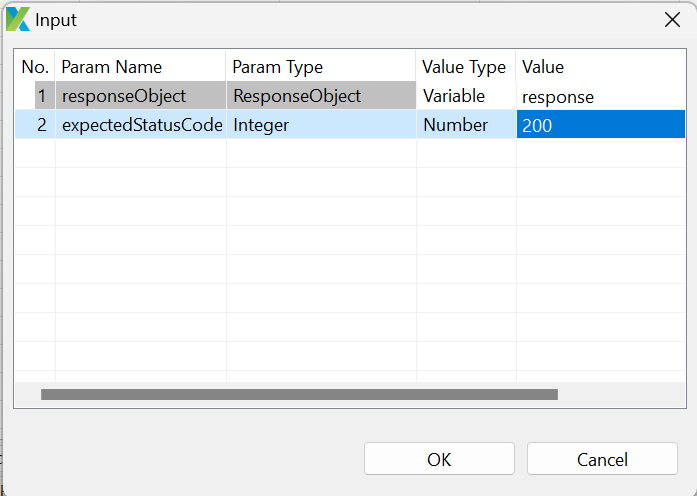
1. Pada Output isikan “response”



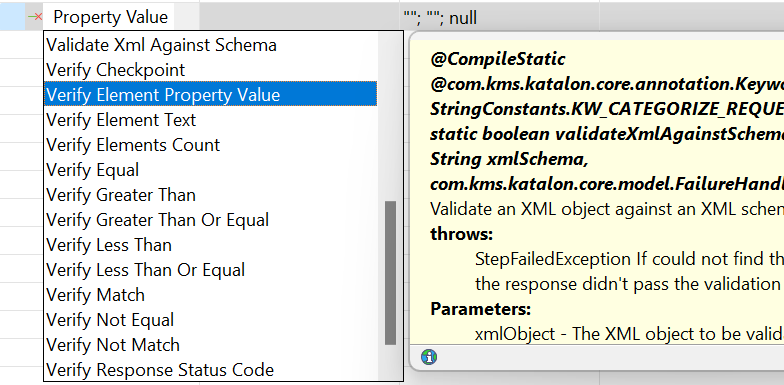
1. Tambahkan “Verify Response Status Code”



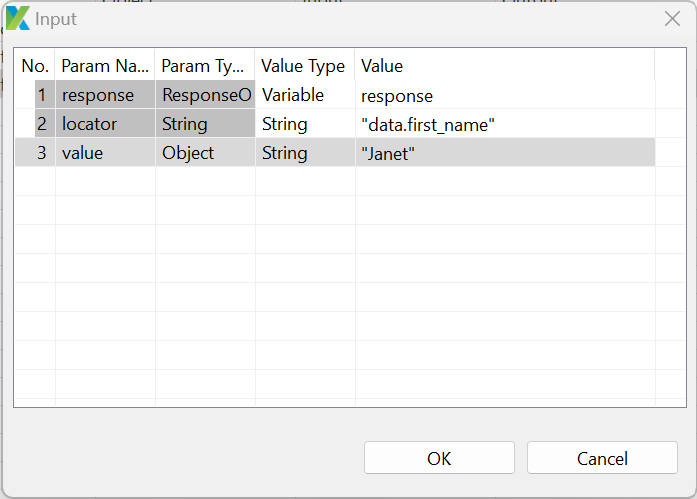
1. Ubah input dari “Verify Response Status Code” seperti berikut



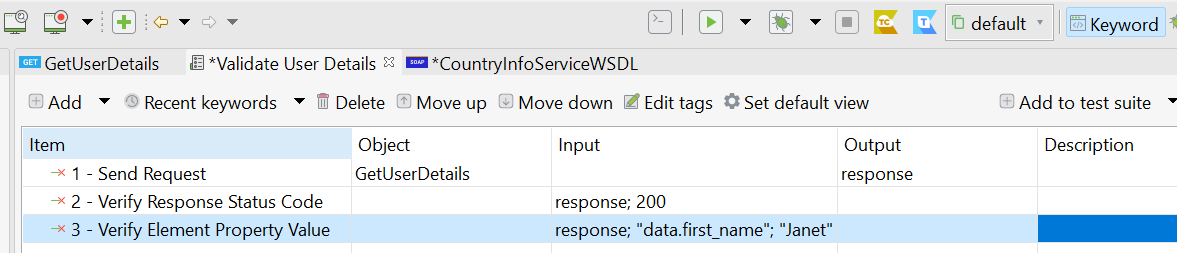
1. Kemudian tambahkan “Verify Element Property Value”



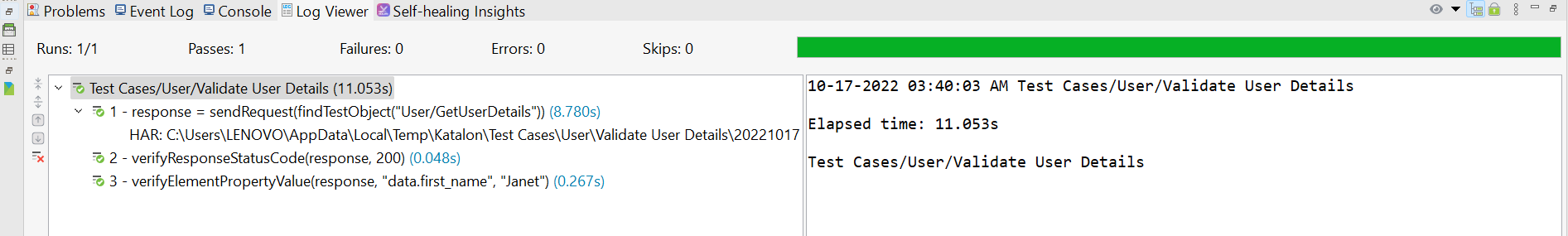
1. Ubah input dari “Verify Element Property Value” seperti berikut



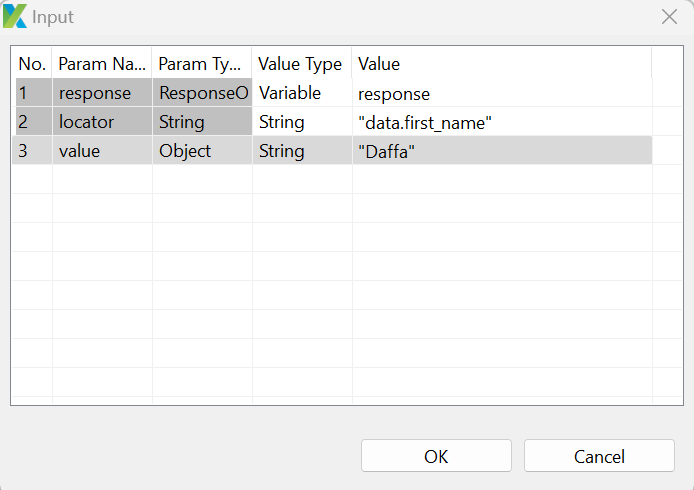
1. Sehingga menjadi seperti berikut, selanjutnya klik icon play



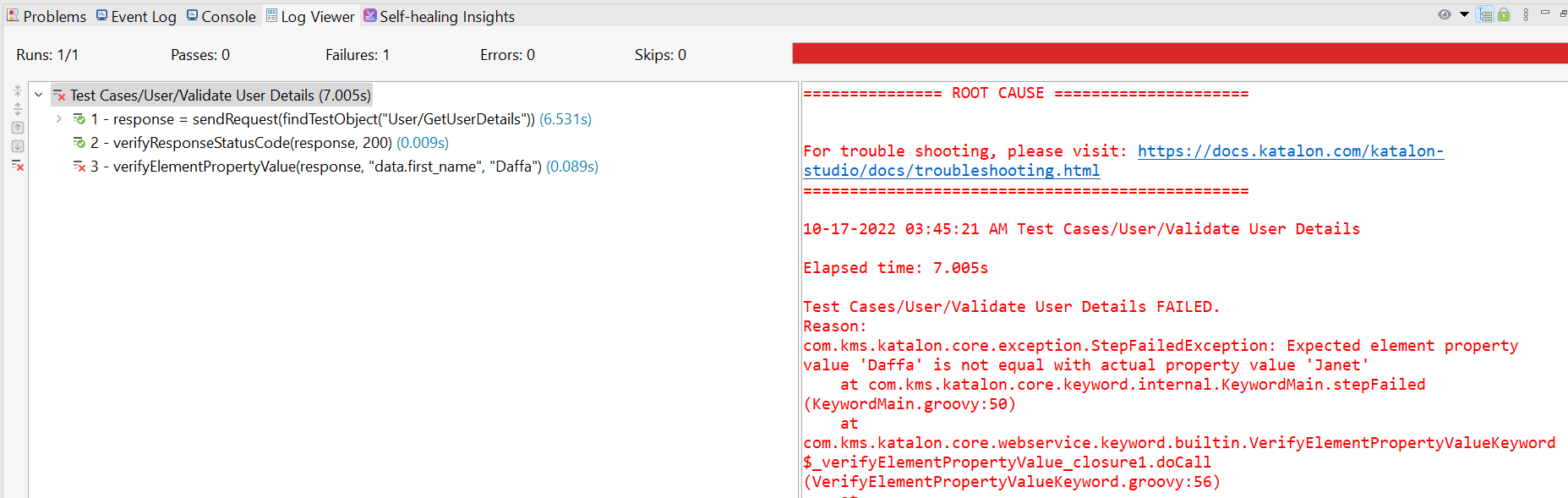
1. Berikut hasilnya, dapat dilihat lewat Log Viewer, hasilnya tidak ada errornya



1. Namu Ketika kita ganti inputan pada “Verify Element Property Value” seperti berikut

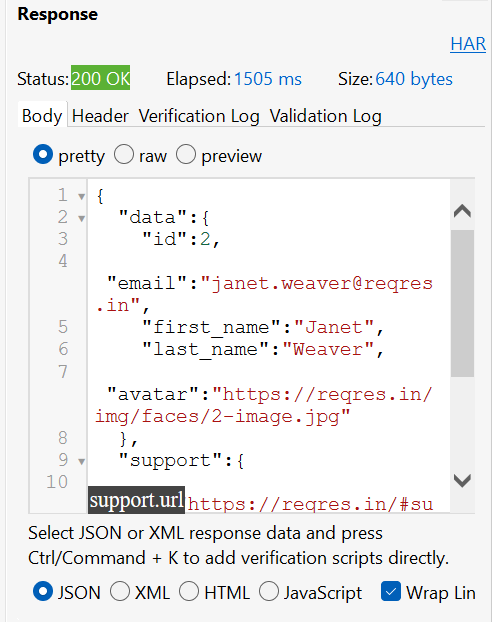


1. Maka hasilnya akan error, seperti berikut

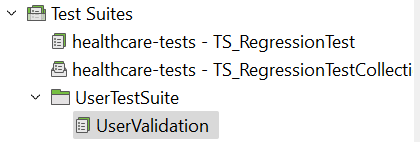


1. Penyebab Error dikarenakan pada Object Respository terdapat folder User yang didalamnya memiliki file GetUserDetails yang berisi sebagai berikut.

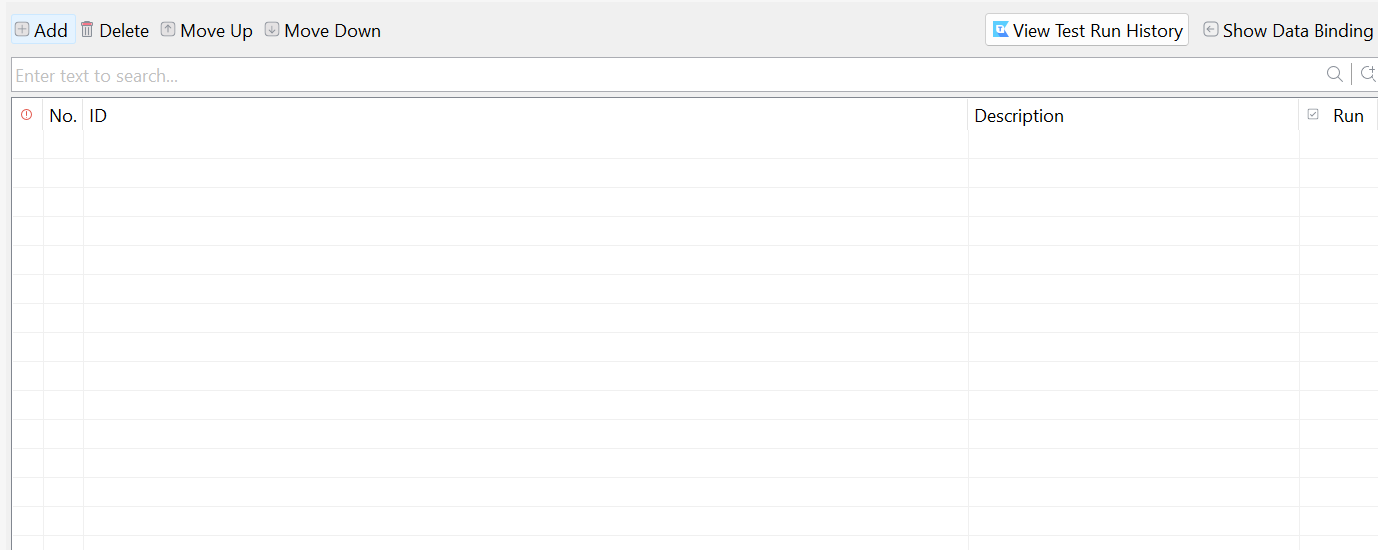
Oleh karena itu inputan pada “Verify Element Property Value” hanya menerima “Janet” sebagai string selain itu akan error.



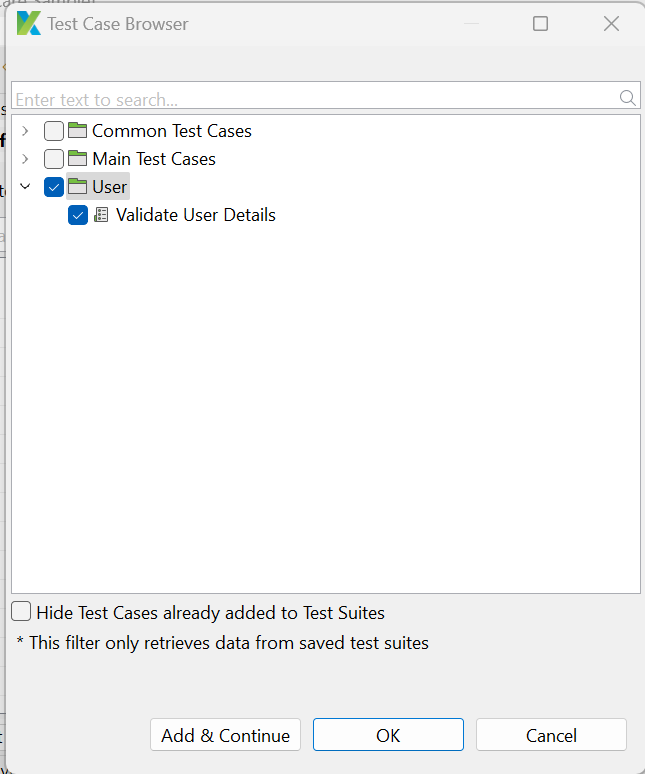
1. Buatkan test suite didalam folder UserTestSuite berikan nama “UserValidation”



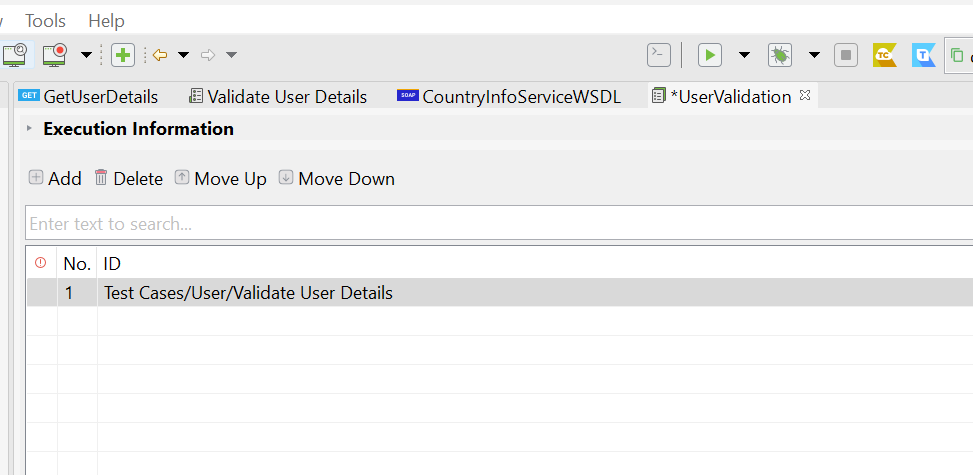
1. Berikut tampilannya, klik add



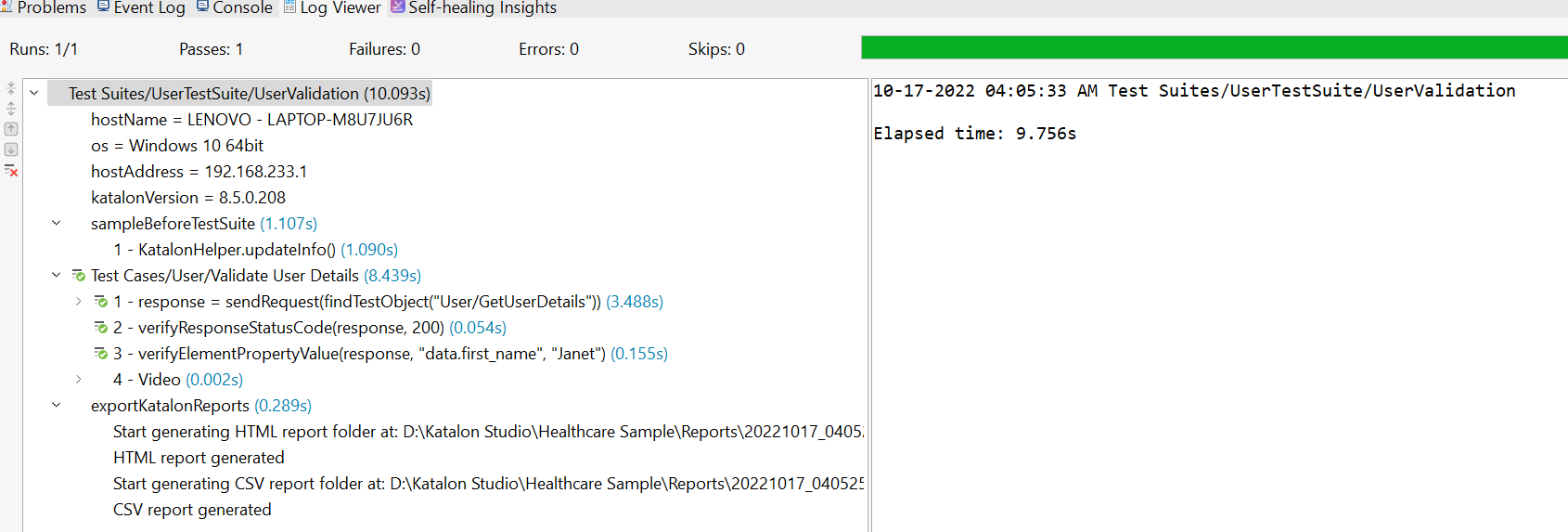
1. Centang Validate User Details



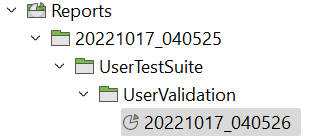
1. Berikut setelah di add, klik icon play



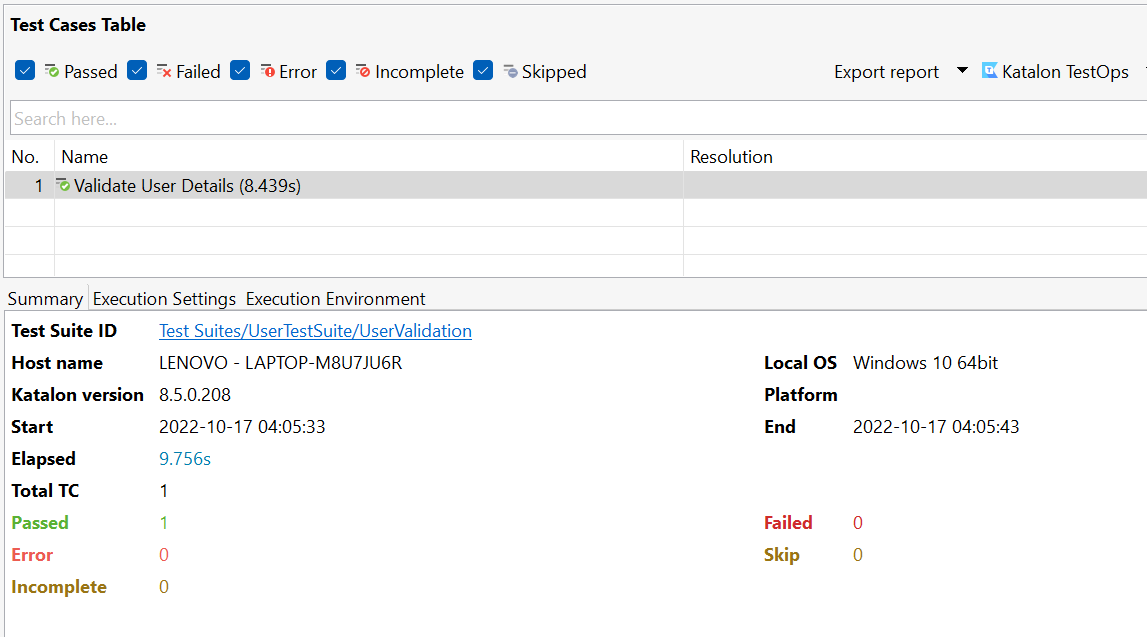
1. Berikut hasil dari Log Viewer



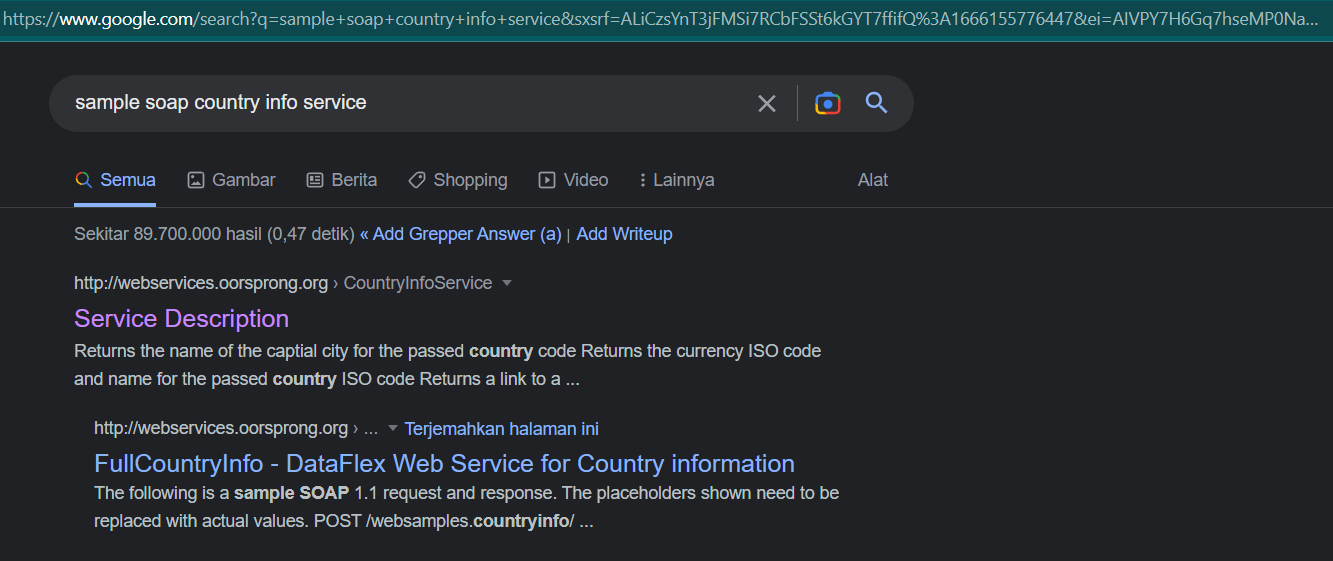
1. Selanjutnya kita buka Reports untuk melihat hasil pengujian



1. Berikut hasil report dari pengujian file json yang telah dilakukan



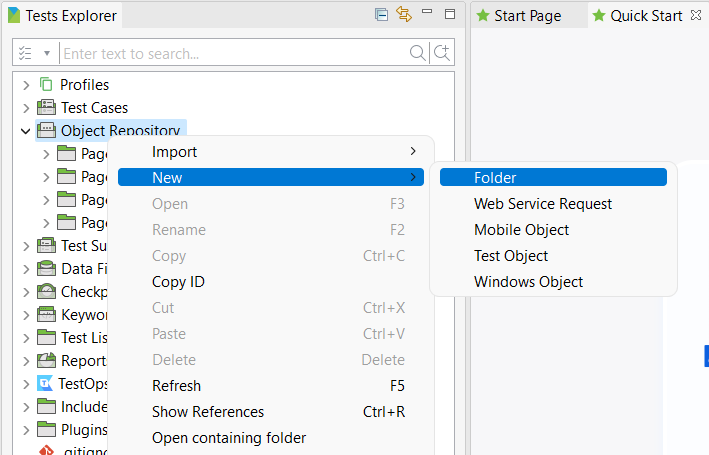
1. **API CHAINING | SOAP-XML | HOW TO SEND VALUE FROM ONE API TO OTHER**
2. W



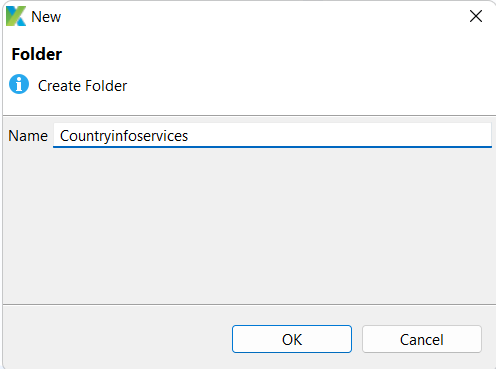
1. W



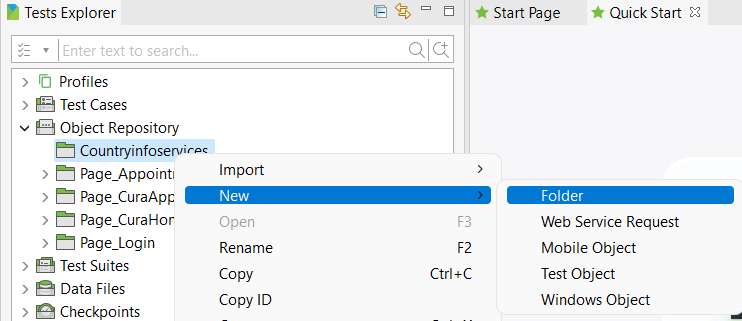
1. W



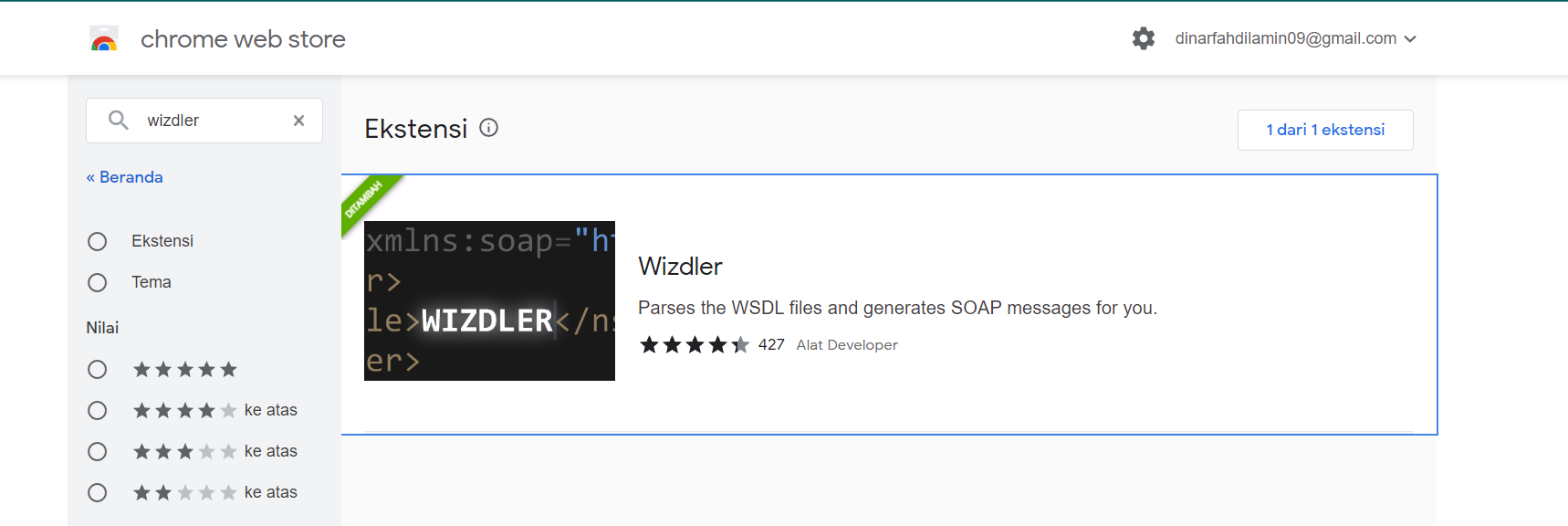
1. W



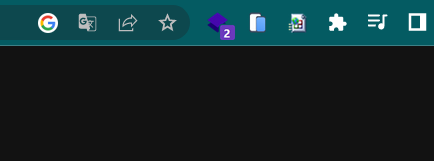
1. W



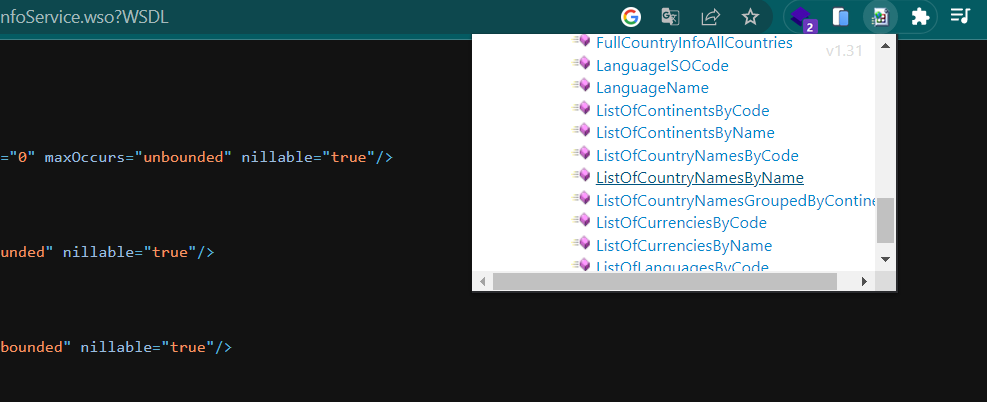
1. W



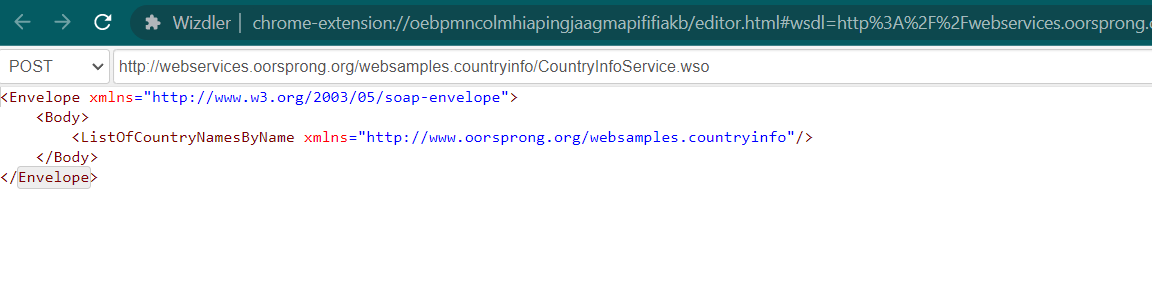
1. W



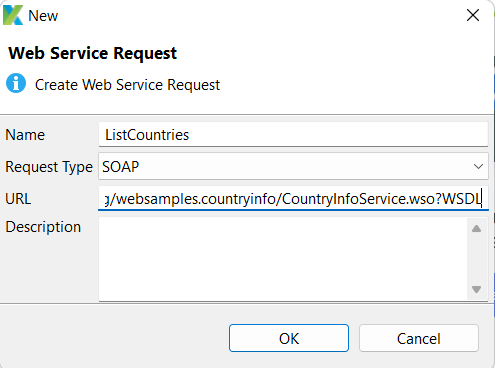
1. W



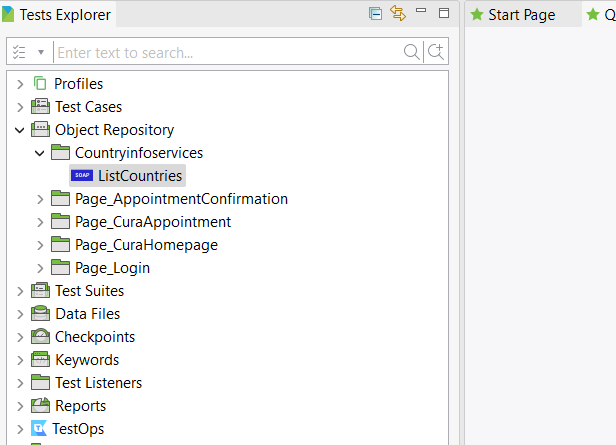
1. W



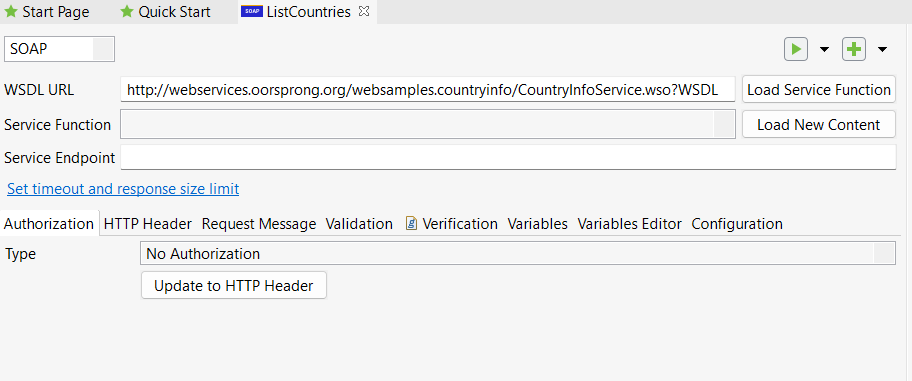
1. W



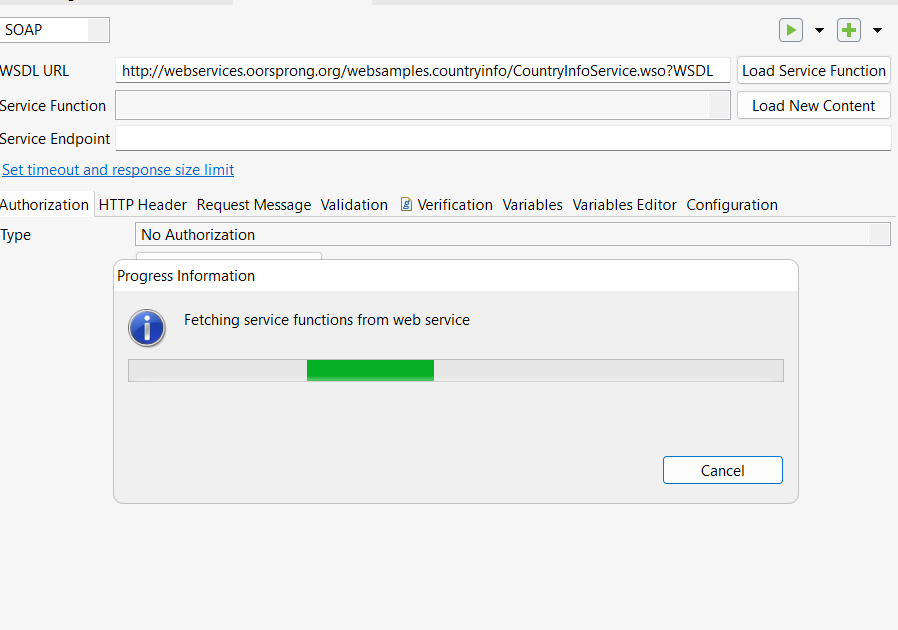
1. W



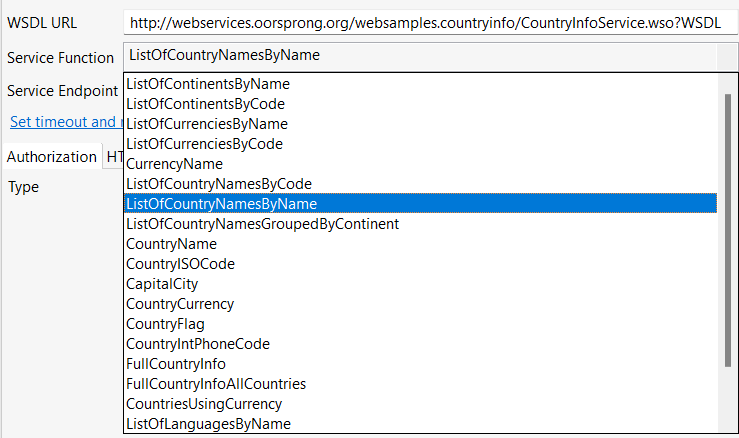
1. W



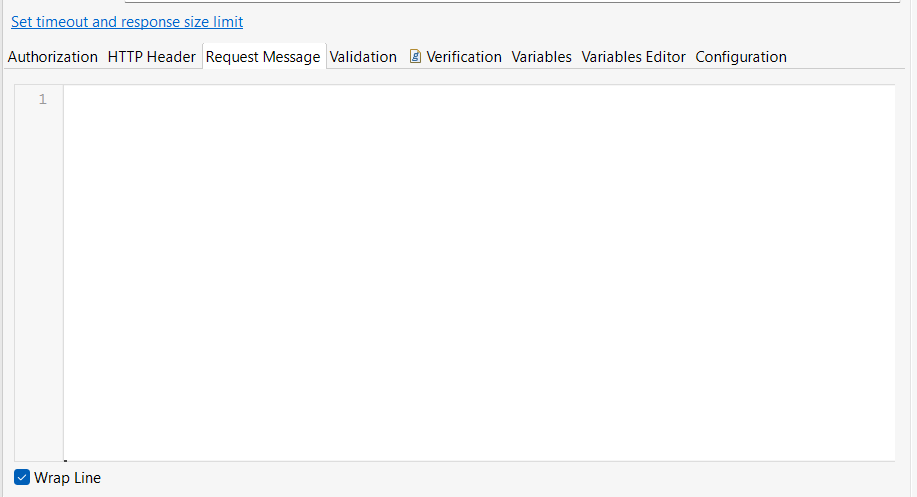
1. W



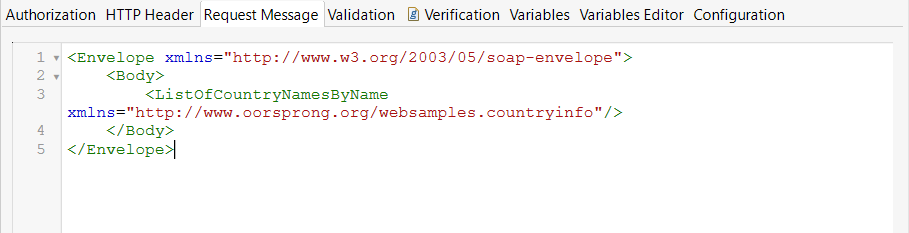
1. W



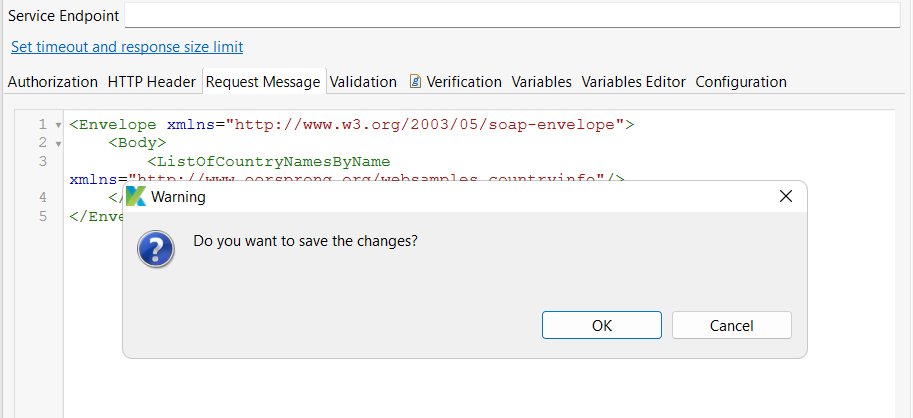
1. W



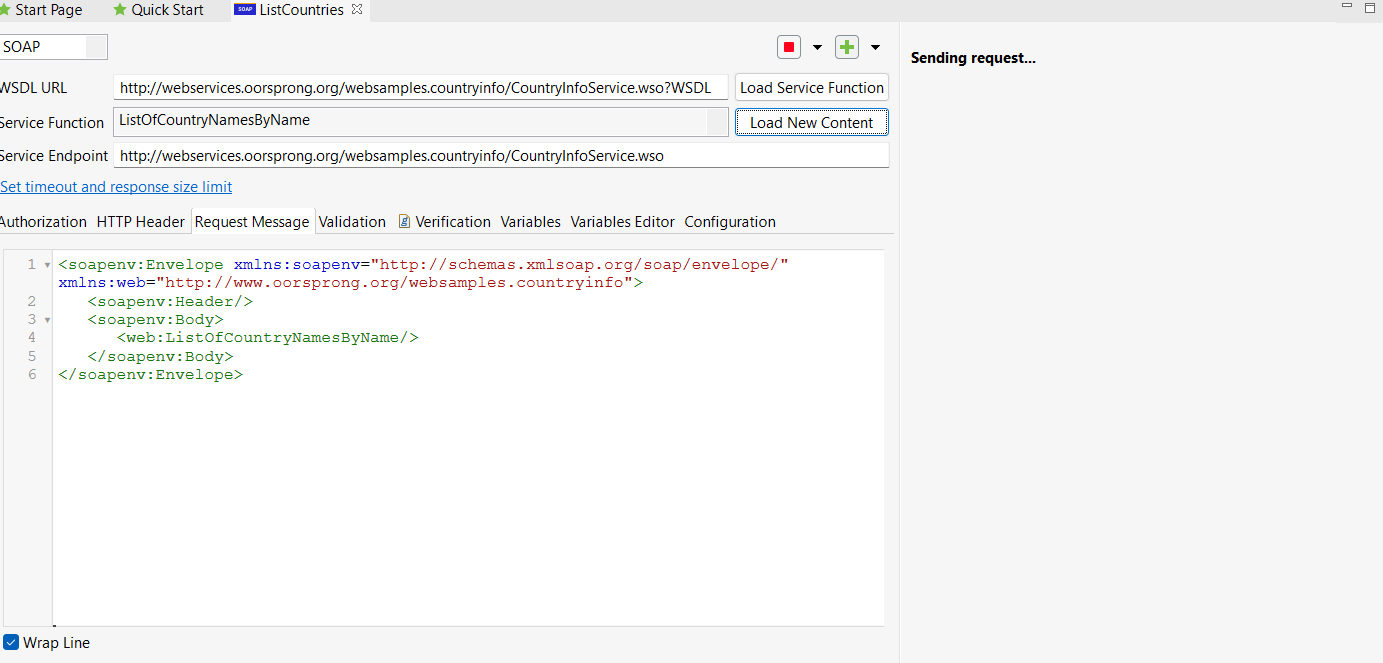
1. W



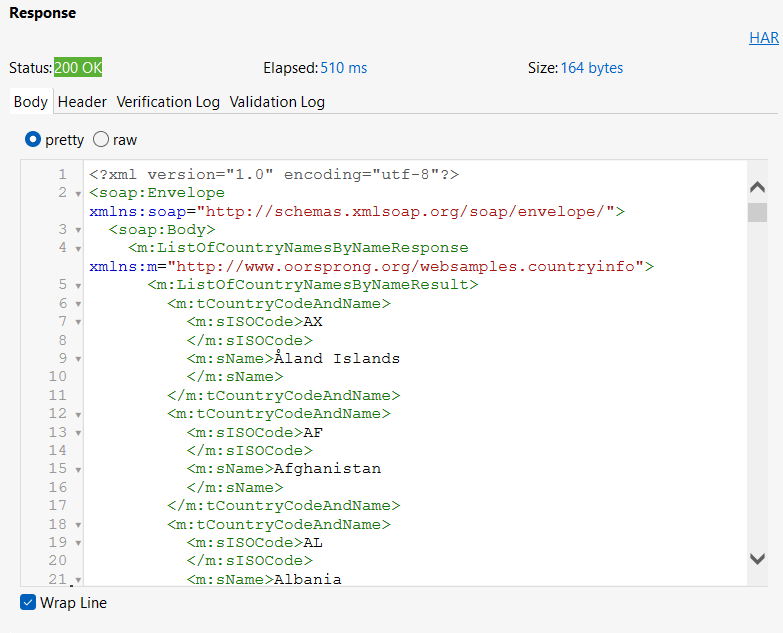
1. W



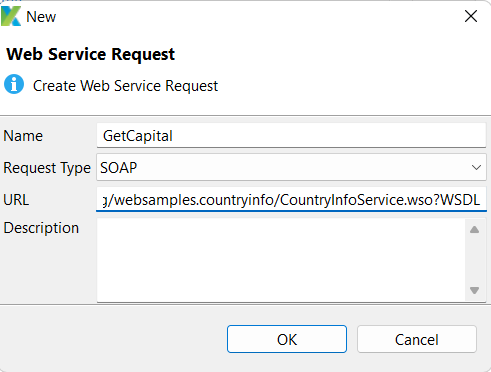
1. W



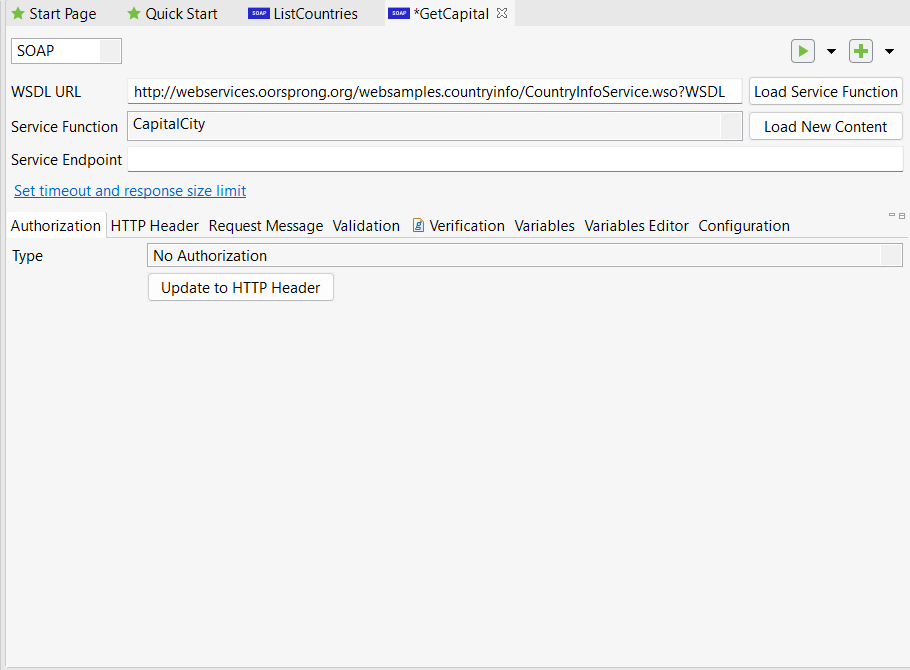
1. W



1. W



1. W



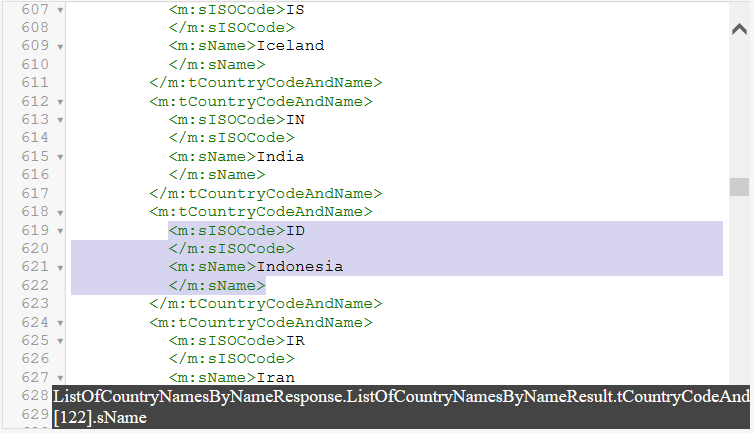
1. W



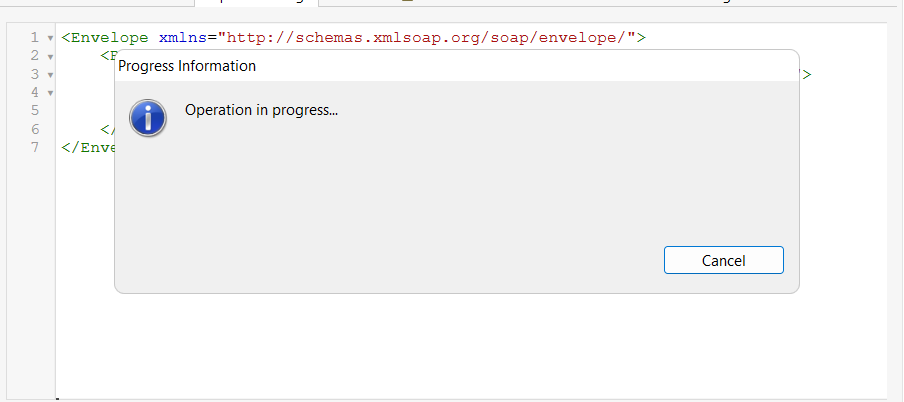
1. W



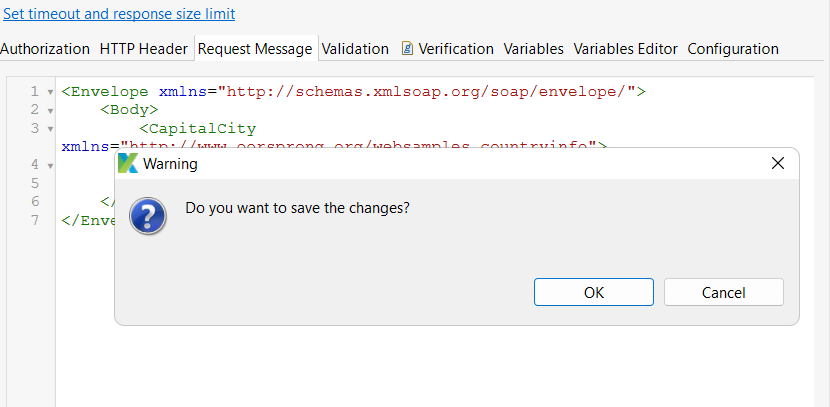
1. W



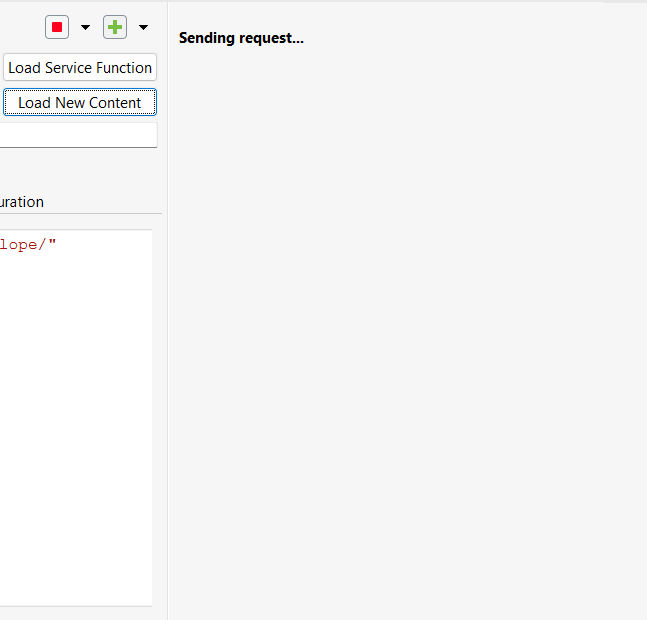
1. W



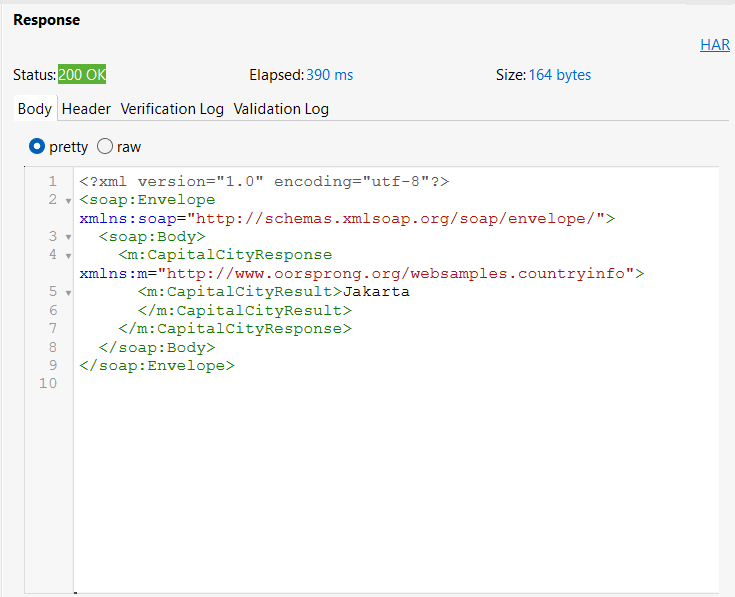
1. w



1. w



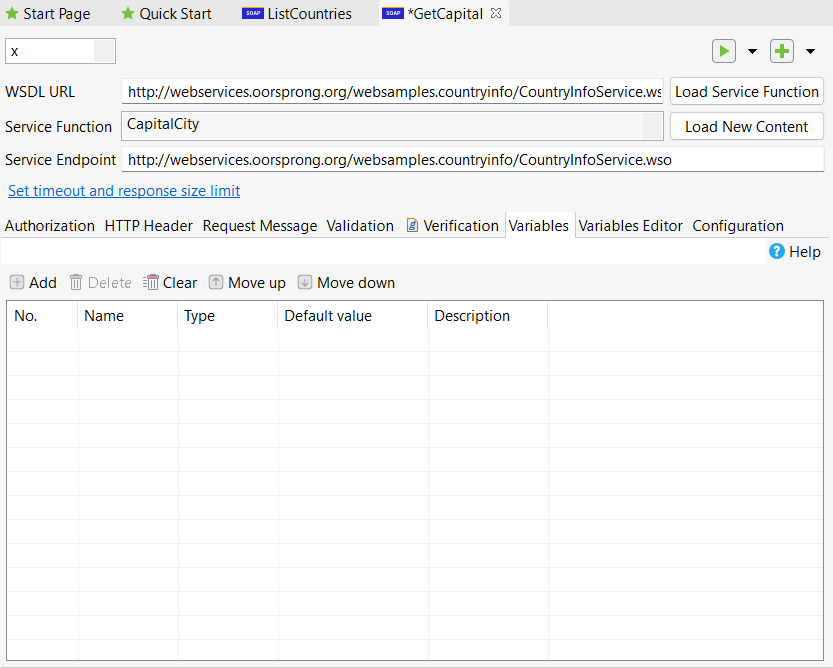
1. w



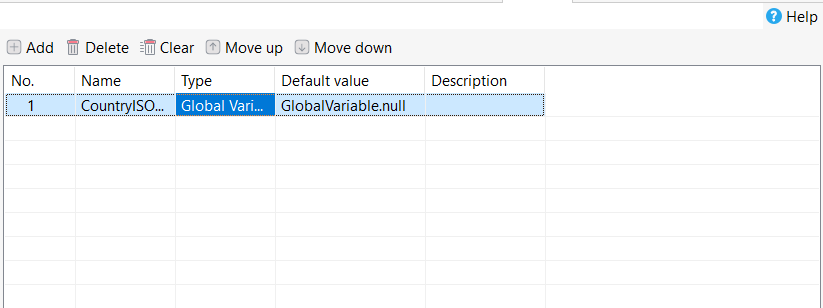
1. w



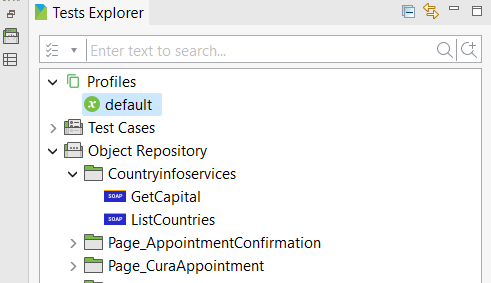
1. w



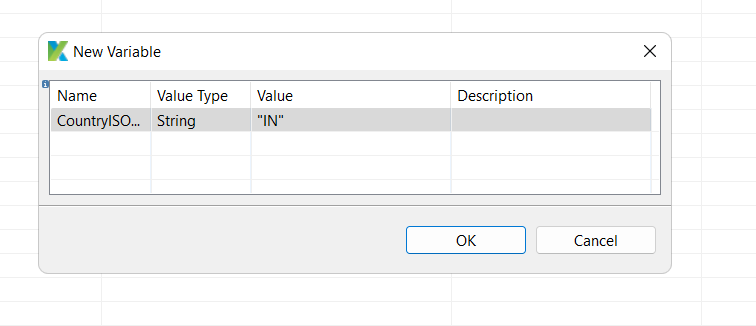
1. w



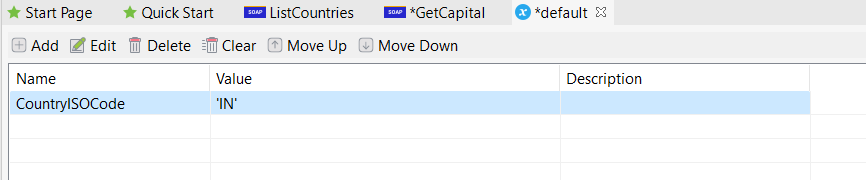
1. w



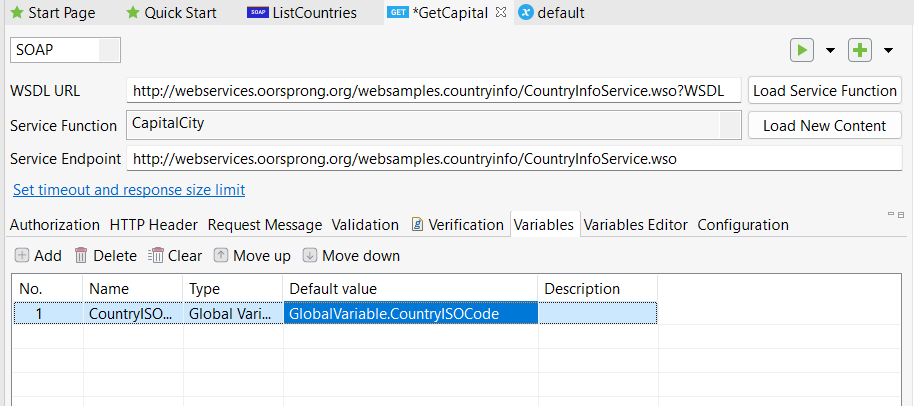
1. w



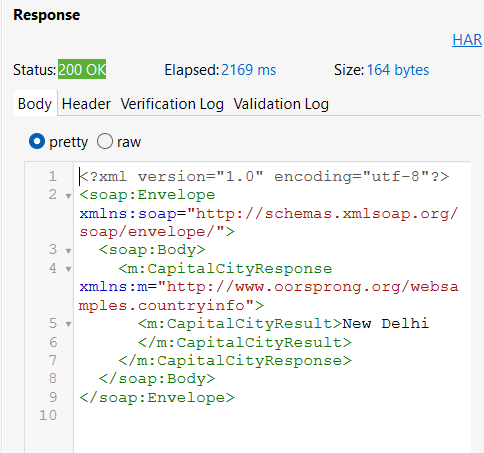
1. w



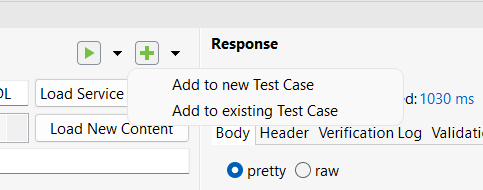
1. w



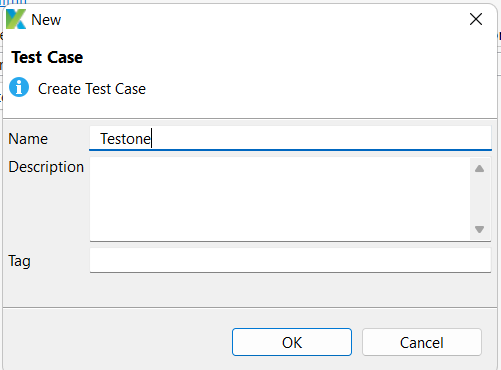
1. w



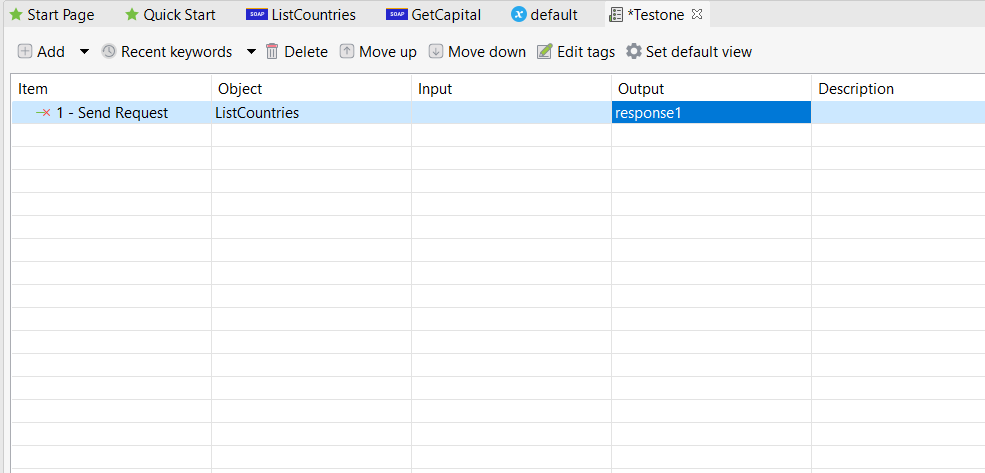
1. w



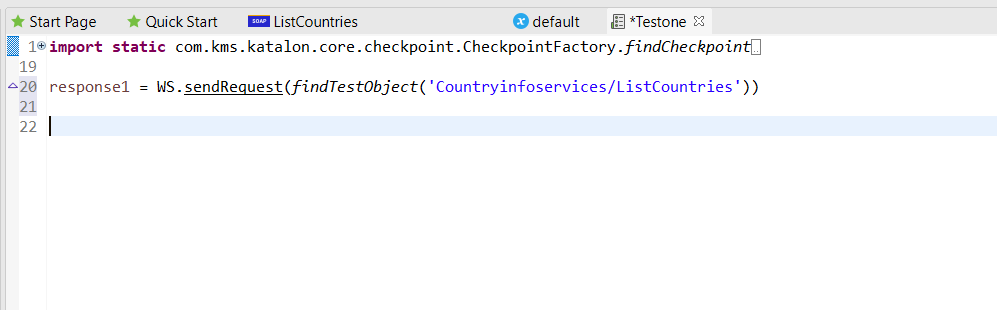
1. w



1. w



1. w



1. w



1. w