Webhooks

About Webhooks

Webhooks are ways of integrating Jiwa with other 3rd party API's. When an event in Jiwa occurs (such as when a product is created), a 3rd party API can be notified instantly using webhooks.

Webhooks eliminate the need for polling, which can be inefficient and introduce latency between the event occurrence and when an action in response to that event occurs.

Implementation

Webhooks in Jiwa are implemented by augmenting the existing REST API with webhook functionality via a plugin. It is based on a **subscriber** / **subscription** model.

> **Webhooks requires Jiwa 07.02.01.00 or later**
> Versions earlier than Jiwa 07.02.01 used a plugin to extend the REST API to add functionality. This is no longer supported and if Webhooks are required then you must update to Jiwa 07.02.01 or later.

Subscribers are defined in Jiwa, and a subscriber once given their **SubscriberID** can register a **subscription** to the possible webhooks available in Jiwa. When events occur in Jiwa, a **message** is generated and sent to the webhook **subscriptions**. The result of that message is then stored in a **Message response**.

Messages which fail to be sent to a subscriber are queued for retry based on some system settings.

Subscribers can also remove their own subscriptions and inspect what messages have been sent or attempted to be sent and the current status of the message.

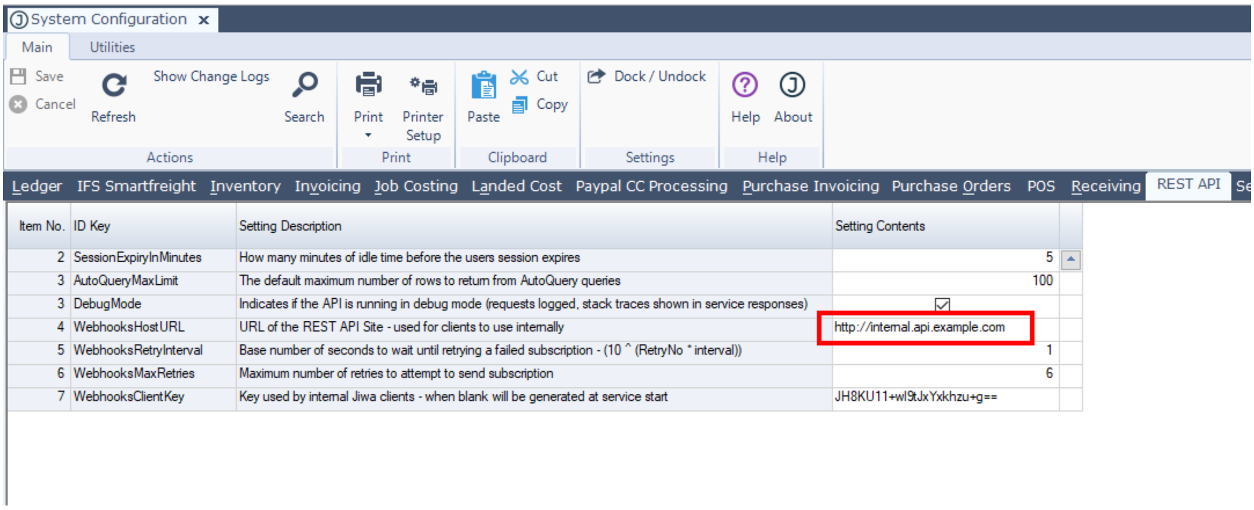
All webhook messages are sent as a POST operation with any relevant document DTO as the body.

Enabling Webhooks

Webhooks are enabled when the REST API plugin is enabled, and either the Self Hosted service or IIS are configured and running.

An additional step is to configure the WebhooksHostURL system setting of the REST API Plugin. This can be done via the System Configuration form, on the REST API tab.

The value should be the URL of your REST API reachable by internal Jiwa users. For webhooks to function, each Jiwa client will POST events to this internal URL and the service will then forward those messages to subscribers.



Item No.	ID Key	Setting Description	Setting Contents
2	SessionExpiryInMinutes	How many minutes of idle time before the users session expires	5
3	AutoQueryMaxLimit	The default maximum number of rows to return from AutoQuery queries	100
3	DebugMode	Indicates if the API is running in debug mode (requests logged, stack traces shown in service responses)	☑
4	WebhooksHostURL	URL of the REST API Site - used for clients to use internally	http://internal.api.example.com
5	WebhooksRetryInterval	Base number of seconds to wait until retrying a failed subscription - (10 ^ (RetryNo * interval))	1
6	WebhooksMaxRetries	Maximum number of retries to attempt to send subscription	6
7	WebhooksClientKey	Key used by internal Jiwa clients - when blank will be generated at service start	JH8KU11+wI9tJxYxkhzu+g==

NOTE: In most circumstances the value for the WebhooksHostURL should **not** be http(s)://localhost or http(s)://127.0.0.1 - this must be the address your Jiwa users can reach via a HTTP POST. If all your users are inside the firewall this can be the machine DNS name or local IP Address. Check if the WebhooksHostURL is correct by putting the value of WebhooksHostURL in a browser address bar on all the Jiwa client machines - the REST API metadata page should appear.

Operations for use by Subscribers

The following is a list of operations subscribers can use to manage their subscriptions and related messages and message responses. Note no authentication is required.

List Webhook Events

List all the published events a subscriber can subscribe to

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var eventsListRequest = new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksEventsGETRequ
est() {  };
List<WebHookEvent> eventsListResponse =
client.Get(eventsListRequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{
    responsebody =
webClient.DownloadString("https://api.jiwa.com.au/Webhooks
/Events");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X GET
https://api.jiwa.com.au/Webhooks/Events
```

Web Browser

```
https://api.jiwa.com.au/Webhooks/Events?format=json
```

Note the ?**`format=json`** in the above URL this overrides the content type returned. For browsers the default content type is HTML - if a content type override is omitted, then a HTML razor view of the data will be returned instead of json. xml and csv are also valid overrides for the content type to be returned.

Example Response:

```
[
 {
  "Name":"debtor.created",
  "Description":"Occurs when a new debtor (customer) is
created"
 },
 {
  "Name":"debtor.deleted",
  "Description":"Occurs when a debtor (customer) is deleted"
 }
]
```

The possible events that are published by default are listed in the following table. A plugin can add more events as required.

Event	Description
bookin.created	Occurs when a new book in is created
bookin.updated	Occurs when a book in is modified
creditor.created	Occurs when a new creditor (supplier) is created
creditor.deleted	Occurs when a creditor (supplier) is deleted
creditor.updated	Occurs when a creditor (supplier) is modified
creditorclassification.created	Occurs when a new creditor classification is created
creditorclassification.deleted	Occurs when a creditor classification is deleted
creditorclassification.updated	Occurs when a creditor classification is modified
debtor.created	Occurs when a new debtor (customer) is created
debtor.deleted	Occurs when a debtor (customer) is deleted
debtor.updated	Occurs when a debtor (customer) is modified
debtorcategory.created	Occurs when a new debtor category is created
debtorcategory.deleted	Occurs when a debtor category is deleted
debtorcategory.updated	Occurs when a debtor category is modified
debtorclassification.created	Occurs when a new debtor classification is created
debtorclassification.deleted	Occurs when a debtor classification is deleted
debtorclassification.updated	Occurs when a debtor classification is modified
goodsreceivednote.created	Occurs when a new goods received note is created
goodsreceivednote.updated	Occurs when a goods received note is modified
inventory.created	Occurs when a new inventory item (product) is created
inventory.deleted	Occurs when an inventory item (product) is deleted
inventory.updated	Occurs when an inventory item (product) is modified
inventory.stocklevel	Occurs when an inventory item stock level changes
inventorycategory.created	Occurs when a new inventory category is created
inventorycategory.deleted	Occurs when an inventory category is deleted
inventorycategory.updated	Occurs when an inventory category is modified

inventoryclassification.created	Occurs when a new inventory classification is created
inventoryclassification.deleted	Occurs when an inventory classification is deleted
inventoryclassification.updated	Occurs when an inventory classification is modified
purchaseorder.created	Occurs when a new purchase order is created
purchaseorder.deleted	Occurs when a purchase order is deleted
purchaseorder.updated	Occurs when a purchase order is modified
salesorder.created	Occurs when a new sales order is created
salesorder.updated	Occurs when a sales order is modified
salesquote.created	Occurs when a new sales quote is created
salesquote.updated	Occurs when a sales quote is modified
shipment.created	Occurs when a new shipment is created
shipment.updated	Occurs when a shipment is modified
warehousetransferin.created	Occurs when a new warehouse transfer in is created
warehousetransferin.updated	Occurs when a warehouse transfer in is modified
warehousetransferout.created	Occurs when a new warehouse transfer out is created
warehousetransferout.updated	Occurs when a warehouse transfer out is modified

Add a new Subscriber

Subscribers are not intended to be added by external source, and as such require authentication - see Authenticating under Consuming the REST API

Once a subscriber is added, the SubscriberID (RecID returned by the POST to /Webhooks/Subscribers) is the unique identifier you would perhaps provide to external to allow them to manage their own subscriptions.

Add a new subscriber

Add a new subscriber

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");
var authResponse = client.Get(new
ServiceStack.Authenticate() { UserName = "admin", Password
= "password" });

var webhooksSubscriptionsPOSTRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscribersPO
STRequest { Name = "My Test Subscriber", IsEnabled = true
};
JiwaFinancials.Jiwa.JiwaServiceModel.SY_WebhookSubscription
WebhooksSubscribersPOSTResponse =
client.Post(WebhooksSubscribersPOSTRequest );
```

C#

```
using (var webClient = new System.Net.WebClient())
{
 string json =
Newtonsoft.Json.JsonConvert.SerializeObject(new
     {
         Name = "My Test Subscriber",
         IsEnabled = true
        });
    responsebody =
webClient.UploadString("https://api.jiwa.com.au/Webhooks/S
ubscribers/", "POST", json);
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X POST
https://api.jiwa.com.au/Webhooks/Subscribers/?Name="My Test
Subscriber"&IsEnabled=true
```

Instead of using URL parameters as above, you can also use a DTO to set the parameters:

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X POST
https://api.jiwa.com.au/Webhooks/Subscribers/ -d
'{"Name":"My Test Subscriber","IsEnabled":"true"}'
```

Add a new Subscription (Subscribe to a webhook event)

Add a new subscription

Add a subscription for Subscriber with ID b25a2922-931b-4447-9160-3984b91c02f4 - when a sales order is created, perform a POST operation on the URL https://example.com/api/dosomething

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");
var authResponse = client.Get(new
ServiceStack.Authenticate() { UserName = "admin", Password
= "password" });

var webhooksSubscriptionsPOSTRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscriptions
POSTRequest{ SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4", URL =
"https://example.com/api/dosomething", EventName =
"salesorder.created" };
JiwaFinancials.Jiwa.JiwaServiceModel.SY_WebhookSubscription
webhooksSubscriptionsPOSTResponse =
client.Post(webhooksSubscriptionsPOSTRequest);
```

Some subscribers may wish for one or more headers to be provided. Often API's require an API Key to be provided in the request header. The Headers property of the DTO can optionally be provided which is a list of name value pairs of request headers. When provided when defining a subscription, all messages for that subscription are sent with the request headers set.

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");
var authResponse = client.Get(new
ServiceStack.Authenticate() { UserName = "admin", Password
= "password" });

var webhooksSubscriptionsPOSTRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscriptions
POSTRequest{ SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4", URL =
"https://example.com/api/dosomething", EventName =
"salesorder.created", Headers = new
List<JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscrip
tionHeader> { new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscriptionH
eader { Name = "ApiKey", Value = "AAABBBCCC"} } };
JiwaFinancials.Jiwa.JiwaServiceModel.SY_WebhookSubscription
webhooksSubscriptionsPOSTResponse =
client.Post(webhooksSubscriptionsPOSTRequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{
 string json =
Newtonsoft.Json.JsonConvert.SerializeObject(new
     {
         SubscriberID =
"2a84b900-d178-4de4-8d11-18b318c0276b",
         URL = "https://example.com/api/dosomething",
         EventName = "salesorder.created"
        });
    responsebody =
webClient.UploadString("https://api.jiwa.com.au/Webhooks/S
ubscribers/b25a2922-931b-4447-9160-3984b91c02f4/Subscripti
ons/", "POST", json);
}
```

Some subscribers may wish for one or more headers to be provided. Often API's require an API Key to be provided in the request header. The Headers property of the DTO can optionally be provided which is a list of name value pairs of request headers. When provided when defining a subscription, all messages for that subscription are sent with the request headers set.

```
using (var webClient = new System.Net.WebClient())
{
 string json =
Newtonsoft.Json.JsonConvert.SerializeObject(new
     {
         SubscriberID =
"2a84b900-d178-4de4-8d11-18b318c0276b",
         URL = "https://example.com/api/dosomething",
         EventName = "salesorder.created",
            Headers = new List<object>() { new { Name =
"ApiKey", Value = "AAABBBCCC" } }
        });
    responsebody =
webClient.UploadString("https://api.jiwa.com.au/Webhooks/S
ubscribers/b25a2922-931b-4447-9160-3984b91c02f4/Subscripti
ons/", "POST", json);
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X POST
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/?URL="https://exampl
e.com/api/dosomething"&EventName="salesorder.created"
```

Instead of using URL parameters as above, you can also use a DTO to set the parameters:

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X POST
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/ -d
'{"URL":"https://example.com/api/dosomething","EventName":
"salesorder.created"}'
```

Some subscribers may wish for one or more headers to be provided. Often API's require an API Key to be provided in the request header. The Headers property of the DTO can optionally be provided which is a list of name value pairs of request headers. When provided when defining a subscription, all messages for that subscription are sent with the request headers set.

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X POST
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/ -d
'{"URL":"https://example.com/api/dosomething","EventName":
"salesorder.created","Headers":[{"Name":"ApiKey","Value":"
AAABBBCCC"}]}'
```

List all subscriptions for a Subscriber

Lists all subscriptions for a subscriber

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var webhooksSubscriptionsGETRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscriptions
GETRequest() { SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4" };
List<SY_WebhookSubscription>
webhooksSubscriptionsGETResponse =
client.Get(webhooksSubscriptionsGETRequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{
    responsebody =
webClient.DownloadString("https://api.jiwa.com.au/Webhooks
/Subscribers/b25a2922-931b-4447-9160-3984b91c02f4/Subscrip
tions/");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X GET
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/
```

Web Browser

```
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/?format=json
```

Note the ?**`format=json`** in the above URL this overrides the content type returned. For browsers the default content type is HTML - if a content type override is omitted, then a HTML razor view of the data will be returned instead of json. xml and csv are also valid overrides for the content type to be returned.

Example Response:

```
[
 {
  "RecID":"2a84b900-d178-4de4-8d11-18b318c0276b",

"SY_WebhookSubscriber_RecID":"b25a2922-931b-4447-9160-3984b91
c02f4",
  "EventName":"salesorder.created",
  "URL":"https://example.com/api/dosomething",
  "ItemNo":1,
  "LastSavedDateTime":"\/Date(1511400032893-0000)\/",
  "RowHash":"AAAAAAAAmns="
 }
]
```

Delete a Subscribers Subscription

Deletes an existing subscription

Given Subscriber "b25a2922-931b-4447-9160-3984b91c02f4" has an existing subscription with ID "2a84b900-d178-4de4-8d11-18b318c0276b", delete it

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var webhooksSubscriptionsDELETERequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksSubscriptions
DELETERequest { SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4", SubscriptionID =
"2a84b900-d178-4de4-8d11-18b318c0276b" };
client.Delete(WebhooksSubscriptionsDELETERequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{

webClient.Headers[System.Net.HttpRequestHeader.ContentType]
= "application/json";
 responsebody =
webClient.UploadString("https://api.jiwa.com.au/Webhooks/S
ubscribers/b25a2922-931b-4447-9160-3984b91c02f4/Subscripti
ons/2a84b900-d178-4de4-8d11-18b318c0276b", "DELETE", "");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X DELETE
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/2a84b900-d178-4de4-8
d11-18b318c0276b
```

Messages

This is a queryable request, meaning filtering, pagination, ordering and limiting what fields are returned is possible through either URL parameters or DTO property values

List all Messages for a subscriber

This is a queryable request, meaning filtering, pagination, ordering and limiting what fields are returned is possible through either URL parameters or DTO property values.

Lists all messages for a subscriber

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var webhooksMessagesGETRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksMessagesGETRe
quest() { SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4" };
QueryDb<v_SY_WebhookSubscriber_Messages>
webhooksMessagesGETResponse =
client.Get(webhooksMessagesGETRequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{
    responsebody =
webClient.DownloadString("https://api.jiwa.com.au/Webhooks
/Subscribers/b25a2922-931b-4447-9160-3984b91c02f4/Messages
/");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X GET
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Messages/
```

Web Browser

```
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Messages/?format=json
```

Note the ?**`format=json`** in the above URL this overrides the content type returned. For browsers the default content type is HTML - if a content type override is omitted, then a HTML razor view of the data will be returned instead of json. xml and csv are also valid overrides for the content type to be returned.

Example Response:

```
{
 "Results" : [{
   "SubscriberID" : "b25a2922-931b-4447-9160-3984b91c02f4",
   "SubscriptionID" : "2a84b900-d178-4de4-8d11-18b318c0276b",
   "MessageID" : "7d00f575-1159-49b4-bdd5-5b560d2dcd21",
   "EventName" : "salesorder.created",
   "URL" : "https://example.com/api/dosomething",
   "Body" : "DTO Json would be in here",
   "ItemNo" : 3,
   "Status" : 2,
   "Retries" : 6,
   "AddedDateTime" : "\/Date(1511372206197-0000)\/",
   "LastSavedDateTime" : "\/Date(1511694312630-0000)\/",
            "LastMessageResponseHTTPCode" : 404
   "LastMessageResponseMessage" : "The remote name could not
be resolved: 'example.com'"
  }
 ],
 "Meta" : {}
}
```

Filtered, Curated List of Messages for a subscriber

Lists first 10 messages for a subscriber where the status is 2 (failed pending retry), but limit fields returned and order by #retries

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var webhooksMessagesGETRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksMessagesGETRe
quest() { SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4", Take = 10, Status =
2, Fields="MessageID,EventName,URL,Retries",
Orderby=Retries };
QueryDb<v_SY_WebhookSubscriber_Messages>
webhooksMessagesGETResponse =
client.Get(webhooksMessagesGETRequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{
    responsebody =
webClient.DownloadString("https://api.jiwa.com.au/Webhooks
/Subscribers/b25a2922-931b-4447-9160-3984b91c02f4/Messages
/?Take=10&Status=2&Fields=MessageID,EventName,URL,Retries,
Orderby=Retries");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X GET
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Messages/?Take=10&Status=2&Fields=
MessageID,EventName,URL,Retries,Orderby=Retries
```

Web Browser

```
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Messages/?Take=10&Status=2&Fields=
MessageID,EventName,URL,Retries,Orderby=Retries&format=json
```

Note the &**`format=json`** in the above URL this overrides the content type returned. For browsers the default content type is HTML - if a content type override is omitted, then a HTML razor view of the data will be returned instead of json. xml and csv are also valid overrides for the content type to be returned.

Example Response:

```
{
 "Results" : [{
   "EventName" : "salesorder.created",
   "URL" : "https://example.com/api/dosomething",
   "Retries" : 6,
  }
 ],
 "Meta" : {}
}
```

Message Statuses

Status Value	Description
0	Not sent
1	Successful
2	Failed, Retry Pending
3	Failed

Delete a Message

Deletes an existing message

Given Subscriber "b25a2922-931b-4447-9160-3984b91c02f4" has an existing subscription with ID "2a84b900-d178-4de4-8d11-18b318c0276b" which in turn has a message with ID "7d00f575-1159-49b4-bdd5-5b560d2dcd21", delete it

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var webhooksMessagesDELETERequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksMessagesDELET
ERequest{ SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4", SubscriptionID =
"2a84b900-d178-4de4-8d11-18b318c0276b", MessageID =
"7d00f575-1159-49b4-bdd5-5b560d2dcd21" };
client.Delete(webhooksMessagesDELETERequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{

webClient.Headers[System.Net.HttpRequestHeader.ContentType]
= "application/json";
 responsebody =
webClient.UploadString("https://api.jiwa.com.au/Webhooks/S
ubscribers/b25a2922-931b-4447-9160-3984b91c02f4/Subscripti
ons/2a84b900-d178-4de4-8d11-18b318c0276b/Messages/7d00f575
-1159-49b4-bdd5-5b560d2dcd21", "DELETE", "");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X DELETE
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b
-4447-9160-3984b91c02f4/Subscriptions/2a84b900-d178-4de4-8
d11-18b318c0276b/Messages/7d00f575-1159-49b4-bdd5-5b560d2d
cd21
```

List all Message Responses for a Subscriber

This is a queryable request, meaning filtering, pagination, ordering and limiting what fields are returned is possible through either URL parameters or DTO property values.

Lists first 10 message responses for a subscriber where the HTTP Response code is 404

ServiceStack Client C#

```
var client = new
ServiceStack.JsonServiceClient("https://api.jiwa.com.au");

var webhooksMessageResponsesGETRequest= new
JiwaFinancials.Jiwa.JiwaServiceModel.WebhooksMessageResponsesGETRequest() { SubscriberID =
"b25a2922-931b-4447-9160-3984b91c02f4", Take = 10, HTTPCode
= 404 };
QueryDb<v_SY_WebhookSubscriber_Messages>
webhooksMessageResponsesGETResponse=
client.Get(webhooksMessageResponsesGETRequest);
```

C#

```
using (var webClient = new System.Net.WebClient())
{
    responsebody =
webClient.DownloadString("https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b-4447-9160-3984b91c02f4/Messages/Responses/?Take=10&HTTPCode=404");
}
```

Curl

```
curl -H 'Accept: application/json' -H 'Content-Type:
application/json' -X GET
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b-4447-9160-3984b91c02f4/Messages/Responses/?Take=10&HTTPCode=404
```

Web Browser

```
https://api.jiwa.com.au/Webhooks/Subscribers/b25a2922-931b-4447-9160-3984b91c02f4/Messages/Responses/?Take=10&HTTPCode=404&format=json
```

Note the &`format=json` in the above URL this overrides the content type returned. For browsers the default content type is HTML - if a content type override is omitted, then a HTML razor view of the data will be returned instead of json. xml and csv are also valid overrides for the content type to be returned.

Example Response:

```
{
 "Results" : [{
   "SubscriberID" : "b25a2922-931b-4447-9160-3984b91c02f4",
   "SubscriptionID" : "2a84b900-d178-4de4-8d11-18b318c0276b",
   "MessageID" : "7d00f575-1159-49b4-bdd5-5b560d2dcd21",
   "MessageResponseID" :
"9c788af7-697a-4d10-8241-1575b4000384",
   "EventName" : "salesorder.created",
   "URL" : "https://example.com/api/dosomething",
   "Body" : "Body DTO In here",
   "MessageItemNo" : 3,
   "Status" : 2,
   "Retries" : 6,
   "AddedDateTime" : "\/Date(1511372206197-0000)\/",
   "MessageLastSavedDateTime" :
"\/Date(1511694312630-0000)\/",
   "HTTPCode" : 404,
   "Message" : "The remote server returned an error: (404)
Not Found.",
   "ItemNo" : 4,
   "LastSavedDateTime" : "\/Date(1511372321263-0000)\/"
  }, {
   "SubscriberID" : "b25a2922-931b-4447-9160-3984b91c02f4",
   "SubscriptionID" : "2a84b900-d178-4de4-8d11-18b318c0276b",
   "MessageID" : "7d00f575-1159-49b4-bdd5-5b560d2dcd21",
   "MessageResponseID" :
"805c9edd-b807-4123-a3fd-1ce3f5b403dd",
   "EventName" : "salesorder.created",
   "URL" : "https://example.com/api/dosomething",
   "Body" : "Body DTO In here",
   "MessageItemNo" : 3,
   "Status" : 2,
   "Retries" : 6,
   "AddedDateTime" : "\/Date(1511372206197-0000)\/",
   "MessageLastSavedDateTime" :
"\/Date(1511694312630-0000)\/",
   "HTTPCode" : 404,
   "Message" : "The remote server returned an error: (404)
Not Found.",
   "ItemNo" : 3,
   "LastSavedDateTime" : "\/Date(1511372220007-0000)\/"
  }
 ],
 "Meta" : {}
}
```

Message Retries & Resilience

Webhook messages are sent to subscribers as they occur in Jiwa immediately and asynchronously - meaning it happens in the background and the time taken to send the message does not delay or impact users of Jiwa.

If a message should fail, then it is retried based on a time schedule. All messages are sent by the REST API service, not the Jiwa clients themselves - so the Jiwa client that originally generated the webhook event does not need to remain powered on.

Messages are persisted to a SQL Table SY_WebhookMessage, and that table is read when the REST API Service starts and unsent messages are queued for delivery. By default messages are retried after 1 second, then 10 seconds, 100 seconds, and so on until after the 6th retry the message if marked as failed (Status 3) and no longer retried.

System settings under the "REST API Webhooks" tab of the system configuration form control how long the retry interval is, and the maximum number of retries to attempt.

This strategy of persisting the messages to an SQL table and retrying delivery of failed messages at growing intervals provides the resilience required to integrate with other API's

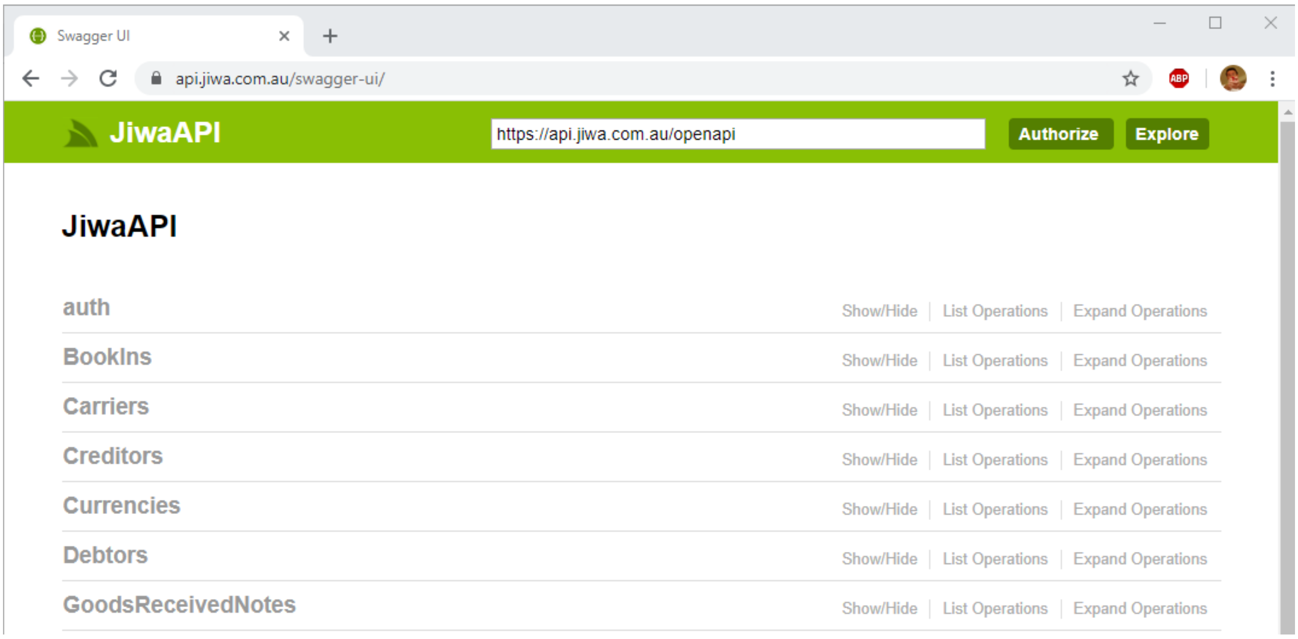
---

Tutorial - Using SwaggerUI to add a subscription

In this step by step example, we show how to add a subscription to the inventory.stocklevel webhook, so that a http endpoint https://example.com/api/dosomething is invoked whenever the stock level for a product changes.

Step 1 - Visit the SwaggerUI page

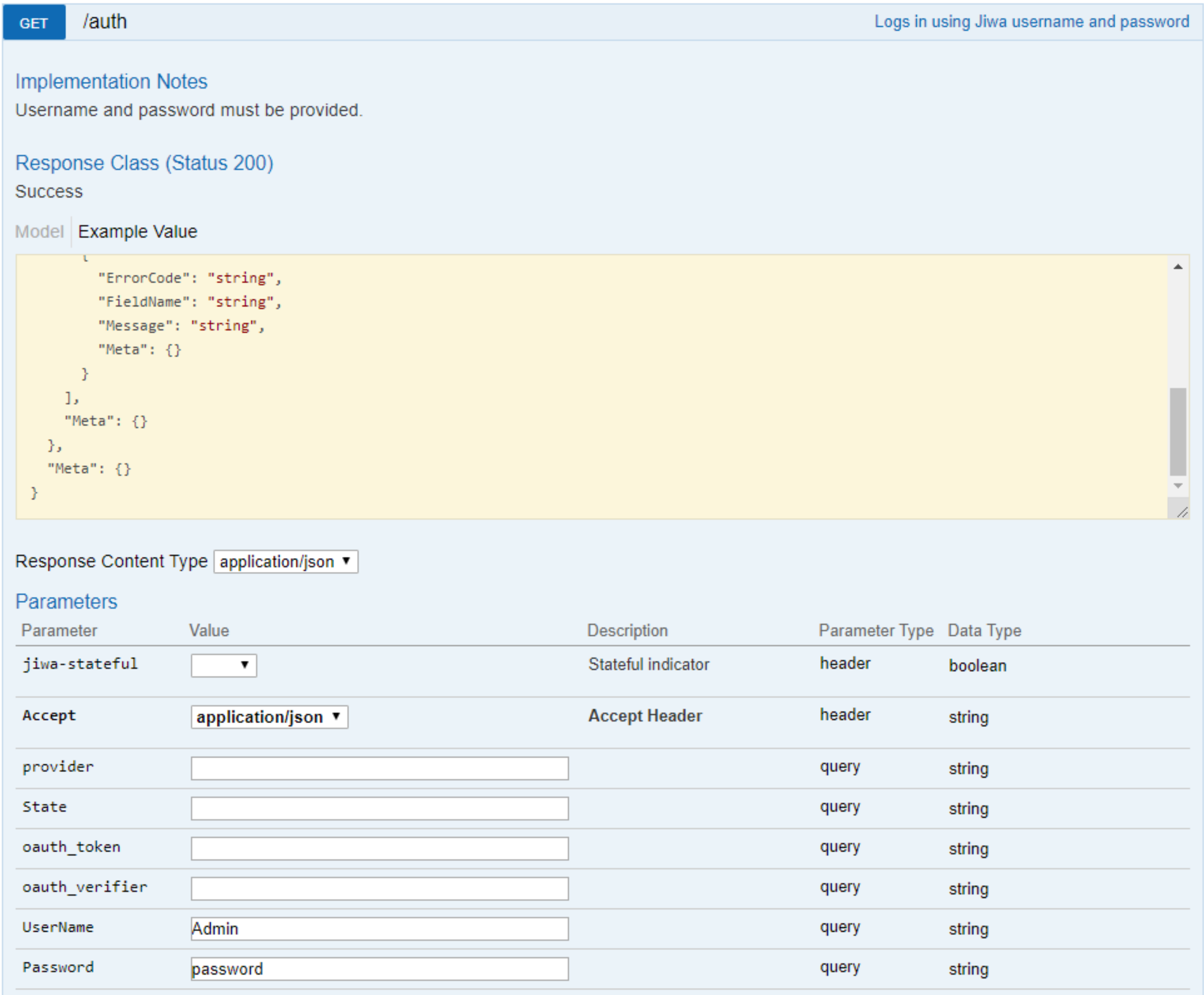
Visit the /swagger-ui route of your api in a web browser. For example, for our demo Jiwa api it is https://api.jiwa.com.au/swagger-ui/



https://api.jiwa.com.au/openapi

Step 2 - Authenticate

Locate and expand the **auth** section and then expand the section for **GET /auth**. Enter the **UserName** and **password** fields.



/auth

Logs in using Jiwa username and password

Implementation Notes

Username and password must be provided.

Response Class (Status 200)

Success

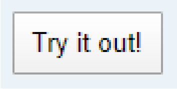
```
          {
            "ErrorCode": "string",
            "FieldName": "string",
            "Message": "string",
            "Meta": {}
          }
        ],
        "Meta": {}
      },
      "Meta": {}
    }
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
jiwa-stateful		Stateful indicator	header	boolean
Accept	**application/json**	**Accept Header**	header	string
provider			query	string
State			query	string
oauth_token			query	string
oauth_verifier			query	string
UserName	Admin		query	string
Password	password		query	string

Press the Try it out! button



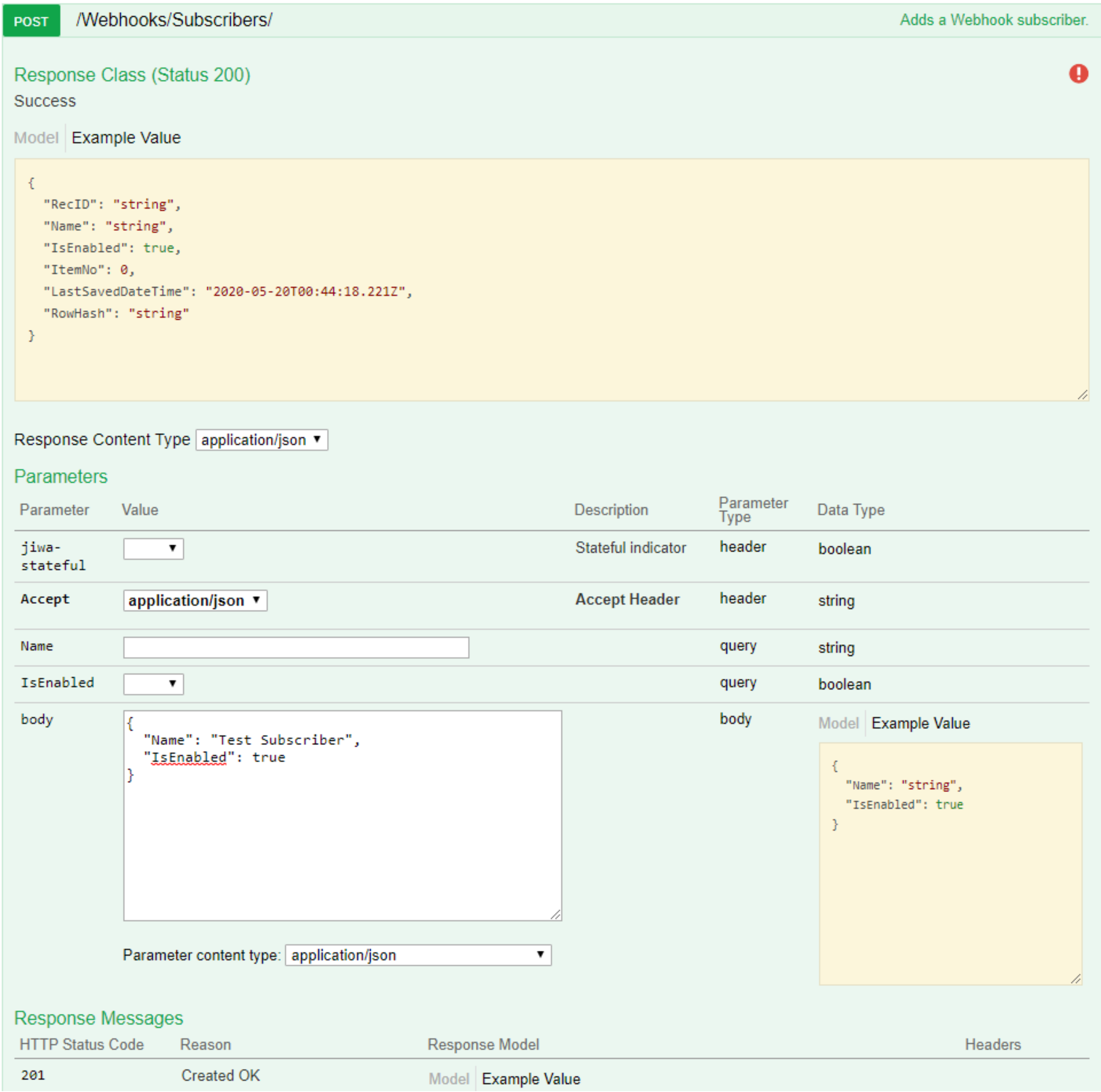
Step 3 - Create a Subscriber

Locate and expand the **Webhooks** section and then expand the section for **POST /Webhooks/Subscribers**.

Change the **Parameter content type:** to be **application/json**.

Click the json fragment in the **Example Value** area to pre-populate the **body** with the example json.

Edit the **body** to set your desired **Name** for the subscriber - "Test Subscriber" is shown below.



POST /Webhooks/Subscribers/ Adds a Webhook subscriber.

Response Class (Status 200)

Success

```
{
  "RecID": "string",
  "Name": "string",
  "IsEnabled": true,
  "ItemNo": 0,
  "LastSavedDateTime": "2020-05-20T00:44:18.221Z",
  "RowHash": "string"
}
```

Response Content Type: application/json

Parameters

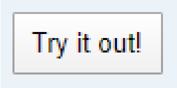
Parameter	Value	Description	Parameter Type	Data Type
jiwa-stateful		Stateful indicator	header	boolean
Accept	application/json	Accept Header	header	string
Name			query	string
IsEnabled			query	boolean
body	`{ "Name": "Test Subscriber", "IsEnabled": true }` Parameter content type: application/json		body	Example Value: `{ "Name": "string", "IsEnabled": true }`

```
{
  "Name": "Test Subscriber",
  "IsEnabled": true
}
```

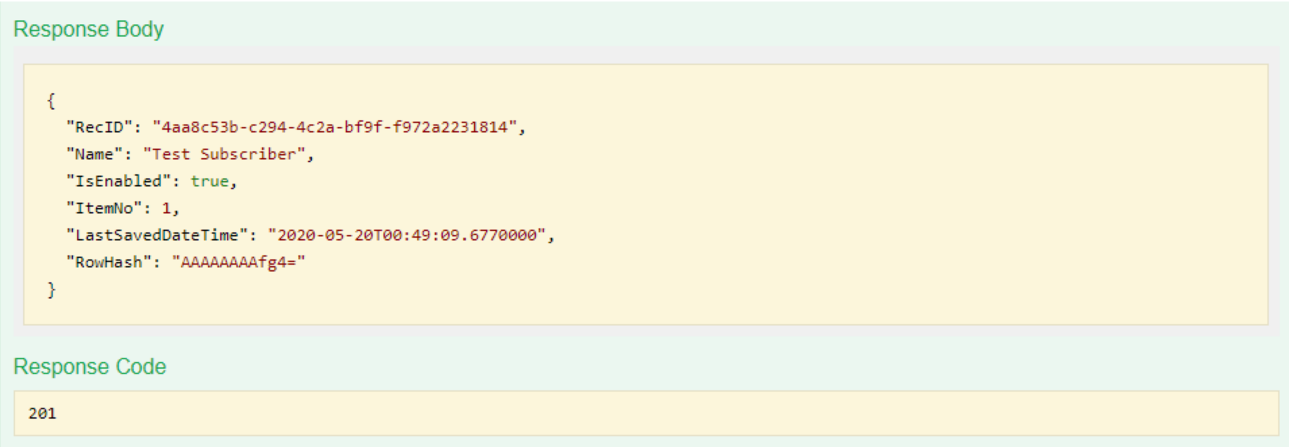
Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created OK	Model Example Value	

Press the Try it out! button



The response will be shown. The **RecID** in the response is the unique identifier for the subscriber - shown as **4aa8c53b-c294-4c2a-bf9f-f972a2231814** below. This will be needed for the next step, so select and copy the **RecID** value.



Response Body

```
{
  "RecID": "4aa8c53b-c294-4c2a-bf9f-f972a2231814",
  "Name": "Test Subscriber",
  "IsEnabled": true,
  "ItemNo": 1,
  "LastSavedDateTime": "2020-05-20T00:49:09.6770000",
  "RowHash": "AAAAAAAAfg4="
}
```

Response Code

```
201
```

Step 4 - Create a Subscription

Locate and expand the **Webhooks** section and then expand the section for **POST /Webhooks/Subscribers/{*SubscriberID*}/Subscriptions/**.

Change the **Parameter content type:** to be **application/json**.

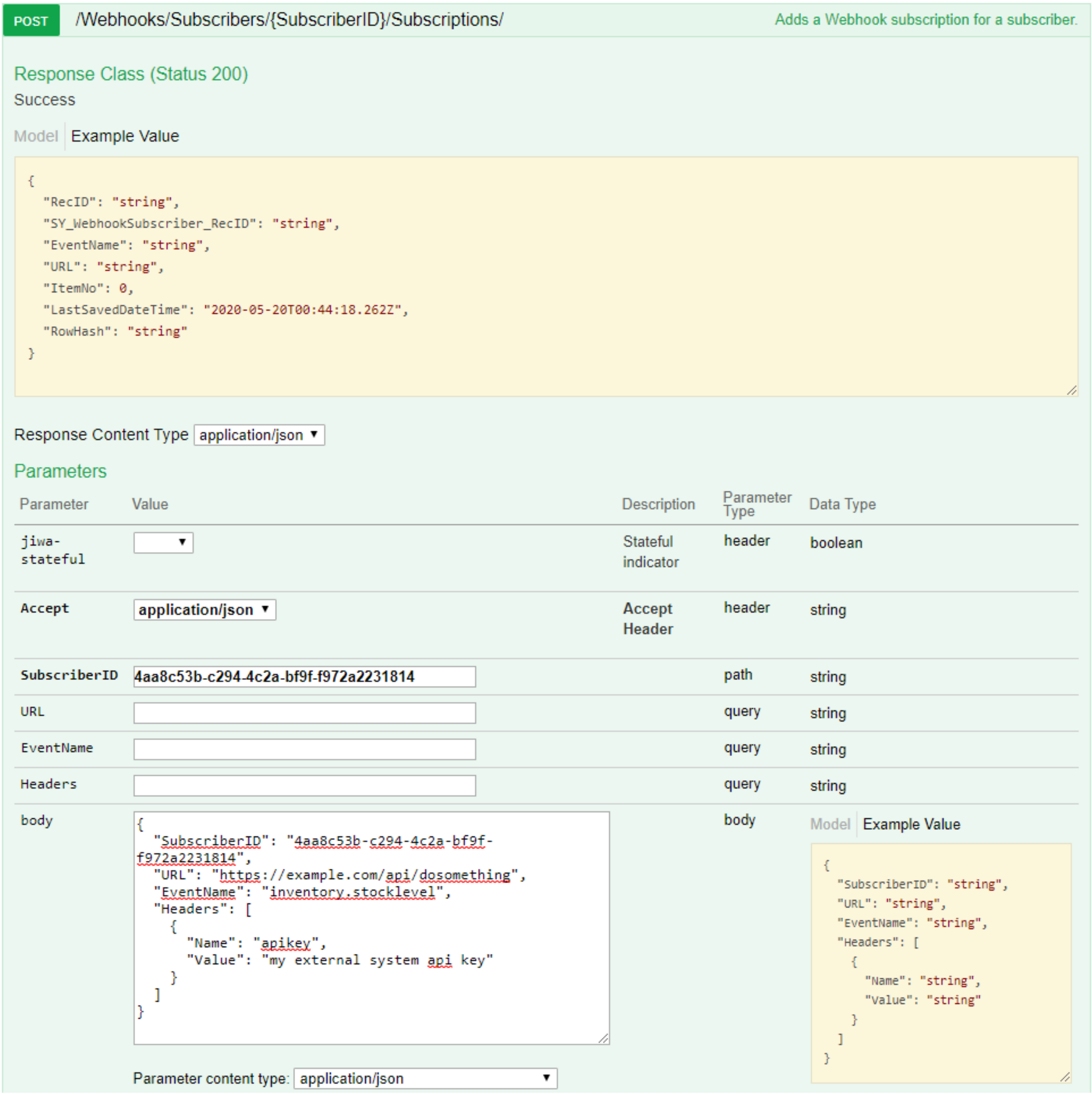
Click the json fragment in the **Example Value** area to pre-populate the **body** with the example json.

Edit the **body** to set the **SubscriberID** this is the RecID returned in the response of the previous step creating a subscription - "**4aa8c53b-c294-4c2a-bf9f-f972a2231814**" is shown below.

Edit the **body** to set your desired **URL** for the subscription - "https://example.com/api/dosomething" is shown below.

Edit the **body** to set your desired **EventName** for the subscription - "**inventory.stocklevel**" is shown below.

If required, set any headers the external system requires - the example below adds a header for setting an api key - the request POST sent to https://example.com/api/dosomething will contain these headers.



POST /Webhooks/Subscribers/{SubscriberID}/Subscriptions/ Adds a Webhook subscription for a subscriber.

Response Class (Status 200)

Success

Model | Example Value

```
{
  "RecID": "string",
  "SY_WebhookSubscriber_RecID": "string",
  "EventName": "string",
  "URL": "string",
  "ItemNo": 0,
  "LastSavedDateTime": "2020-05-20T00:44:18.262Z",
  "RowHash": "string"
}
```

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
jiwa-stateful	▾	Stateful indicator	header	boolean
Accept	application/json ▾	Accept Header	header	string
SubscriberID	4aa8c53b-c294-4c2a-bf9f-f972a2231814		path	string
URL			query	string
EventName			query	string
Headers			query	string
body	(see below)		body	Model | Example Value

body value:

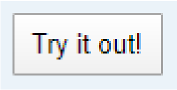
```
{
  "SubscriberID": "4aa8c53b-c294-4c2a-bf9f-
f972a2231814",
  "URL": "https://example.com/api/dosomething",
  "EventName": "inventory.stocklevel",
  "Headers": [
    {
      "Name": "apikey",
      "Value": "my external system api key"
    }
  ]
}
```

body Example Value:

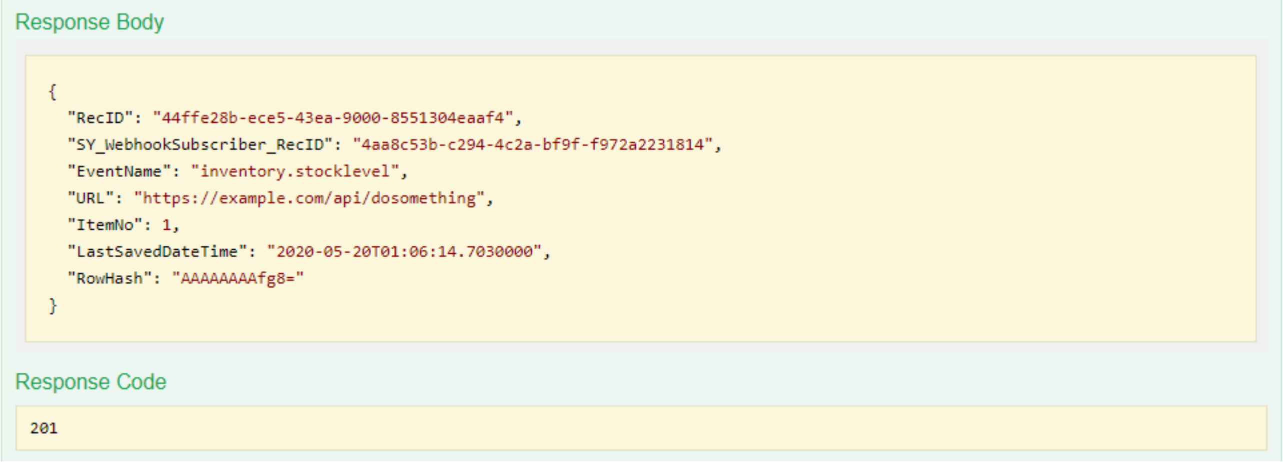
```
{
  "SubscriberID": "string",
  "URL": "string",
  "EventName": "string",
  "Headers": [
    {
      "Name": "string",
      "Value": "string"
    }
  ]
}
```

Parameter content type: application/json ▾

Press the Try it out! button



The response will be returned.



Response Body

```
{
  "RecID": "44ffe28b-ece5-43ea-9000-8551304eaaf4",
  "SY_WebhookSubscriber_RecID": "4aa8c53b-c294-4c2a-bf9f-f972a2231814",
  "EventName": "inventory.stocklevel",
  "URL": "https://example.com/api/dosomething",
  "ItemNo": 1,
  "LastSavedDateTime": "2020-05-20T01:06:14.7030000",
  "RowHash": "AAAAAAAAfg8="
}
```

Response Code

```
201
```

Once the above steps are completed, whenever a product stock level changes in Jiwa, a POST on the URL https://example.com/api/dosomething with a DTO containing the stock level information will be performed.