



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Mobile Services API Reference

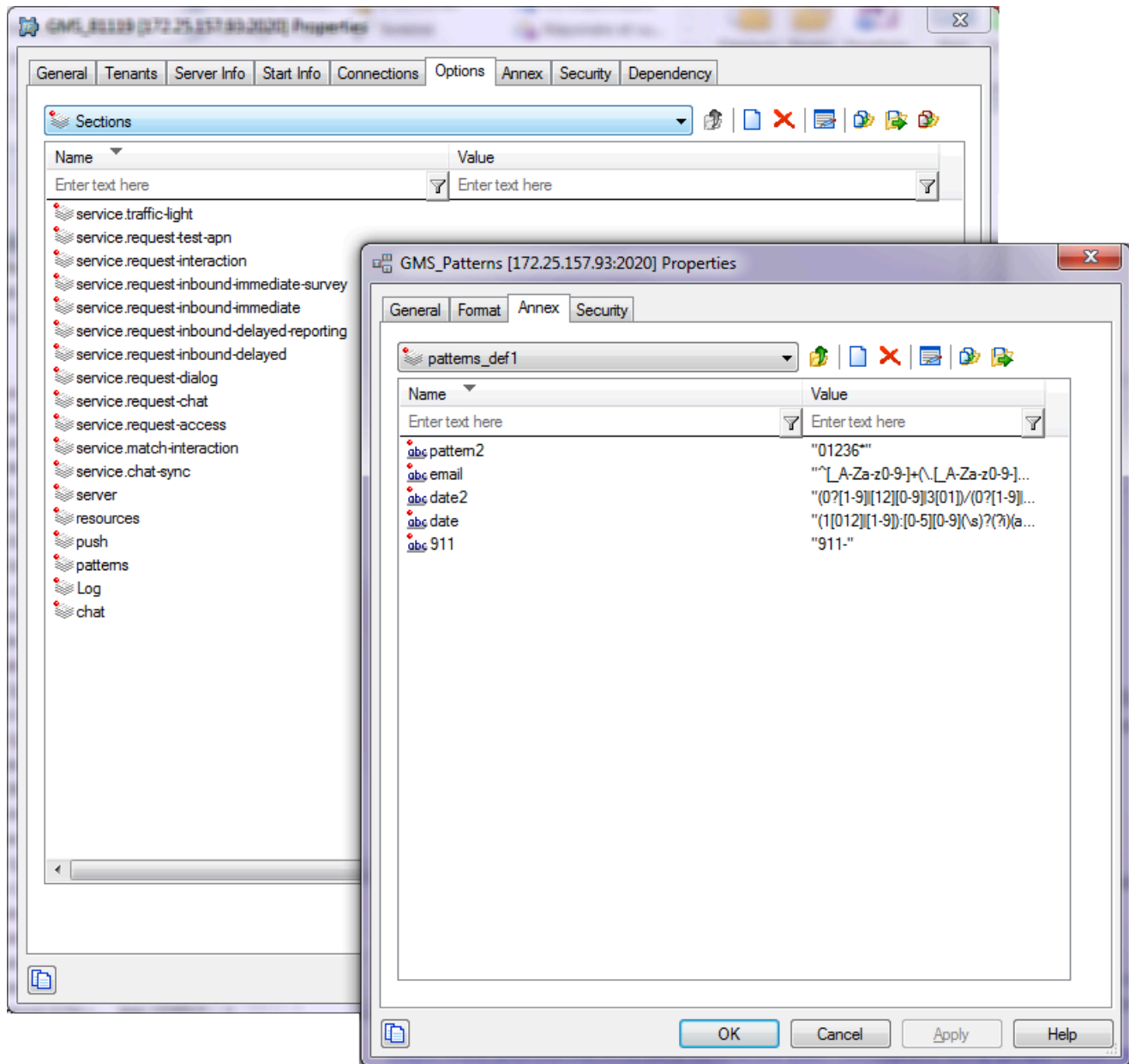
Pattern Matcher API

Contents

- [1 Pattern Matcher API](#)
 - [1.1 Getting Started](#)
 - [1.2 List of API Queries](#)
 - [1.3 Sample Errors](#)

Pattern Matcher API

Getting Started



The Pattern Matcher API allows you to create and manage pattern lists that you can use to check parameter values and define exceptions in your GMS service.

First, configure a list of patterns or groups of list of patterns in your [GMS Configuration object](#) or per service, in the GMS Service Management UI as detailed in the [Callback Solution Guide](#).

For example, the left-side screenshot shows the GMS configuration of the `patterns_def1` group .

Once you have defined some patterns, use the Pattern Matcher API queries to check the validity of your parameters. Each query will return the name of the matching pattern (if any) for each parameter. As a result, if none of the parameters match the patterns, the response will be an empty array.

For example, if you submit the following parameters:

```
param1 = 1234562
param2 = john.doe@gmail.com
param3 = 911
```

The API result would be:

```
"param3": "911", "param2": "email"
```

Pattern Format

The exception patterns are regular expressions as defined in the [Java Pattern Class](#).

For the `_customer_number` parameter used in Callback, note that the system internally adds a + sign to the phone number by default. As a result, if `_customer_number=12345678901`, it will become `_customer_number=+12345678901` in the system before going through the processing of the patterns exception. So, to define patterns exception for specific phone numbers, you need to add the + sign at the start of your pattern exception, for example:

```
[cb_errors]
12345678901=\+12345678901
```

List of API Queries

- [POST genesys/1/patterns](#): Verify parameters against general patterns list.
- [POST genesys/1/patterns/group/{groupName}](#): Verify parameters against a specific group in the patterns list.

Verify Parameters Against General Patterns List

Use this query to submit a list of parameters to verify. The method returns a JSON array of the parameters that match one of the patterns, where: the keys are the parameters and the values are the name of the matching pattern in the general pattern list. Only strings that match one of the patterns are returned; others are ignored. As a result, if none of the parameters match the patterns, the response will be an empty array.

Operation

POST genesys/1/patterns

Body: The body can be either a MultiPart form or x-www-form-urlencoded form that consists of key/value pairs representing the strings to test.

For example: param1=<string to check>¶m2=<string to check>&...¶m-n=<string to check>

Response

HTTP code	200
HTTP Message	OK
Body	A JSON array of key-value pairs where the key is the parameter name and the value is the name of the matching pattern. For example: {"param1": "pattern1", "param2": "pattern2", ..., "param-n": "pattern-n"} where pattern-i is the name of a pattern.

Example

The following example verifies param1, param2, and param3.

```
POST http://127.0.0.1:8080/genesys/1/patterns HTTP/1.1
Host: 127.0.0.1:8080
Connection: keep-alive
Content-Length: 347
Cache-Control: no-cache
Origin: chrome-extension://fdmmgilgnpjigdojojpjoooidkmcomcm
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.22
(KHTML, like Gecko) Chrome/25.0.1364.152 Safari/537.22
Content-Type: multipart/form-data;
boundary=---WebKitFormBoundaryTkdw0u7LG1bBGbnj
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,fr;q=0.6,fr-FR;q=0.4
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
-----WebKitFormBoundaryTkdw0u7LG1bBGbnj
Content-Disposition: form-data; name="param1"
12
-----WebKitFormBoundaryTkdw0u7LG1bBGbnj
Content-Disposition: form-data; name="param2"
john.doe@gmail.com
-----WebKitFormBoundaryTkdw0u7LG1bBGbnj
Content-Disposition: form-data; name="param3"
911-
-----WebKitFormBoundaryTkdw0u7LG1bBGbnj--
```

The following response indicates that **param2** and **param3** match some patterns defined in Configuration Manager.

```
HTTP/1.1 200 OK
Date: Thu, 14 Mar 2013 16:13:06 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json; charset=UTF-8
```

Content-Length: 44

```
{"param3": "911", "param2": "email"}
```

Verify Parameters Against a Specific Group in the Patterns List

Use this query to submit a list of parameters to verify against the patterns defined in a specific group that is part of the general pattern list. The method returns a JSON array of the parameters that match one of the patterns of this group, where the keys are the parameters and the values are the name of the matching pattern in the group pattern list. Only strings that match one of the patterns are returned; others are ignored. As a result, if none of the parameters match the patterns, the response will be an empty array.

Operation

POST genesys/1/patterns/group/{groupName}			
Parameter	Type	Mandatory	Description
URI Parameters			
{groupName}	String	Yes	The group to which the patterns belong.
<p>Body: The body can be either a MultiPart form or x-www-form-urlencoded form consisting of key/value pairs, representing the strings to test.</p> <p>For example:</p> <pre>param1=<string to check>¶m2=<string to check>&...¶m-n=<string to check></pre>			

Response

HTTP code	200
HTTP Message	OK
Body	A JSON array of key-value pairs where the key is the parameter name and the value is the pattern found for this parameter. {"param1": "pattern1", "param2": "pattern2", ..., "param-n": "pattern-n"} where pattern-i is the name of a pattern.

Example

The following example verifies if the param1 and param2 match the group of patterns "patterns_def1".

Operation

```
POST http://127.0.0.1:8080/genesys/1/patterns/group/patterns_def1
HTTP/1.1
Host: 127.0.0.1:8080
Connection: keep-alive
Content-Length: 41
```

```
Cache-Control: no-cache
Origin: chrome-extension://fdmmgilgnpjigdojojpjoooidkmcomcm
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
  AppleWebKit/537.22 (KHTML, like Gecko)
Chrome/25.0.1364.152 Safari/537.22
Content-Type: application/x-www-form-urlencoded
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,fr;q=0.6,fr-FR;q=0.4
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
param1=john.doe%40gmail.com¶m2=blabla
```

Result

```
HTTP/1.1 200 OK
Date: Thu, 14 Mar 2013 16:22:30 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Type: application/json;charset=UTF-8
Content-Length: 30
{"param1":"email"}
```

The following response indicates that param1 matches the email pattern of the **patterns_def1** group defined in Configuration Manager.

Sample Errors

Your application can receive an HTTP error 403 Forbidden if your service and your GMS configuration do not include the required patterns or group of patterns.

```
HTTP/1.1 403 Forbidden
Date: Thu, 14 Mar 2013 16:23:35 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Length: 107
{"message":"Group (patterns_def1s) unknown.",
"exception":"com.genesyslab.gsg.services.pattern.MatchPattern"}
```

```
HTTP/1.1 403 Forbidden
Date: Thu, 14 Mar 2013 16:24:32 GMT
Pragma: no-cache
Cache-Control: no-cache
Cache-Control: no-store
Content-Length: 142
{"message":"Service (match-interaction) not configured
to support exceptions.",
"exception":"com.genesyslab.gsg.services.pattern.MatchPattern"}
```