Table of Contents

[How Gatling can send multiple requests or parallel requests during runtime? 2](#_Toc96866910)

[How to capture both key and value of a key-value pair using correlation? 3](#_Toc96866911)

[How to do correlation for redirection urls? See code below 4](#_Toc96866912)

[Don’t follow redirection 6](#_Toc96866913)

[While recording a HTTPS site, user got certificate error on browser. How to fix this issue to enable recording. 10](#_Toc96866914)

[User login once, perform mutlile activities and log out 12](#_Toc96866915)

[Balck List HTML Resources 13](#_Toc96866916)

[Infer html Resources and show in Report 17](#_Toc96866917)

[Create Group for login and log out Download Resources but don’t show in Report 21](#_Toc96866918)

[Gatling - exception handling 23](#_Toc96866919)

# How Gatling can send multiple requests or parallel requests during runtime?

val *scnRefactor*= scenario("Fligh Booking run sequentially" )  
 .exec(Reserve.*methodReserve*)  
 .exec(Purchase.*methodPurchase*)  
 .exec(Confirmation.*methodConfirmation*)  
  
 val *scnParallel* =scenario(("Run parallelly"))  
 .exec( Reserve.*methodReserve*, Purchase.*methodPurchase*, Confirmation.*methodConfirmation*)

val *scnRefactor*= scenario("Fligh Booking" )  
 .during(60){  
 feed((*csvFedderBlazeDemoFilightDetails*))  
 .exec(Reserve.*methodReserve*)  
 .exec(Purchase.*methodPurchase*)  
 .exec(Confirmation.*methodConfirmation*)  
 }

# How to capture both key and value of a key-value pair using correlation?

Printing json Key value pair

SessionAttribute(Session(BlazeDemoLoginLogout,1,HashMap(gatling.http.cache.baseUrl -> https://jsonplaceholder.typicode.com, jsonKEYVALUE1 -> Vector({"userId":1,"id":3,"title":"fugiat veniam minus","completed":false}), gatling.http.ssl.sslContexts -> io.gatling.http.util.SslContexts@2a2fc6fb, gatling.http.cache.dns -> io.gatling.http.resolver.ShufflingNameResolver@6ae5dc0d, gatling.http.cache.contentCache -> [io.gatling.core.util.cache.Cache@4e7e6e5),OK,List(),io.gatling.core.protocol.ProtocolComponentsRegistry$$Lambda$361/1685134322@38dc26fb,io.netty.channel.nio.NioEventLoop@1e097d59),jsonKEYVALUE1](mailto:io.gatling.core.util.cache.Cache@4e7e6e5),OK,List(),io.gatling.core.protocol.ProtocolComponentsRegistry$$Lambda$361/1685134322@38dc26fb,io.netty.channel.nio.NioEventLoop@1e097d59),jsonKEYVALUE1))

SessionAttribute(Session(BlazeDemoLoginLogout,1,HashMap(gatling.http.cache.baseUrl -> https://jsonplaceholder.typicode.com, jsonKEYVALUE1 -> Vector({"userId":1**,"id":3,"**title":"fugiat veniam minus","completed":false}, {"userId":1,"**id":4,"**title":"et porro tempora","completed":true}), gatling.http.ssl.sslContexts -> io.gatling.http.util.SslContexts@68c6a0f1, gatling.http.cache.dns -> io.gatling.http.resolver.ShufflingNameResolver@6e5ce968, gatling.http.cache.contentCache -> io.gatling.core.util.cache.Cache@40b15a0a),OK,List(),io.gatling.core.protocol.ProtocolComponentsRegistry$$Lambda$361/732189840@5c5b1eee,io.netty.channel.nio.NioEventLoop@1e097d59),jsonKEYVALUE1)

package blazedemo  
  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class JsonResponseSaveKEYValue extends Simulation {  
 val *httpProtocol* = http  
 .baseUrl("https://jsonplaceholder.typicode.com")  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 .exec(  
 http("T0 GetKEYVALUE")  
 .get("/todos/")  
 .check(  
 jsonPath("$[2,3]").findAll.optional.saveAs("jsonKEYVALUE1")  
 )  
 )  
 .exec(  
 session =>{  
 *println*("Printing json Key value pair")  
 *println*(session("jsonKEYVALUE1"))  
 session;  
 }  
 )  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

# How to do correlation for redirection urls? See code below

1 Url Redirection:

2 codes, first allows rediretion

Secod code, disables redirection

It will feth redirected url from repsone body as well from response headers

Uses newly fetch url to navigate to application

package blazedemo  
  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class BlazeDemoLoginLogout extends Simulation {  
  
 var *tokenExpr*="""<input type="hidden" name="\_token" value="(.\*)">"""  
 var *redirectingUrlAfterLoginPost*="""<title>Redirecting to (.\*)"""  
  
 *// <title>Redirecting to https://blazedemo.com/home</title>* val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 *//.disableFollowRedirect* .inferHtmlResources(BlackList(""".\*\.js""", """.\*\.css""", """.\*\.gif""", """.\*\.jpeg""", """.\*\.jpg""", """.\*\.ico""", """.\*\.woff""", """.\*\.woff2""", """.\*\.(t|o)tf""", """.\*\.png""", """.\*detectportal\.firefox\.com.\*"""), WhiteList())  
 .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9")  
 .acceptEncodingHeader("gzip, deflate")  
 .acceptLanguageHeader("en-GB,en;q=0.9,en-US;q=0.8,mr;q=0.7,hi;q=0.6")  
 .userAgentHeader("Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.109 Safari/537.36")  
  
 val *headers\_0* = *Map*(  
 "origin" -> "https://blazedemo.com",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *headers\_1* = *Map*(  
 "accept" -> "text/css,\*/\*;q=0.1",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "style",  
 "sec-fetch-mode" -> "no-cors",  
 "sec-fetch-site" -> "cross-site",  
 "x-client-data" -> "CJO2yQEIprbJAQjEtskBCKmdygEInvnLAQjnhMwBCJuczAEIrJzMAQ==")  
  
 val *headers\_2* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *uri2* = "https://fonts.googleapis.com/css"  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 *// clear browser cache, close all browser , to avoid 419 status code error.  
 // try to register new user as users are flushed out from demo sites* .exec(flushHttpCache)  
 .exec(  
 http("T0 get Token from Login get call")  
 .get("/login")  
 .headers(*headers\_0*)  
 .check(regex(*tokenExpr*).saveAs("\_token"))  
 )  
 .exec(http("T1 Login")  
 .post("/login")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token}")  
 .formParam("email", "zanjadkamlesh@gmail.com")  
 .formParam("password", "Testing@1")  
 *//.check(status.in(302,304))  
 //.check(regex(redirectingUrlAfterLoginPost).saveAs("redirectUrlAfterLogin"))  
// .resources(http("BlazeDemoLoginLogout\_1:GET\_https://fonts.googleapis.com/css?family=Raleway:300\_400\_600")  
// .get(uri2 + "?family=Raleway:300,400,600")  
// .headers(headers\_1))* )  
 .pause(2)  
  
 .exec(http("T2 Navigate to base url")  
 .get("/")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("T3 Navigate to Vacation")  
 .get("/vacation.html")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("T4 Navigate to Home")  
 .get("/home")  
 .headers(*headers\_2*)  
 .check(regex(*tokenExpr*).saveAs("\_token2"))  
  
*// .resources(http("BlazeDemoLoginLogout\_5:GET\_https://fonts.googleapis.com/css?family=Raleway:300\_400\_600")  
// .get(uri2 + "?family=Raleway:300,400,600")  
// .headers(headers\_1))* )  
 .pause(2)  
 .exec(http("T5 Logout")  
 .post("/logout")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token2}")  
 *//.formParam("\_token", "FLyQO3cggY4Is8OLbDRntffY8QkJzAQIUukYcfb1")* )  
 .pause(2)  
 .exec(flushHttpCache)  
  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/blazedemologinlogout-20220227052905670/index.html

Table

Description automatically generated

# Don’t follow redirection

package blazedemo  
  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class BlazeDemoDisableRedirect extends Simulation {  
  
 var *tokenExpr*="""<input type="hidden" name="\_token" value="(.\*)">"""  
 var *redirectingUrlAfterLoginPost*="""<title>Redirecting to (.\*)"""  
  
 *// <title>Redirecting to https://blazedemo.com/home</title>* val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 **.disableFollowRedirect**  
 .inferHtmlResources(BlackList(""".\*\.js""", """.\*\.css""", """.\*\.gif""", """.\*\.jpeg""", """.\*\.jpg""", """.\*\.ico""", """.\*\.woff""", """.\*\.woff2""", """.\*\.(t|o)tf""", """.\*\.png""", """.\*detectportal\.firefox\.com.\*"""), WhiteList())  
 .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9")  
 .acceptEncodingHeader("gzip, deflate")  
 .acceptLanguageHeader("en-GB,en;q=0.9,en-US;q=0.8,mr;q=0.7,hi;q=0.6")  
 .userAgentHeader("Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.109 Safari/537.36")  
  
 val *headers\_0* = *Map*(  
 "origin" -> "https://blazedemo.com",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *headers\_1* = *Map*(  
 "accept" -> "text/css,\*/\*;q=0.1",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "style",  
 "sec-fetch-mode" -> "no-cors",  
 "sec-fetch-site" -> "cross-site",  
 "x-client-data" -> "CJO2yQEIprbJAQjEtskBCKmdygEInvnLAQjnhMwBCJuczAEIrJzMAQ==")  
  
 val *headers\_2* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *uri2* = "https://fonts.googleapis.com/css"  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 **.exec(flushHttpCache)**  
 .exec(  
 http("T0 get Token from Login get call")  
 .get("/login")  
 .headers(*headers\_0*)  
 .check(regex(*tokenExpr*).saveAs("\_token"))  
 )  
 .exec(http("T1 Login")  
 .post("/login")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token}")  
 .formParam("email", "zanjadkamlesh@gmail.com")  
 .formParam("password", "Testing@1")  
 .check(*status*.in(200,302,304))  
 **.check(regex(*redirectingUrlAfterLoginPost*).saveAs("redirectUrlAfterLogin"))  
 .check(*header*("location").saveAs("gLocation"))**  
*// .resources(http("BlazeDemoLoginLogout\_1:GET\_https://fonts.googleapis.com/css?family=Raleway:300\_400\_600")  
// .get(uri2 + "?family=Raleway:300,400,600")  
// .headers(headers\_1))* )  
 .pause(2)  
 .exec(session => {  
 val fetchURLFromResponseBody = session("redirectUrlAfterLogin")  
 val fetchURLFromResponseHeaders = session("gLocation")  
 *println*("Printing session varialbes")  
 *println*("fetchURLFromResponseBody '" + fetchURLFromResponseBody +"'")  
 *println*("fetchURLFromResponseHeaders '" + fetchURLFromResponseHeaders +"'")  
 session  
 })  
 .exec(  
 **http(" T2 Redirect Url")  
 .get("${gLocation}")  
 .headers(*headers\_1*)** */// header for redirect url will be different, need to fetch it using programs* .check(regex("You are logged in!").exists)  
 .check(regex("""<div class="panel-heading">Dashboard</div>""").exists)  
 )  
 .exec(http("BlazeDemoLoginLogout\_2:GET\_https://blazedemo.com/")  
 .get("/")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("BlazeDemoLoginLogout\_3:GET\_https://blazedemo.com/vacation.html")  
 .get("/vacation.html")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("BlazeDemoLoginLogout\_4:GET\_https://blazedemo.com/home")  
 .get("/home")  
 .headers(*headers\_2*)  
 .check(regex(*tokenExpr*).saveAs("\_token2"))  
 )  
 .pause(2)  
  
 .exec(flushHttpCache)  
  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/blazedemodisableredirect-20220227055916249/index.html

Chart, table, calendar

Description automatically generated

Log details

# While recording a HTTPS site, user got certificate error on browser. How to fix this issue to enable recording.

Refer Presentation

Training\_4\_GatlingDemoStore\_Ecomerce 2

Slide 10 onward

* **Configure browser proxy**
* Network traffic should be redirected through Gatling Recorder in order to be captured in HTTP Proxy mode. By default, recorder works on 8000 port, but this is configurable. If a web application is being tested then the browser should be configured to work through the Gatling Recorder proxy.

Firefox is configured from Tools -> Options -> Advanced -> Network -> Settings… -> Manual proxy configurations: -> localhost with Port: 8000, Use this proxy server for all protocols.

Graphical user interface, text, application

Description automatically generated

val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 .inferHtmlResources()  
 .useOpenSsl()

# User login once, perform mutlile activities and log out

Solution:

.exec(Login.*methodLogin*)  
  
 .during(60){  
 exec(http("T2 Navigate to base url")  
 .get("/")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("T3 Navigate to Vacation")  
 .get("/vacation.html")  
 .headers(*headers\_2*))  
 .pause(2)  
 }  
 .exec(Login.*methodLogout*)

# Report without ramp up and ramp down period

* Capture results only for steady state from simulation log file.
* exclude ramp up and ramp down filter and just show steady state

# Balck List HTML Resources

package blazedemo  
  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class BlazeDemoOneLoginLogoutMultipleOperation extends Simulation {  
  
 var *tokenExpr*="""<input type="hidden" name="\_token" value="(.\*)">"""  
 var *redirectingUrlAfterLoginPost*="""<title>Redirecting to (.\*)"""  
  
 *// <title>Redirecting to https://blazedemo.com/home</title>* val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 *//.disableFollowRedirect* .inferHtmlResources(BlackList(""".\*\.js""", """.\*\.css""", """.\*\.gif""", """.\*\.jpeg""", """.\*\.jpg""", """.\*\.ico""", """.\*\.woff""", """.\*\.woff2""", """.\*\.(t|o)tf""", """.\*\.png""", """.\*detectportal\.firefox\.com.\*"""), WhiteList())  
 .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9")  
 .acceptEncodingHeader("gzip, deflate")  
 .acceptLanguageHeader("en-GB,en;q=0.9,en-US;q=0.8,mr;q=0.7,hi;q=0.6")  
 .userAgentHeader("Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.109 Safari/537.36")  
  
 val *headers\_0* = *Map*(  
 "origin" -> "https://blazedemo.com",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *headers\_1* = *Map*(  
 "accept" -> "text/css,\*/\*;q=0.1",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "style",  
 "sec-fetch-mode" -> "no-cors",  
 "sec-fetch-site" -> "cross-site",  
 "x-client-data" -> "CJO2yQEIprbJAQjEtskBCKmdygEInvnLAQjnhMwBCJuczAEIrJzMAQ==")  
  
 val *headers\_2* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *uri2* = "https://fonts.googleapis.com/css"  
  
  
 object Login{  
 val *headers\_0* = *Map*(  
 "origin" -> "https://blazedemo.com",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *headers\_2* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 def methodLogin={  
 exec(flushHttpCache)  
 .exec(  
 http("T0 get Token from Login get call")  
 .get("/login")  
 .headers(*headers\_0*)  
 .check(regex(*tokenExpr*).saveAs("\_token"))  
  
 )  
 .pause(2)  
 .exec(http("T1 Login")  
 .post("/login")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token}")  
 .formParam("email", "zanjadkamlesh@gmail.com")  
 .formParam("password", "Testing@1")  
 .check(*status*.in(302,304, 200))  
 )  
 .pause(2)  
 }  
  
 def methodLogout ={  
 exec(http("T4 Navigate to Home")  
 .get("/home")  
 .headers(*headers\_2*)  
 .check(regex(*tokenExpr*).saveAs("\_token2"))  
 )  
 .pause(2)  
 .exec(http("T5 Logout")  
 .post("/logout")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token2}")  
 *//.formParam("\_token", "FLyQO3cggY4Is8OLbDRntffY8QkJzAQIUukYcfb1")* )  
 .pause(2)  
 .exec(flushHttpCache)  
 }  
 }  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 .exec(Login.*methodLogin*)  
  
 .during(60){  
 exec(http("T2 Navigate to base url")  
 .get("/")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("T3 Navigate to Vacation")  
 .get("/vacation.html")  
 .headers(*headers\_2*))  
 .pause(2)  
 }  
 .exec(Login.*methodLogout*)  
  
  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

duration 20 second, login 1, T2, T3 are multiple time, logout is once.

Table

Description automatically generated

60 seconds

Graphical user interface, table, treemap chart

Description automatically generated

# Infer html Resources and show in Report

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/src/test/scala/blazedemo/BlazeDemoShowResourcesInReport.scala

package blazedemo  
  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class BlazeDemoShowResourcesInReport extends Simulation {  
  
 var *tokenExpr*="""<input type="hidden" name="\_token" value="(.\*)">"""  
 var *redirectingUrlAfterLoginPost*="""<title>Redirecting to (.\*)"""  
  
 *// <title>Redirecting to https://blazedemo.com/home</title>* val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 .inferHtmlResources()  
 *//.nameInferredHtmlResourcesAfterAbsoluteUrl  
 //.disableFollowRedirect  
 //.inferHtmlResources(BlackList(""".\*\.js""", """.\*\.css""", """.\*\.gif""", """.\*\.jpeg""", """.\*\.jpg""", """.\*\.ico""", """.\*\.woff""", """.\*\.woff2""", """.\*\.(t|o)tf""", """.\*\.png""", """.\*detectportal\.firefox\.com.\*"""), WhiteList())* .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9")  
 .acceptEncodingHeader("gzip, deflate")  
 .acceptLanguageHeader("en-GB,en;q=0.9,en-US;q=0.8,mr;q=0.7,hi;q=0.6")  
 .userAgentHeader("Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.109 Safari/537.36")  
  
 val *headers\_0* = *Map*(  
 "origin" -> "https://blazedemo.com",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *headers\_1* = *Map*(  
 "accept" -> "text/css,\*/\*;q=0.1",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "style",  
 "sec-fetch-mode" -> "no-cors",  
 "sec-fetch-site" -> "cross-site",  
 "x-client-data" -> "CJO2yQEIprbJAQjEtskBCKmdygEInvnLAQjnhMwBCJuczAEIrJzMAQ==")  
  
 val *headers\_2* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *uri2* = "https://fonts.googleapis.com/css"  
  
  
 object Login{  
 val *headers\_0* = *Map*(  
 "origin" -> "https://blazedemo.com",  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 val *headers\_2* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1",  
 "upgrade-insecure-requests" -> "1")  
  
 def methodLogin={  
 exec(flushHttpCache)  
 .exec(  
 http.("T0 get Token from Login get call")  
 .get("/login")  
 .headers(*headers\_0*)  
 .check(regex(*tokenExpr*).saveAs("\_token"))  
  
 )  
 .pause(2)  
 .exec(http("T1 Login")  
 .post("/login")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token}")  
 .formParam("email", "zanjadkamlesh@gmail.com")  
 .formParam("password", "Testing@1")  
 .check(*status*.in(302,304, 200))  
 )  
 .pause(2)  
 }  
  
 def methodLogout ={  
 exec(http("T4 Navigate to Home")  
 .get("/home")  
 .headers(*headers\_2*)  
 .check(regex(*tokenExpr*).saveAs("\_token2"))  
 )  
 .pause(2)  
 .exec(http("T5 Logout")  
 .post("/logout")  
 .headers(*headers\_0*)  
 .formParam("\_token", "${\_token2}")  
 *//.formParam("\_token", "FLyQO3cggY4Is8OLbDRntffY8QkJzAQIUukYcfb1")* )  
 .pause(2)  
 .exec(flushHttpCache)  
 }  
 }  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 .exec(Login.*methodLogin*)  
  
 .during(20){  
 exec(http("T2 Navigate to base url")  
 .get("/")  
 .headers(*headers\_2*))  
 .pause(2)  
 .exec(http("T3 Navigate to Vacation")  
 .get("/vacation.html")  
 .headers(*headers\_2*))  
 .pause(2)  
 }  
 .exec(Login.*methodLogout*)  
  
  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

<file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/blazedemoshowresourcesinreport-20220227065206172/index.html>

Table

Description automatically generated

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/blazedemodonotshowresourcesinreport-20220227070322274/index.html

val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 .inferHtmlResources()  
 .silentResources

# Create Group for login and log out Download Resources but don’t show in Report

class BlazeDemoDonotShowResourcesInReport extends Simulation {

Table

Description automatically generated

Table

Description automatically generated

class BlazeDemoGroupShowResourcesInReport extends Simulation {  
  
 var *tokenExpr*="""<input type="hidden" name="\_token" value="(.\*)">"""  
 var *redirectingUrlAfterLoginPost*="""<title>Redirecting to (.\*)"""  
  
 *// <title>Redirecting to https://blazedemo.com/home</title>* val *httpProtocol* = http  
 .baseUrl("https://blazedemo.com")  
 .inferHtmlResources()

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/blazedemogroupshowresourcesinreport-20220227070739988/index.html

Calendar

Description automatically generated

# Gatling - exception handling

package blazedemo  
  
import io.gatling.core.Predef.\_  
import io.gatling.core.session  
import io.gatling.http.Predef.\_  
  
import scala.Predef.*->*class TryMaxExitOnFailure extends Simulation {  
 val *httpProtocol* = http  
 .baseUrl("https://jsonplaceholder.typicode.com")  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 tryMax(2){  
 exec(  
 http("T0 GetKEYVALUE")  
 .get("/todos/")  
 .check(*status*.is(session ->  
 200 + java.util.concurrent.ThreadLocalRandom.*current*().nextInt(2) *// 2* ))  
  
 )  
 }.exitHereIfFailed  
  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

package blazedemo  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class TryMaxExitOnFailure extends Simulation {  
 val *httpProtocol* = http  
 .baseUrl("https://jsonplaceholder.typicode.com")  
  
 val *scn* = scenario("BlazeDemoLoginLogout")  
 tryMax(2){  
 exec(  
 http("T0 GetKEYVALUE")  
 .get("/todos/")  
 .check(*status*.is(200+ scala.util.Random.nextInt(2)))  
 )  
 }.exitHereIfFailed  
  
 setUp(*scn*.inject(atOnceUsers(1))).protocols(*httpProtocol*)  
}

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/trymaxexception-20220227092533039/index.html

Graphical user interface, table

Description automatically generated

Increase random int rom 2 to 20

.check(*status*.is(200+ scala.util.Random.nextInt(2**0**)))

Graphical user interface

Description automatically generated

val *scn* = scenario("BlazeDemoLoginLogout")  
 .tryMax(5){  
 exec(  
 http("T0 GetKEYVALUE")  
 .get("/todos/")  
 .check(*status*.is(200+scala.util.Random.nextInt((20))))  
 )  
 }.exitHereIfFailed

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/trymaxexception-20220227093003181/index.html

All time only one value 217 as we have not use session variable

Graphical user interface, chart

Description automatically generated

val *scn* = scenario("BlazeDemoLoginLogout")  
 .tryMax(5){  
 exec(  
 http("T0 GetKEYVALUE")  
 .get("/todos/")  
 *//.check(status.is(200+scala.util.Random.nextInt((20)))) // all file times only one vale 217* .check(  
 *status*.is(  
 session=>200+scala.util.Random.nextInt((20)  
 )  
 )  
 )  
 )  
  
 }.exitHereIfFailed

Different session value are generated.

file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/trymaxexception-20220227093602346/index.html

Chart, table

Description automatically generated