**Build-In Predicate Factories**

Built-in Route Predicate Factories in Spring Cloud API Gateway

Below is a list of built-in predicates that you can use when creating routes in Spring Cloud API Gateways. These are built-in predicates and you do not need to write any additional Java code for them to work.

Note: The description of the following built-in predicates is taken from the [Spring Cloud API Gateway documentation page](https://cloud.spring.io/spring-cloud-gateway/reference/html/#gateway-request-predicates-factories).

**1. The After Route Predicate Factory**

The After route predicate factory takes one parameter, a datetime (which is a java ZonedDateTime). This predicate matches requests that happen after the specified datetime. The following example configures an after route predicate:

1. spring.cloud.gateway.routes[0].id = after\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = After=2017-01-20T17:42:47.789-07:00[America/Denver]

This route matches any request made after Jan 20, 2017 17:42 Mountain Time (Denver).

**2. The Before Route Predicate Factory**

The Before route predicate factory takes one parameter, a datetime (which is a java ZonedDateTime). This predicate matches requests that happen before the specified datetime. The following example configures a before route predicate:

1. spring.cloud.gateway.routes[0].id = before\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Before=2017-01-20T17:42:47.789-07:00[America/Denver]

This route matches any request made before Jan 20, 2017 17:42 Mountain Time (Denver).

**3. The Between Route Predicate Factory**

The Between route predicate factory takes two parameters, datetime1 and datetime2 which are java ZonedDateTimeobjects. This predicate matches requests that happen after datetime1 and before datetime2. The datetime2 parameter must be after datetime1. The following example configures a between route predicate:

1. spring.cloud.gateway.routes[0].id = between\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Between=2017-01-20T17:42:47.789-07:00[America/Denver], 2017-01-21T17:42:47.789-07:00[America/Denver]

This route matches any request made after Jan 20, 2017 17:42 Mountain Time (Denver) and before Jan 21, 2017 17:42 Mountain Time (Denver). This could be useful for maintenance windows.

**4. The Cookie Route Predicate Factory**

The Cookie route predicate factory takes two parameters, the cookie name and a regexp (which is a Java regular expression). This predicate matches cookies that have the given name and whose values match the regular expression. The following example configures a cookie route predicate factory:

1. spring.cloud.gateway.routes[0].id = cookie\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Cookie=chocolate, ch.p

This route matches requests that have a cookie named chocolate whose value matches the ch.p regular expression.

**5. The Header Route Predicate Factory**

The Header route predicate factory takes two parameters, the header name and a regexp (which is a Java regular expression). This predicate matches with a header that has the given name whose value matches the regular expression. The following example configures a header route predicate:

1. spring.cloud.gateway.routes[0].id = header\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Header=X-Request-Id, \d+

This route matches if the request has a header named X-Request-Id whose value matches the \d+ regular expression (that is, it has a value of one or more digits).

**6. The Host Route Predicate Factory**

The Host route predicate factory takes one parameter: a list of host name patterns. The pattern is an Ant-style pattern with . as the separator. This predicates matches the Host header that matches the pattern. The following example configures a host route predicate:

1. spring.cloud.gateway.routes[0].id = host\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Host=\*\*.somehost.org,\*\*.anotherhost.org

URI template variables (such as {sub}.myhost.org) are supported as well.

This route matches if the request has a Host header with a value of www.somehost.org or beta.somehost.org or www.anotherhost.org.

This predicate extracts the URI template variables (such as sub, defined in the preceding example) as a map of names and values and places it in the ServerWebExchange.getAttributes() with a key defined in ServerWebExchangeUtils.URI\_TEMPLATE\_VARIABLES\_ATTRIBUTE. Those values are then available for use by [GatewayFilterfactories](https://cloud.spring.io/spring-cloud-gateway/reference/html/#gateway-route-filters)

**7. The Method Route Predicate Factory**

The Method Route Predicate Factory takes a methods argument which is one or more parameters: the HTTP methods to match. The following example configures a method route predicate:

1. spring.cloud.gateway.routes[0].id = method\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Method=GET,POST

This route matches if the request method was a GET or a POST.

**8. The Path Route Predicate Factory**

The Path Route Predicate Factory takes two parameters: a list of Spring PathMatcher patterns and an optional flag called matchOptionalTrailingSeparator. The following example configures a path route predicate:

1. spring.cloud.gateway.routes[0].id = path\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Path=/red/{segment},/blue/{segment}

This route matches if the request path was, for example: /red/1 or /red/blue or /blue/green.

This predicate extracts the URI template variables (such as segment, defined in the preceding example) as a map of names and values and places it in the ServerWebExchange.getAttributes() with a key defined in ServerWebExchangeUtils.URI\_TEMPLATE\_VARIABLES\_ATTRIBUTE. Those values are then available for use by [GatewayFilterfactories](https://cloud.spring.io/spring-cloud-gateway/reference/html/#gateway-route-filters)

**9. The Query Route Predicate Factory**

The Query route predicate factory takes two parameters: a required param and an optional regexp (which is a Java regular expression). The following example configures a query route predicate:

1. spring.cloud.gateway.routes[0].id = query\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Query=green

The preceding route matches if the request contained a green query parameter.

**10. The RemoteAddr Route Predicate Factory**

The RemoteAddr route predicate factory takes a list (min size 1) of sources, which are CIDR-notation (IPv4 or IPv6) strings, such as 192.168.0.1/16 (where 192.168.0.1 is an IP address and 16 is a subnet mask). The following example configures a RemoteAddr route predicate:

1. spring.cloud.gateway.routes[0].id = remoteaddress\_route
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = RemoteAddr=192.168.1.1/24

This route matches if the remote address of the request was, for example, 192.168.1.10.

**11. The Weight Route Predicate Factory**

The Weight route predicate factory takes two arguments: group and weight (an int). The weights are calculated per group. The following example configures a weight route predicate:

1. spring.cloud.gateway.routes[0].id = weight\_high
2. spring.cloud.gateway.routes[0].uri = https://example.org
3. spring.cloud.gateway.routes[0].predicates[0] = Weight=group1, 8
5. spring.cloud.gateway.routes[1].id = weight\_low
6. spring.cloud.gateway.routes[1].uri = https://example.org
7. spring.cloud.gateway.routes[1].predicates[0] = Weight=group1, 2

This route would forward ~80% of traffic to [weighthigh.org](https://weighthigh.org/) and ~20% of traffic to [weighlow.org](https://weighlow.org/)