



SoapUI Pro: API 機能テスト ツール LoadUI Pro: API 負荷テスト ツール ServiceV Pro: サービス仮想化ツール VirtServer: 仮想サービスサーバー

ReadyAPI スタートガイド

For ReadyAPI 3.0



エクセルソフト株式会社 作成: 2019.12.13

はじめに	4 -
第 1 章: SoapUI スタートガイド	5 -
基本概念	5 -
用語	5 -
Web サービスをテストする方法	
1. 機能テストの作成	11 -
必要要件	11 -
使用する Web サービス定義	11 -
テストを作成	- 12 -
2. テストプロジェクトの探索	
3. SoapUI テストの変更	19 -
リクエストテストステップの追加	19 -
リクエスト パラメーターの変更	
4. SoapUI テストの実行	
個別のリクエストを実行	
テストケースの実行	
テストスイートとプロジェクトを実行する	26 -
5. SoapUI テストにアサーションを追加	28 -
例 1 – SLA アサーション	28 -
例 2 – レスポンス内容を確認	31 -
次のステップ	36 -
第2章: セキュリティ テスト	38 -
1. セキュリティ テストの作成および実行	39 -
2. テスト結果の表示	43 -
3. セキュリティ スキャンへのアサーションの追加	45 -
次のステップ	47 -
第 3 章: LoadUI スタートガイド	48 -
負荷テスト エディター インターフェイス	49 -
1. 新規の負荷テストの作成	53 -
2. 作成した負荷テストの実行	58 -
3. テスト結果の表示	61 -
4. 負荷テストの変更	65 -
5. 複数のシナリオの実行	69 -

 負荷テストへのアサーションの追加
次のステップ
第 4 章: ServiceV スタートガイド 76 -
基本概念の理解 76‐
仮想サービス76‐
オペレーション (アクション)
仕組み 77 -
その他78‐
1. 仮想サービスの作成 79 -
2. サービス レスポンスの設定 ⁻ 84 -
 レスポンス ディスパッチの設定 88 -
4. 仮想サービスの実行およびリクエストの送信
次のステップ 95 -
第 5 章: VirtServer のチュートリアル 96 -
1. VirtServer のインストールと実行 97 -
2. 仮想サービスの展開 99 -
3. 仮想サービスの実行とテスト 104 -
次のステップ106-
サポート 107 -
お問合せ先 107 -

はじめに

このガイドブックは、ReadyAPI (v3.0.0) ユーザー ガイドの中で、スタート ガイドに関し て説明した部分を抜粋して翻訳したものです。 ReadyAPI はいくつかの手段を組み合わ せて、Web サービスの包括的なテストの作成を支援します。

- 機能テストでは、サービスが期待どおりに機能することを確認します。
- セキュリティテストは、サービスがハッキング攻撃に対してどのように抵抗するか を検証します。
- 負荷テストは、サービスが大量のユーザーを処理できるかどうかを確認します。
- 仮想サービスは、作成される前にサービスの実際の動作をシミュレートします。

SoapUI: SoapUI スタート ガイド

- Secure: Secure スタート ガイド
- LoadUI: LoadUI スタート ガイド
- ServiceV: ServiceV スタート ガイド
- VirtServer: VirtServer チュートリアル

インストール ファイルの入手、各機能の詳細を含んだ最新バージョンの ReadyAPI ユーザー ガイドは、SmartBear 社の下記のサイトで参照できます。

https://support.smartbear.com/readyapi/docs/get-started.html

© 2019 SmartBear Software. All rights reserved.

Translated by XLsoft Corporation

第1章: SoapUI スタートガイド

ReadyAPI を使用すると、Web サービスの機能テスト、負荷テスト、およびセキュリティテ ストを簡単に実行できます。サービスのローカル仮想コピーを作成して、実際のテストが オンラインになる前にテストを実行することもできます。

このチュートリアルでは、SoapUI で基本的な機能テストを作成する方法について説明し ます。 ファイルから Web サービス定義をロードし、1 つの操作のテストを作成し、このテ ストを実行し、アサーションを使用してテスト結果を検証します。

基本概念

ReadyAPIを使用してテストを作成および実行するには、Webサービステクノロジー とテストの原則を一般的に理解する必要があります。 これらのトピックは膨大で、 ReadyAPIドキュメントの範囲外ですが、ReadyAPIをより早く開始できるように、 概要をまとめました。

用語

- Web サービスは、クライアントとサーバーが HTTP プロトコルまたは HTTP に基づいた他のプロトコルを介して Web 上でデータを交換するクライアント/サーバーアプリケーションです。このようなアプリケーションの例には、ナビゲーションソフトウェア、オンラインバンククライアント、気象監視システムなどが含まれます。
- クライアントがリクエストを送信する URL には、テストされるサーバー(host)、通信
 に使用されるポート番号、およびリクエストされたサーバーリソース (ページやファ イルパスなど) に関する情報が含まれます。

http://local-server:8080/api/v2/test-sets/id35/results
Protocol Host Port Resource
(http or https)

• クライアントがサーバーに送信するリクエストの構造は次のとおりです。

- HTTP メソッド (GET、POST、DELETE など)、ターゲット URL、プロトコル バージョンを指定する開始行。
- レスポンス データの予期される形式、またはリクエスト データのサイズと 形式などの追加情報を渡すヘッダー。
- (オプション) リクエスト ボディ。一部のリクエストタイプはそれを使用しません。



レスポンスには同様の構造があります。

- レスポンスコードとメッセージのある開始行。頻繁に使用されるコードには、200 OK (成功) と 404 Not Found (失敗、リクエストされたリソースが見つかりません) があります。
- レスポンスデータ形式を説明し、Cookie、サーバー情報などの追加の値 を含むヘッダー。
- レスポンスボディ。たとえば、リクエストされたデータ、画像、ファイルなどの配列。

Start line	HTTP/1.1 200 OK
	Cache-Control: private, max-age=0
	Content-Type: text/xml; charset=utf-8
Headers	Vary: Accept-Encoding
	Server: Microsoft-IIS/7.0
	X-AspNet-Version: 4.0.30319
	X-Powered-By: ASP.NET
	Date: Thu, 15 Feb 2018 08:44:00 GMT
	xml version="1.0" encoding="utf-8"? <soap:Envelope xmlns:soap="http://schemas.xmlsoap.</td>
Response	<table></table>
body	<country>Sweden</country>
body	<city>Linkoping / Malmen</city>

| | |

- リクエストおよびレスポンスのボディでよく使用される形式は、JSON および XML です。
- クライアントが実行のためにサーバーに送信するコマンドは、サービスのアーキテ クチャスタイル (SOAP または REST、以下を参照) に応じて、アクション、メソッド、 または操作と呼ばれます。
- Web サービスの 2 つの一般的なアーキテクチャスタイルは、SOAP と REST です。
 - SOAP サービスは、HTTP 上に構築された SOAP プロトコルを使用します。これらのサービスは、POST タイプの HTTP リクエストを使用し、リクエストおよびレスポンスのボディで XML 形式のデータを渡します。すべてのリクエストは同じ URL に送られ、実行される操作は、リクエストボディの特別なリクエストヘッダーまたは XML 要素によって指定されます。

SOAP サービスは、サービスでサポートされる操作、そのパラメーターの 種類、およびデータ形式を厳密に記述する WSDL 定義を使用します。

 REST サービスは HTTP 経由で機能します。実行される操作は、HTTPメ ソッドとリクエストされたリソース名の組み合わせによって設定されます。 たとえば、オンライン ペットストアの RESTful サービスには、/ pets リソー スを含めることができます。

POST http://petstore.io/pets リクエストはペットに関する情報をデー タベースに追加でき、GET http://petstore.io/pets リクエストは利用 可能なペットに関する情報を取得できます。 REST サービス定義には、OpenAPI (Swagger)、WADL、およびその他の いくつかの形式があります。ただし、一部の開発者は、RESTful サービス の定義をまったく提供していません。

Web サービスをテストする方法

Web サービスが正常に機能することを確認するには、機能テストを作成して実行します。

これらのテストは、サーバーにリクエストを送信し、そのレスポンスを検証します。

ReadyAPI では、**SoapUI** で機能テストを作成します。 SoapUI という名前にもか かわらず、SOAP および REST サービスの両方のテストをそこで作成できます。 特別なエディターでリクエストを簡単にシミュレートし、パラメーターをカスタマイズ できます。

Projects	Dashboard	Integrations	Endpoint Explorer @	þ
Navigator		×		3
Search Projects		۹	Test Step Request 2 (Add pet))
Project APIs	1		Nettonoa Endpoint Nesource Parameters POST http://petstore.swagger.io /v2/pet ?replace=8sz= ▶ Send Add Assertion]
⊖-SoapUl ⊖-	Project 1 TestSuit	e 1	Request Generate Values Response Smart Assertion Proved Rev Outline Server VM ISON VITH Rev Outline Server	, I
		e (Log in)	NML JOUN HIML Kew Outline Overview Q ▲ Image: Comparison of the second sec	1
Request 2 (Add pet) Request 3 (Find by stat Request 4 (Update pet) Request 5 (Log out)			Name Value type replace S[DataSourceFreplace] QUERY sz S[DataSourceFreplace] QUERY	
test step	Security Tests Pro	te Loop	Media Type assolication/ison Post QueryStric Request editor	
ServiceV Pro			"category": ["inter: 134, "name": "sting"), "name": "Lucky", "photoEvick:: ["meting"],	
			"togs": [] "name": "sting"]), "status": "avslable")	
			Auth Attachments (0) Headers (0)	
			♠ ●Assertions (2)	
C REST Reque	est Properties		Request Log (0)]
Custom Pr	operties		Learn about REST Request test steps	

レスポンスデータとレスポンスコードを検証するには、アサーションをテストリクエ ストに追加します。

Test Step Request 2 (Add pet)			🕜 API Request 🖉 API
Method Endpoint F	Resource	Parameters	
POST http://petstore.swagger.io	/v2/pet	?replace=&sz=	Send Add Assertion
Request Request Raw Outline Form	Generate Values Resp	ponse /IL JSON HTML Raw	Market Constraint Street Street Constraint Assertion
Q • • •	0 1	T T T T T T	🔄 Transfer to 🗙 Assert 👻 🌔
Name Value replace \${DataSource#replace} sz \${DataSource#sz} Media Type application/ison Implication #id#: 0, Implication/ison Implication "category": { "id#: 1234, "name": "string" Auth Attachments (0)	Type	JSON Node id category id name name photoUrls id o tags id o id id name name photoUrls id o tags id o o o id o o o o o o o o o o o o o	Value -9223372036854775808 1234 string Lucky string 0 ▼
			•

サーバーが正常に動作しているかどうかを判断する最も簡単な方法は、レスポン スコードを確認することです。200 OK は通常、サーバーがリクエストを正常に処 理したことを意味します。

実際には、クライアントは通常、一連のリクエストをサーバーに送信します。たとえば、オンラインショップの場合、最初のリクエストはサインインに使用され、後続のリクエストは一部の製品の購入に使用されます。

SoapUI では、リクエストやその他のテストステップをテストケースに整理すること により、この実際の動作をシミュレートします。



連携して動作する複数のテストケースは**テストスイート**にグループ化され、テスト スイートはテストプロジェクトに属します (上の画像を参照)。

テストプロジェクトを作成し、自動化された機能テストを追加しましょう。

1. 機能テストの作成

必要要件

ReadyAPI で **SOAP** サービスをテストするには、このサービスの WSDL 定義が必要で す。この定義では、サービスの操作、およびリクエストとレスポンスの形式について説明 します。 ReadyAPI はこの情報を使用してリクエストをシミュレートします。

REST サービスにも定義を含めることができます。 最も頻繁に使用される定義形式は、 OpenAPI (以前の Swagger)、WADL、およびその他のいくつかです。 これらの定義を ReadyAPI にロードし、これらの定義の情報に基づいてテストケースを作成できます。

一般的な場合、REST サービスには定義がまったくない場合があります。サービス URL へのリクエストを記録することにより、このようなサービスのテストを ReadyAPI で作成で きます (これは <u>API Discovery</u> と呼ばれます)。 ReadyAPI は、追跡されたトラフィックに 基づいてリクエストおよびレスポンス パラメーターに関する情報を取得します。 ただし、 この「観測」データは定義からの情報ほど正確ではないため、可能な場合は定義を使用 することをお勧めします。

使用する Web サービス定義

Petstore サンプル Web サービスのテストを作成します。 これは REST サービスです。 ここでその定義を見つけることができます-

http://petstore.swagger.io/v2/swagger.json

この定義には OpenAPI 2.0 (Swagger) 形式があります。 今すぐ定義をダウンロードす る必要はありません。 ReadyAPI は、後で機能テストを作成するときにこれを行います。 下記参照。

テストを作成

 スタートページを開き、[Create Functional Test] をクリックし、後続のダイアログで で [API Definition (API 定義)] を選択します。

File Edit View Project Suite Cas	e Step Request TestRuns Help
Projects Dashboard Integration	ons Endpoint Explorer 🛞
Navigator X Search Q Projects	SMARTBEAR ReadyAPI.
	Create Create Create Virtual Service Send Functional Test Image: Service Virtual Service Send Recently Opened Create a functional test from: X
	Start Start License Applied License of REST API.
Workspace Properties	Learn about creating a test
Property Value	Authentication Sample Project
Name Projects	
Description	4
Project Root	
	18 Always show this page on launch

2. 後続のウィザードで、Web サービスの定義の URL を指定します。このチュートリアルでは、次の URL を使用します。

http://petstore.swagger.io/v2/swagger.json



Next をクリックして、続行します。

3. 追加した定義の新しいプロジェクトを作成するか、既存のプロジェクトに追加する かを選択します。

2 New Functional Test	×
API Definition Setup	
Specify Elia Specify Endpoint Specify project Add Assertings G Create Test	
Add to a new or existing project:	
<new project=""></new>	
Select an existing project if you want to create a test suite for a new definition within it.	
Click to proceed	
Learn about creating a test	
Back Next Cancel	

注意: ワークスペースでプロジェクトが開いていない場合、ウィザードはこの手順 をスキップします。 この場合、ReadyAPI は新しいプロジェクトを作成し、 新しいテストを追加します。

Next をクリックして、続行します。

4. ウィザードのこのページで、テストに追加するアサーションを選択できます。 ReadyAPIは、選択したアサーションを新しいテスト リクエストに追加します。

<u>アサーション</u>は、API が期待どおりに機能することを確認します。 このチュートリアルの後半で、これらの詳細について説明しますが、ここで選択をクリアし、**Next**をクリックします。

	2 New Functional Test	×
	API Definition Setup	
	Specify File Specify Endpoint Specify project Add Assertions Greate Test	
	Choose assertions	
	Valid HTTP Status Codes	
	200	
1. Clear selection	Swagger Compliance Assertion	
	Swagger / OAS URL: http://petstore.swagger.io/v2/swagger.json Browse	
	Image: Response SLA Assertion Max response time: 200 milliseconds	
	2. Click to proceed.	
	Learn about creating a test Back Next Cannel	

5. Web サービスに定義されているすべての操作に対して1つのテストケースを使 用するか、複数のテストケース (各操作に1つ)を使用するかを選択します。 後 者のオプションを使用しましょう。

🚾 New Functional Test				×
	API Defi	nition Setup		
1 Specify File	Specify Endpoint 3 Specify	ecify project 4 Add Assertions	5 Create Test	
Select how to group your tests	52			
	Single TestCase	Multiple TestCases		
	All of your API requests will be contained in one TestCase.	Each of your API requests will have its own TestCase.		
	Select	Selected		
Learn about creating a test				
		Back	Finish Canc	el

Finish をクリックして、テストを作成します。

6. ReadyAPI は、テストプロジェクトを作成し、テストケースを追加します。

その後、もう1つのダイアログボックスが表示され、作成したテストを実行したり、 データソースを追加したりできます。このチュートリアルでは、これらのオプション は使用しません。このダイアログを閉じます。



左側のナビゲーターパネルでプロジェクトを確認できます。このチュートリアルの次のス テップでは、作成されたプロジェクト、サービス、およびその操作を調べます。

2. テストプロジェクトの探索

作成されたテストプロジェクトは、左側の[Navigator]パネルで確認できます。

File Edit View Project Suite Case Step Reque	st Test Runs Help	
Projects Dashboard Integrations		Endpoint Explorer (2)
Navigator ×	🛞 🍥 🕨 🕇 Test Case 🖹 Report 😥 🛣 🗘	Default environment • 🔂 Proxy • 🔅
Search Q	Project 1 TestSuite 1	
My Projects	▶ ★ ■ 🖽 ⊐ ⊗ 🖻 🖯	TestCases Transaction Log Compare History Coverage
or APIs ⊡∵SoapUI Pro	+ × ♥▼ Test cases: 22	Show: 🔻 🗐
Project 1 TestSuite 1 O	/pet/(petId)-Request 1 TestCase	`
·································	/pet/(petid)-Request 1 TestCase	
D…✔ /pet-Request 1 TestCase		
Image: A constraint of the second	/pet/[petId]-Request 1 TestCase	
Image: A state of the state	(and ((and d)) (and an all more Pressent 1 Test) are	
	//pet/(petra)/uproadimage-request + restLase	

プロジェクトには複数のテストケースがあります (操作ごとに1つ)。 これらはすべてテス トスイートにグループ化され、テストスイートはプロジェクトに属します。

この場合、各テストケースには1つのリクエストテストステップのみがあります。実際には、テストケースには通常複数のステップがあります。



このチュートリアルの次のステップで、テストケースにリクエストを追加する方法を説明します。

サービスを調べるには、[Navigator]パネルで API ノードを展開します。 サービスのリ ソースとリクエストのツリーのような構造が表示されます。



最上位のノードは Web サービスに対応しています。その子ノードはリソースに対応して います。 リソースノードには、順番に、Web サービス仕様のリソースに対して定義された リクエストに一致する子ノードがあります。

右側のエディターを使用して、選択したサービス、リソース、またはリクエストのパラメー ターを表示します。

一部のリソースには、複数のリクエストが定義されています。これらのリクエストには通常、異なる HTTP メソッドがあります。他のリソースには、リクエストが1つしかありません。プロジェクトのツリーに表示されるリクエストは、テンプレートリクエストとして機能します。たとえば、ここで各リクエストに異なるパラメーターを設定し、これらのリクエストをSoapUIのリクエストテストステップのベースとして使用できます。

Projects Dashbo	ard Integrations														Endpo	int Explore		8
Navigator		×	©)		6	+ Create Test	A C			Δ	8	R	Default env	vironment	•	Prox	y •	ŝ
Search		۹	API Re	ques	t Re	quest 1										Add to Te	estCas	
Project 1			Method	E	ndpoin	t .		Resource				Para	meters					
⊖ APIs			POST	_	http://	petstore.swagger.io		 /v2/pet 									Sen	
😑 苯 Swagger P	letstore																12 +	6
⊕□] /pet [/\	2/pet]		Request							esponse	•							
e- 🚹 add	iPet		Request	Raw	Ou	tline Form				XML	JSON	HTM	IL Raw	Outline	Overvi	ew		
ST DF	Request 1		$+\times$	-12 Ç) v	^		0		11	= =	<u> </u>	<u> </u>	_ <u>+</u> XP				Ð
ST	Request 2																	
	udD. Centus 11:0/acc/findD. Ce																	
0 - I /pet/fir	dByTags [/v2/pet/findByTags]										_							
0-0 /pet/{p	etid} [/v2/pet/{petid}]				V	iew request pa	rameters in a t	able Request		litor								
0-🛄 /pet/{p	etid}/uploadimage [/v2/pet/{petic					+ Ade	Parameter	Request	. et									
🛛 🖓 /store/	inventory [/v2/store/inventory]					1 100					_							
0 - 🔲 /store/	order [/v2/store/order]																	
0 - 🖵 /store/	order/{orderld} [/v2/store/order/-																	
🗰 🖵 /user [/v2/user]																	
🗴 🛄 /user/o	reateWithArray [/v2/user/create.	•	Advar	iced opt	ions													
0 U /user/o	reateWithList [/v2/user/createWi		Media Ty	pe appl	ication	/ison 💌 🖸	Post QueryStri	ng	View an outline of the response's JSON/XML of			I/XML co	ntent					
0 U /user/i	ogin [/v2/user/login]		((Send a request to get a response)									
	ogout (/v2/user/logout)	,	"id" : 0, "categor	v:e														
a grosery	usernamer (vz/usernusername,	· 🖃	TIOT : 0,															
Request Properties			name"	: "string														
Property	Value		"name" :	"doggie"	ring 1													
Name	Request 1		"tags" :	() 	and P													
Description			"id" : 0, "name"	: "string"														
Encoding		17)],															
Endpoint	http://petstore.swagger.io	11	status"	: availat	ne.													
Timeout		111							41									
Bind Address		•	🔒 Auth	(Inherit	From I	Parent) Attachmen	ts (0) Headers (0)		I L									
		_																

プロジェクトでは、リクエストに対して必要な数のテンプレートを作成できます。

リクエストエディターからリクエストを実行して、このリクエストが正しく機能するかどうかを 確認することもできます。ただし、これは実行される個々のリクエストであることに注意し てください。実際のシナリオをシミュレートするには、複数のリクエストでテストケースを実 行する必要があります。

チュートリアルの次のステップでは、リクエストをテストケースに追加し、リクエストパラメー タを変更します。

3. SoapUI テストの変更

テストステップをテストケースに追加し、リクエストパラメータを変更する方法を見てみま しょう。 "/pet/findByStatus-Request 1 Test Case" を変更します。 REST リクエストのテス トステップを追加します。

リクエストテストステップの追加

- 1. 左側の Navigator ツリーで、/pet/findByStatus-Request 1 Test Case ノード にカーソルを合わせ、右のボタン(■) をクリックします。
- 2. フライアウトメニューで、**REST リクエスト**テストステップを選択します。すばやく見 つけるには、[検索]フィールドに名前を入力して始めることができます。



 後続のダイアログで、テストステップ名を入力し、作成するテストステップのテンプ レートリクエストを選択し、アサーション チェックボックスをオフにして、Create Test をクリックします。



リクエストがテストケースに追加されます。

Projects
🗆 💼 Project 1
oAPIs
🗄 SoapUI Pro
Project 1 TestSuite 1
/pet/{petld}-Request 1 TestCase
/pet/{petld}-Request 1 TestCase
/pet/{petld}-Request 1 TestCase
/pet/{petld}/uploadImage-Request 1 TestCase
/pet-Request 1 TestCase
/pet-Request 2 TestCase
/pet-Request 1 TestCase
/pet-Request 2 TestCase
→ /pet/findByStatus-Request 1 TestCase
Request 2
Image: pet/findByTags-Request 1 TestCase
Interview of the second sec

リクエスト パラメーターの変更

現在、同じ操作をシミュレートする2つのリクエストがあります。 リクエストが異なるデー タのセットを返すようにパラメーターを変更しましょう。

- 1. Navigator ツリーで、GET Request テストステップを選択します。
- リクエストエディターで、status 行の[Value]セル内のどこかをクリックし、
 available を入力します。 ENTER を押して変更を確認します。



3. 同様に、テストケースの2番目のリクエストを選択し、*status*パラメーターの [**Value**]セルに sold と入力します。Enter キーを押して変更を確認します。



これで、テストケースを実行する準備が整いました。これを次のステップで行います。

4. SoapUI テストの実行

ReadyAPI では、リクエスト、テストケース、テストスイート、またはテストプロジェクト全体を 実行できます。

個々のリクエストを実行することは、これらのリクエストがどのように機能するかを確認す る必要がある場合に意味があります。 ユーザーシナリオをシミュレートするには、複数の リクエストをシミュレートするテストケースを実行する必要があります。 テストスイートは、 複数のテストケースの実行に役立ちます。テストプロジェクトの実行は、このプロジェクト に属するすべてのテストスイートの実行を意味します。

個別のリクエストを実行

通常、テストを作成するときに個々のリクエストを実行します。これにより、レスポンス データをすばやく表示したり、必要に応じてリクエスト パラメーターを変更したり、アサー ションを追加したりできます。複雑なテストでは、リクエストは多くの場合、前のテストス テップのデータに依存します。これらのリクエストは、個別に実行すると失敗します。

個別のリクエストを実行するには、左側の Navigator パネルで選択し、メインツールバー

の ▶ をクリックするか、リクエストエディター ツールバーの ▶ Send をクリックします。



リクエストエディターの右側にレスポンスの内容が表示されます。

Test Step GE	T Reques	st									API Request 🖉 API
Method	Endpoint		Resource		Parameters						
GET	http://petstor	re.swagger.io	•	/v2/pet/findByStatus		?stat	tus=availal	ble		► Send	Add Assertion
GET Request Request Name status	http://petstor Outline For	re.swagger.io	QUER	/v2/pet/findByStatus Generate Values Type Y	Response XML 1E 2E 3 4 5 5 6 7 7 8 9 6 6 7 7 8 9 6 9 9 10 11 12 13 14 15 16 16 17 18 20 21 22 23 24 25 26 27 7 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	3 (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	<pre>us=availal HTML "id" : "catege "d" "st ", "st ", "st ", "st ", "id" : "stage" { "id" : ", ", "stage" { "id" : "stage" { "id" : "stage" { "stage"</pre>	Raw 65449 orsy": : : "c0,me": : "d0 :: : "d0 :: : "d0 :: : "id":: : "id":: : "id":: : ": : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :	Outline 25, { 3*string" ggie", : { 0, : "string" 20, { 3*string" *string" 0, : "stri	► Send	Add Assertion
Auth Attachments	(0) Headers (0	n			Header	(8)					ľ

テストケースで別のリクエストを実行し、そのレスポンスを確認します。.

テストケースの実行

テストケースを実行するには、Navigator パネルでテストケースを選択し、メインまたはエ ディターのツールバーの ▶ をクリックします。



SoapUIは、テストケースのテストステップを1つずつ実行します。 テストケースエディ ターに結果が表示されます。 [**Transaction Log**] ページには、テストの実行に関する時 間情報が表示されます。

Navigator ×	🕑 🅘 🕨 🖶 T	est Step 📄 Report 🕢		4	🖯 👷 Defau	It environment 🔹	Proxy • 🔞
Search Q Projects Project 1 APIs	/pet/findByStatus-Rec > * Compared Co	res two runs Compare History Coverage	Requirements Step-by-Step	✓ PA: Run Test On Demand	55		
⊖-SoapUI Pro ⊖	Results: 0 Steps PASS	0 Steps FAIL			Filter Bu		
 // pet/(petld)-Request 1 TestCase // pet/(petld)-Request 1 TestCase // pet/(petld)/uploadimage-Request 1 	Name	Test Case	Test Suite	Start Time 🔺	Time Taken	Status	
	✓ /pet/findByStatus-R		Project 1 TestSuite 1	2019-11-25 16:36:46.631	18885ms	PASS	Open
	🔛 GET Request	/pet/findByStatus-Req	Project 1 TestSuite 1	2019-11-25 16:36:46.667	17104ms	UNKNOWN	Open
Request 2 	🔛 Request 2	/pet/findByStatus-Req	Project 1 TestSuite 1	2019-11-25 16:37:03.980	1781ms	UNKNOWN	Open

ご覧のとおり、テストケースエディターには、テスト実行の傾向を表示できる History (履歴)や、2 つのテストログを比較できる Compare (比較)など、テストステップレベルでは使用できないページがあります。

リクエストが正常に実行されたことに気付くかもしれませんが、Transaction Log にはテ ストステップがゼロと報告されています。



これは、テストステップに結果を検証するチェックポイント(アサーション)がないために発生 します。このチュートリアルの次のステップでアサーションを作成します。

次のステップに進む前に、テストスイートとプロジェクトの実行方法について説明します。

テストスイートとプロジェクトを実行する

テストスイートまたはプロジェクトを実行するには、Navigator でこのスイートまたはプロ ジェクトを選択し、メインまたはエディターのツールバーの ▶ をクリックします。

	\odot	Θ	🗩 Run	🐻 Save
Q	Pro	piect	1 TestS	uite 1
		*	11 ₹ 0	
	+	×	•	Test cases: 2
		/pet/{pe	etid}-Reque	st 1 TestCase
	Q Test	Q Pro	Q Project + × /pet/{pet/{pet/{pet/{pet/{pet/{pet/{pet/	Q Project 1 TestS ▶★ ₽ + > /pet/{pet/d}-Request /pet/{pet/d}-Request

エディターにテスト結果が表示されます。

Navigator ×	€ ⊙ ▶ Run 🗟 Save 🕂	Test Case 📄 Report	💭 Monitor APIs 🛛 🎽	₩ JIRA	D Events	Databases	₩ тик	Default enviro	nment •	Proxy •	Preferences
Scotch Q. Projects ■ Project 1 ∳ APIs ∳ SapJU Pro	Project 1 TestSuite 1	X		✔ PASS				TestCases	Transaction Log Show: 💌 All	Compare H	istory Coverage
○ ■ Project TreatSure 1 6 → ✓ /prol/prefs/Request TreatCase 9 → Ø → Ø → Ø → Ø → Ø → Ø → Ø → Ø → Ø →	/pet/(pelid)-Request 1 TestCase /pet/(pelid)-Request 1 TestCase			✓ PASS ✓ PASS							
	/pet/(pet/d)-Request 1 TestCase /pet/(pet/d)/uploedimage-Request 1 1	estCase		✓ PASS ✓ PASS							

 デフォルトでは、テストスイートを実行すると、テストランナーはこのスイートのすべての テストケースを連続して実行します。それらを並行して実行するには、最初にツールバー の こをクリックしてから をクリックします。



プロジェクトは同様の機能を提供します。デフォルトでは、一連のテストスイートを実行し ます。 プロジェクトエディターのツールバーには、テストスイートを並行して実行す る コマンドがあります。

リクエスト、テストケース、またはテストスイートエディターでは、リクエストが正常に実行されたかどうかを簡単に確認できます。レスポンスデータを検証したり、実行時間をリクエストするには、アサーションを使用します。チュートリアルの次のステップでは、テストステップにアサーションを追加します。

5. SoapUI テストにアサーションを追加

アサーションは、ターゲット Web サービスのパフォーマンスを確認するテスト結果に適用 される検証ルールです。SoapUI テストでは、アサーションを使用して、レスポンスコー ド、レスポンスヘッダー、レスポンスボディの個々の値、リクエストの実行時間を確認し、 その他のチェックを行います。テストランナーは、テストステップ(リクエスト)が終了した後 にアサーションを実行します。

例 1 - SLA アサーション

テストされる Web サービスが事前定義された制限時間内に応答するかどうかを確認するアサーションを作成しましょう。

1. SoapUIの[Navigator]パネルで[**GET Request**]を選択し、[**Add Assertion**]をク リックします。



 ダイアログで、左側の SLA のカテゴリを選択し、次に右側の Response SLA を選 択し、[Add]をクリックします。

All assertions Search Recently used SLA Property Content SLA Compliance, Status and Standards Standards Script Validates that the last received response time was within the defined limit. Applicable to TestSteps that send requests and receive responses. JMS Response JMS Response JMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps with	Assertion		
Il assertions Il assertions Search Recently used SLA Property Content Compliance, Status and Standards Validates that the last received response time was within the defined limit. Applicable to TestSteps that send requests and receive responses. SLA JMS Response Security JMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit.		Hide descriptions	
Recently used Property Content Compliance, Status and Standards Script SLA Validates that the last received response time was within the defined limit. Applicable to TestSteps that send requests and receive responses. JMS Response JMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit	ertions	Search Q	
Property Content Compliance, Status and Standards Script SLA MS Response Security JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit.	ntly used	SLA	
Compliance, status and standards Standards Script SLA MS Response Security JMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit.	erty Content	Decesso SIA	1
Station Imit. Applicable to TestSteps that send requests and receive responses. MS Response JMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit.	dards	Validates that the last received response time was within the defined	
SLA MS Response Security JMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit	t	limit. Applicable to TestSteps that send requests and receive responses.	
MS Response jMS Response JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit			1
JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit	Response	IMS Response	
JMS Status Validates that the JMS request of the target TestStep executed successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit.		,	
successfully. Applicable to Request TestSteps with a JMS endpoint. JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit.		JMS Status Validates that the IMS request of the target TestStep executed	
JMS Timeout Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit		successfully. Applicable to Request TestSteps with a JMS endpoint.	
Validates that the JMS statement of the target TestStep did not take longer than the specified duration. Applicable to Request TestSteps wit		IMS Timeout	
longer than the specified duration. Applicable to Request TestSteps wit		Validates that the JMS statement of the target TestStep did not take	
		longer than the specified duration. Applicable to Request TestSteps wit	
Convity		Casulia:	
Security		security	

- **注意:** *SLA* は、service-level agreement (サービスレベル契約)の略です。 通常、この 用語は、操作の実行にかかる最大時間を意味します。 もちろん、この時間は ユーザーに受け入れられるはずです。
- [Add Assertion]ダイアログで[Add]をクリックすると、アサーションパラメーターを 設定できるダイアログボックスが ReadyAPI に表示されます。このダイアログボッ クスは、アサーションごとに外観が異なります。以下の画像は、Response SLA アサーションの検索方法を示しています。

リクエストの最大許容応答時間として 1000 ミリ秒を使用しましょう。1000 を入力 して[**OK**]をクリックします。

Configu	re Response SLA Assertion	<
7	Specify the maximum response time (ms)	
	OK Cancel	

現在、リクエストの完了に指定されたミリ秒数よりも長い時間がかかると、アサー ションがトリガーされ、テストは失敗します。実行時間が指定した値以下の場合、 チェックはパスします。

新しいアサーションは、Assertions パネルで利用できます。

Navigator $ imes$	©)	6 🔊 🍃	ć				4	3 8	Defaul	lt environm	ent 💌	Proxy 🕶	ŝ
Search Q	Test Step G	T Reque	et									Paguart 10	ADI
Projects	Test Step G	_1 iveque	51				_				C AFI	inequese 0	Art
🖃 🖿 Project 1	Method	Endpoint		Resource			Parame	ters			_		
O-APIs -	GET	http://petst	ore.swagger.io	/v2/pet/findByState	us		?statu	s=availabl	le		Send	Add Assert	ion
- SoapUI Pro												77	+ 6
Droject 1 TestSuite 1	Request			Generate	/alues	Respon	se					Smart Asse	ertion
ii 🗸 /pet/{petId}-Request 1 T	Damage Dama	Outline E				VA.0	ICON	1.075.41	Dama	0.45.4	Querrieu.		_
ID…✔ /pet/{petId}-Request 1 T	Request Raw	Outline Fo	orm			XML	JSON	HIML	Kaw	Outline	Overview		
🕼 🗸 /pet/{petid}-Request 1 T	Q • ^				0	1	= ()						-
🕸 ✔ /pet/{petld}/uploadimag	Name		Value	Туре		2	- (CE 4 4 9	25			882
Image: Image: Antipage Anti	status	a	vailable Q	JERY		4	=	"categ	ory" :	-{			
ID- ID- /pet-Request 2 TestCase						5		"id	":0,				
Image: A state of the state						7		"na	me":	"string"			
ID ··· ✓ /pet-Request 2 TestCase						8		"name"	: "dog	ggie",			
/pet/findByStatus-Reque						9	Ξ	"photo	Urls"	: I			
						11		1.	ring.				
Request 2						12	8	"tags"	: 1				-
Image: A state of the state							4						
Image: Image: Store / Inventory-Reques	Auth Attachment	(0) Headow	(0)			Heads	(0)						
ii 🗸 /store/order/{orderld}-R	(Addi) Addeninent	s (v) [Headers i				Treaue	is (o) j						
©··✓ /store/order/{orderld}-R	Assertions (1											6
🕼 ✔ /store/order-Request 1 T													۲ 0
O v /user-Request 1 TestCase	T = 1 A V	~ *											
User/createWithArray-R	FAIL	Response SLA -> Response d	id not meet SLA 4417/1000										
O vser/createWithList-Req													
🕼 🗸 /user/{username}-Reque													
Image: Image: Antipart of the second seco													
O vser/{username}-Reque													
🕸 🗸 /user/login-Request 1 Te													
in M /urer/locout-Request 1 T))											
REST Request Properties	L												
Custom Properties	Learn about REST	Request test s	steps										

リクエストエディターにレスポンスデータがある場合、アサーションはすぐに適用されま す。 今回の場合、これが持っているものです。以前にリクエストを実行し、今までにいくつ かのレスポンスを持っています。

Test Step GET Request	API Request
Method Endpoint Resource	Parameters
GET http://petstore.swagger.io v2/pet/findByStatus	?status=available Send Add Assertion
Request Generate Values	s Response Smart Assertion
Request Raw Outline Form	XML JSON HTML Raw Outline Overview
Name Value Type status available QUERY	1::::::::::::::::::::::::::::::::::::
 ✓ ● Assertions (1) + ¹/₂ × ¹/₂ ∧ ¹ FAll Response SLA 	0
-> Response did not meet SLA 4417/1000	

ReadyAPI は、より新しいレスポンスデータを取得するたびにアサーションを再適用します。

ご覧のとおり、このケースでは、リクエストが1秒よりも長く実行されたため、アサーション は失敗しました。時間制限を変更するには、[Assertions]ページでアサーションをダブル クリックし、後続のダイアログでより大きな値を入力します。

	9
+ 1 × 2 × 4	
-> Response did not meet SLA -> Response did not meet SLA Specify the maximum response time (ms) 3000 OK Cancel	

ヒント: New Functional Test ウィザードを使用して、Response SLA およびその他のア サーションをリクエストに追加できます。

例 2 - レスポンス内容を確認

次に、レスポンスデータを確認する方法を見てみましょう。 サンプルのレスポンスボディ には JSON データ形式があるため、JSON データのアサーションを作成します。

- リクエストにレスポンスがあることを確認します。ない場合は、^{▶ Send}をク リックしてリクエストを送信し、レスポンスを取得します。「個別のリクエストを実行 する」をご覧ください。
- [Assertions]パネルで、+ をクリックして別のアサーションを作成します。
 後続のダイアログで、左側で[Property Content]カテゴリを選択し、右側で JsonPath Match アサーションを選択して、[Add]をクリックします。



これにより、アサーションプロパティダイアログが呼び出されます。

 ダイアログで、レスポンスボディからいくつかのフィールドと、このフィールドの期待 値を抽出する JSONPath 式を入力する必要があります。

Configure JSONPath Match Assertion	×
Specify a JSONPath expression and the expected result.	
JSONPath Expression	
Enter an expression to select a field	
Expected Result	
Select Content 🔻 Select From Current Test	
Enter expected value here	
Allow Wildcards	
0	Save Cancel

目的の式を入力するか、ツールバーの Select node をクリックして、後続のダ イアログで視覚的に値を選択できます。 をクリックして、次のダイアログで最初 の配列項目の名前フィールドを選択し、[OK]をクリックします。

😡 JSONPath Match Assertion				>
	😡 Select JSON		×	
JSONPath Expression	Select JSON value to validate			
		<u>-</u>		
	JSON Node	Value		
	eid	7316746567143381285		
	name	doggie		
	tags status	available		
Expected Result Select Content Select From Curr	⊕1 ⊕2			
	⊕3 ⊕4		-	
	\$[0]['name']	pression matching value		
Allow Wildcards		OK Can	cel	
0				Save

JSONPath Expression フィールドにはセレクターが含まれ、Expected Result に は現在のレスポンスデータから抽出された値が含まれます。

Configure JSONPath Match Assertion	
Specify a JSONPath expression and the expected result.	
JSONPath Expression	
\$[0]['name']	
Expected Result	
Select Content 🔻 Select From Current Test	
doggie	
Allow Wildcards	
Save Cancel]

4. [Save]をクリックして、変更を保存します。

レスポンスデータがあるため、アサーションはすぐに適用され、Assertionsページ に結果が表示されます。

Test Step GET Request												t 🖉 API	
Method	Endpoint Resou			Resource	source Parameters								
GET	http://petstore.swagger.io			/v2/pet/findByStatus			?status=available			► Se	Send Add Assertion		
Request				Genera	ate Values	Respor	nse				Sm	nart Assertion	
Request Raw	Outline Form					XML	JSON	HTML	Raw	Outline	Overview		
Name status	available	Value	QUEF	Type IY	0		1 :: [] 2 :: { 3 : 4 4 : - 5 : 6 6 : 7 8 : 9 : - 0 : 0 1 : 2 : - 3 : - 4 : 5 	"id" : "categ "id "name" "photo "st], "tags" {	65449 ory" : " : 0, me" : : "do Urls" ring" : ["id" : "name"	25, { "string" ggie", : [0, : "stri	na"		
Auth Attachments (0) Headers (0)													
 ♦ Assertions (+ ▲ × 	2)											0	
FAIL PASS	Response SLA -> Response did not JSONPath Match	: meet SLA 3251/3	000										

これで、レスポンスが変更されると、アサーションがトリガーされ、テストステップが失敗します。

■もう一度説明しますが、Select node コマンドは、エディターにレスポンスデータがある 場合にのみ機能することに注意してください。つまり、最初にリクエストを実行し、レスポンスを受信する必要があります。この例では、チュートリアルの前のステップでリクエストを実行したため機能します。

参考文献

Data-Driven Testing

次のステップ

SoapUI でデータ駆動型テストを作成する方法について説明します。

て、ReadyAPIを使用したWebサービステストの詳細をご覧ください。

Using Properties

1つのリクエストから別のリクエストに値を転送する方法を説明します。

チュートリアルは終了しました。ReadyAPI SoapUI で Web サービスの機能テストを作成

もちろん、チュートリアルでは基本的な手順のみを説明しました。これらのリンクに従っ

Property Expansions

プロパティ値を動的に挿入する方法を説明します。

する方法を理解するのに役立つことを願っています。

Authorization

SoapUIテストで承認設定を構成する方法について説明します。

Verifying Results

レスポンスの検証に使用できるアサーションについて説明します。

API Discovery

Web サービスのリクエストとレスポンスを記録する方法を説明します。

その他のチュートリアル

製品の使用を開始するのに役立つチュートリアルがいくつかあります。

Data-Driven Functional Tests

Web サービスのデータ駆動型機能テストを作成する方法について説明します。 ヒント:さまざまなデータソースからデータを読み取る方法については、他のデータ駆動型 のチュートリアルをご覧ください。

Creating Your First Virtual Service

仮想サービスの作成方法について説明します。

Creating Your First Load Test

Web サービスの負荷テストを作成して、複数のクライアントでどのように機能するかを
確認する方法について説明します。

⇒ Getting Started With Security Tests

Web サービスのセキュリティテストを作成する方法について説明します。

第2章: セキュリティ テスト

このチュートリアルでは、ReadyAPI SoapUI ツールを使ったセキュリティ テストの作成 方法について説明します。

1. セキュリティ テストの作成および実行

新規のセキュリティ テストを作成するには:

1. [File] - [New Security Test] を選択します。



– または –

ダッシュボードの [New Test] タイル内で、[All Tests] と [Security Test] をクリッ クします。



注意: 画面に表示される New Test タイルは、タイルの内容がお持ちの ReadyAPI ライセンスに依存するため、上記のタイルと異なる場合があります。 詳細については、 タイルの説明を参照してください。 2. [URL] オプションを選択します。

API Definition	URL	ReadyAPI Project
Swagger/OpenAPI, WSDL, WADL.	Location of REST API	API from an existing project
Start	Start	Start

3. 下記の URL を入力し、[Next] をクリックします。

http://www.datasciencetoolkit.org/maps/api/geocode/xml? address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false

🚾 New Security Test	×				
URL Setup					
1 Enter URL 2 Create Test					
Enter one or more URLs:					
GET http://www.datasciencetoolkit.org/maps/api/geocode/xml?address=1600+Amphitheatre+ 🛛 😣 Remove					
Add URL 4. Select the method. 3. Specify the URL.					
Learn more about security tests					
Back Next Cancel					

ヒント: 複数の URL を同時に指定することができます。別の URL を追加するには、 [Add URL] をクリックします。

- 4. ドロップダウン リストから [HTTP] メソッドを選択し [GET] を選びます。
- 5. テストのためのプロジェクトを選択します。このチュートリアルでは、ドロップダウンリスト から [New Project] をクリックし新規のプロジェクトを作成します。

😥 New Security Test	×
URL Setup	
1 Enter URL 2 Create Test	
Add your security test to a new or existing project	
<new project=""> 5. Select a project for your test.</new>	
Select scans to include in your security test	
🕑 🧐 Boundary Scan	
Cross Site Scripting	24242
🕑 灣 Fuzzing Scan	
V 🖉 Invalid Types	
SQL Injection	
V XPath Injection	
6. Select the scans you need.	
Learn more about security tests	
Back Finish Cance	el

- 6. 必要なスキャンを選択し、[Finish] をクリックします。
- これで、セキュリティ テストの作成が終了しました。セキュリティ テストを直ちに実行するか、または実行する前に変更するかを選択します。今回は、何も設定する必要はありませんので、[Run Security Test] オプションを選択します。

テストの変更を選択した場合は、あとでテストの実行を開始するには [▶] をクリックします。



ReadyAPI Secure が更新されたリクエストをサービスに送信し、レスポンスを評価します。テストの実行終了後、Secure はテストの実行結果を表示します。 [View Full Report] をクリックし、より詳しいレポートを開きます。後でテスト レポートを開く場合

は、 Report をクリックします。

9. レポートを閉じます。

次のステップでは、セキュリティ テストの結果について説明します。

テスト実行中にスキャンに失敗すると、進行状況バーがオレンジ色に変わりますが、スキャンが終了するまでテストは続行します。

SecurityTest 1 (Security Test Suite 1 -> http://www.datasciencetoolkit.org

[Transaction log] ページに切り替えます。このページには、セキュリティ テスト中に送 信されたリクエストのログがすべて表示されます。

左の列には、サービスに送信されたリクエストの情報が含まれます。 右の列には、サービスからのレスポンスと、リクエストのステータス情報が含まれます。

アサーションが失敗すると、[FAIL] というステータスが表示されます。

リクエストが予期しない結果を受け取ると、[WARNINGS] ステータスになりますが、失敗にはなりません。

[FAILED] もしくは [WARNING] ステータスが表示されているスキャンのレスポンスを 選択すると、プロパティ エディターでアラート テストが表示されます。

Se	curity Test 1 (Security Test Suite 1 -> http://www.datasciencetoolkit.	org TestCase)	ompare History	Response from HTTP Method Fuzzing (3) × Response
<u> </u>		Scop Hanacton cog Co	impare matory	→ Message
•	<u> </u>			- Alerts (1)
	r 碊 HTTP Method Fuzzing			 Valid HTTP Status Codes: Response status code:200 is not in acceptable list of status codes
s	ecurityScan 4 [HTTP Method Fuzzing]			Headers (11)
	禱 HTTP Method Fuzzing (1) [method=PATCH]	[PASS] Took 27 ms		> Table > XML
				> Doc
	灣 HTTP Method Fuzzing (2) [method=TRACE]	[PASS] Took 21 ms		
	帝 HTTP Method Fuzzing (3) [method=OPTIONS]	[WARNINGS] Took 26 ms		
	蓉 HTTP Method Fuzzing (4) [method=HEAD]	[WARNINGS] Took 37 ms		
	帝 HTTP Method Fuzzing (5) [method=DELETE]	[PASS] Took 20 ms		
	려 HTTP Method Fuzzing (6) [method=PUT]	[PASS] Took 23 ms		
	端 HTTP Method Fuzzing (7) [method=POST]	[WARNINGS] Took 36 ms		
s	ecurityScan 4 [HTTP Method Fuzzing] No Alerts, took = 190ms			
	🖉 🖉 Invalid Types			
s	ecurityScan 5 [Invalid Types]			
	🖉 Invalid Types (1) [address=true]	[FAIL] Took 23 ms		
Sh	owing 1-124 of 433 items Show all Show interval			

[Message] セクションを展開すると、このリクエストに対するサービスのレスポンスが表示されます。

次のステップでは、さらにアサーションを追加し、サービスが SLA 内で返信することを確認します。

3. セキュリティ スキャンへのアサーションの追加

セキュリティ スキャンの終了後、予期しない入力によってサービスのレスポンスに時間が かかりすぎていないことを確認する必要があります。

- 1. [Setup] タブに切り替えます。
- 2. [Boundary Scan] の [Response] をクリックします。

00:00:00			Setup	Tran
V 🔛 Geocode search (9 scans)			Summary Status:	
😥 Boundary Scan	-	Response Assertions: 1	Scanne	ed:
** Cross Site Scripting	->	Response Assertions: 2		
짝 Fuzzing Scan	->	Response Assertions: 1		

3. アサーション エディターで [+] をクリックします。



4. Add Assertion ダイアログで [SLA] – [Response SLA] を選択し、[Add] をク リックします。

🔁 Add Assertion	X
	Hide descriptions
All assertions	Search Q
Recently used Property Content	SLA
Compliance, Status and Standards	Response SLA Validates that the last received response time was within the defined limit. Applicable to
Script SLA	Script TestSteps and TestSteps that send requests and receive responses.
Security	Security
	Sensitive Information Exposure Checks that the last received message does not expose sensitive information about the target system. Applicable to REST, SOAP and HTTP TestSteps.
	Add Cancel

5. アサーションがトリガーされるまでのレスポンス タイムを指定し、[OK] をクリックしま す。

Configu	re Response SLA Assertion	\times
7	Specify the maximum response time (r	ns)
	OK Cancel	

これで、スキャン中のレスポンスが 200 ミリ秒以上の時間を要した場合、アサーションが トリガーされてセキュリティ テストは失敗となります。

次のステップ

Secure のチュートリアルはこれで終了です。いくつかのヘルプトピックを下に示します。

Security Scans

使用可能なセキュリティスキャンについて説明します。

Security Assertions

使用可能なアサーションについて説明します。

Command-Line Runner

セキュリティ テストを自動化する方法を説明します。

Reports

セキュリティ テストの結果に関するレポートの作成方法を説明します。

第3章: LoadUI スタートガイド

このチュートリアルでは、Load Test エディターのインターフェイスについて説明し、 LoadUI を使った負荷テストの作成方法を示します。このチュートリアルのセクションは、 ReadyAPI での負荷プロジェクトの作成方法、簡単なテストの実行 (<u>run a simple test</u>)、 およびテスト結果の解析 (<u>analyze the results</u>) に関する説明をします。テストはサンプ ルの SoapUI テストをエミュレートします。

 このチュートリアルは、templates を使って新規の Load Test を作成します。これには、LoadUL Pro または trial ライセンスが必要です。これらを持っていない場合は、 URL より新しいテストを<u>手動で</u>作成します。

負荷テスト エディター インターフェイス

LoadUI では、Load Test エディターを使って負荷テストを表示、編集、実行できます。 Load Test エディターのサンプル表示です。

File Edit View P	roject LoadUI Help	
Projects Dasht	board Integrations	Endpoint Explorer &
Projects Dasht Navigator Search Projects Capital Sampli APIs Capital Pro LoadUl Pro LoadUl Pro Sampup ServiceV Pro	Lead Test (Geocoding) Load Test (Geocoding) Load Test (stress test)	Cool Run Save + Scenario Image:
	(Load Scenarios panel Load Scenarios panel 0 3 0 </td
		Min Max Avg Last Count TPS Err Err%
		DataSource 0 0 0 0 0 0.00 0 0.00
		Geocode search 0 0 0 0 0 0.00 0 0.00
		DataSource Loop 0 0 0 0 0 0 0 0 0 0 0 0 0
Properties		lest case Level 0 0 0 0 0 0 0 0 0 0 0
Property	Value	
Name	Baseline Load Test (Learn about assertions
Description	Simulates three virt	
		Learn about load test properties
Custom Properties		Learn about load testing
Show Logs		

Load Test Editor ツール バー

エディターのツール バーでは、テストの継続時間 (<u>test duration</u>) を設定します。 LoadUI は、対象となる Web サーバーへのリクエストを指定したタイムアウトが発生す るまで、またはその他の理由でテストが停止するまでシミュレートします。タイムアウトが 指定されていない場合は、テストを手動で停止する必要があります。

エディターの上部にあるプログレスバー (進行状況バー) は、テストの進行状況を表示し ます。

Running フィールドには、使用中の仮想ユーザーおよび使用可能な仮想ユーザーの最 大数が表示されます。

Load Test Editor ページ

エディターにはいくつかのページがあります。

ページ	説明
Load	このページでは、テストの負荷タイプ (<u>load type</u>) および負荷プロファ イル (<u>load profile</u>) の選択、テスト結果を検証するアサーション (<u>assertions</u>) の作成などの負荷テストの設定を行います。テスト設定 に関する詳細については「Setting Up Load Tests」セクションを参照し てください。
Scheduler	負荷テスト中にシナリオを使用する時間を設定します。詳細について は「Load Scheduler」を参照してください。
Log	特定の条件に応じてリクエストを記録します。詳細については 「Transaction Log Page」を参照してください。
Distribution	リモート コンピューターでテストを実行するためのリモート エージェン トを設定します。詳細については「About Distributed Load Testing」を 参照してください。
Monitoring	テスト実行時におけるサーバーのモニタリング設定をします。 収集 データは他のテスト結果と一緒に保存されます。詳細については 「About Server Monitoring」を参照してください。
Statistics	対象となる Web サービスにおける、テスト間のパフォーマンス メト リックの変化を確認できます。テスト実行の統計情報に関する詳細は、 「Test Results」を参照してください。
Scripts	テストの開始時および終了時に使用されるスクリプトを設定します。詳 細については「Setup and Teardown Scripts」を参照してください。

 LoadUI 基本機能では、Load ページのみにアクセスできます。その他のページを 使用するには、LoadUI Pro ライセンスが必要になります。

Load Scenarios パネル

このパネルでは、負荷タイプを選択し、負荷テストのシナリオを設定できます。

[Load Type] ドロップダウン リストで、負荷テストで使用する load type を選択します。

- [VUs] LoadUI が、対象となる Web サービスにリクエストを送信する仮想ユー ザーをシミュレートします。
- [Rate] LoadUI が、対象となる Web サービスに指定した割合でリクエストを送信 します。

シミュレートした負荷を編集するには、編集する負荷シナリオをクリックし、Editor パネルの [Scenario] セクションで値を編集します。

アサーションを作成するには、アサーションを作成するシナリオまたは TestStep をクリックし、Editor パネルで作成します。

Metrics パネル

Global Metrics パネルでは、全体的なテスト結果のグラフが表示されます。リクエストの処理にかかる最大時間を確認できます。 これらのグラフに表示されるメトリックのテス トアサーションも確認できます。

Test Steps Metrics パネルは、各 TestStep の詳細メトリックを表示します。たとえば、 ステップを完了するのに要する最短および最長時間を確認できます。

また、ドロップダウン リストを使って、テスト実行中に使用した別のシナリオのメトリックに 切り替えることもできます。

これらのパネルに表示される結果は、進行中のテストのみです。過去のテスト情報、またはその他のメトリックを参照するには、Statisticsページのグラフを使用します。

Inspector パネル

シナリオ、TestCase、または TestStep レスポンスをクリックすると、Inspector パネル が表示されます。このパネルでは、選択したオブジェクトを編集できます。たとえば、負荷 シナリオの負荷プロファイルを設定したり、TestStep レスポンスにアサーションを追加し たりできます。

1. 新規の負荷テストの作成

負荷テストを LoadUI で作成するには:

1. [File] – [New] – [Load Test] を選択します。



– または –

ダッシュボードで、[New Test] タイル内の [All Tests] を選択し、[Load Test] をク リックします。



注意: 画面に表示される **New Test** タイルは、タイルの内容がお持ちの **ReadyAPI** ライセンスに依存するため、上記のタイルと異なる場合があります。 詳細については、タイルの説明を参照してください。

2. **[URL]** を選択します。



ヒント: 現在のワークスペースに他のプロジェクトがある場合、ReadyAP は既存のプロジェ クトのテストケースからテストを作成することを提案します。 このチュートリアル では、URL から負荷テストを作成する必要があります。

3. 下記の URL を入力し、[Next] をクリックします。

http://www.datasciencetoolkit.org/maps/api/geocode/xml? address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false

ew Load Test			×
	URL Se	etup	
	1 Enter URL	2 Select Template	
ter one or more URLs	5.		
eocode/xml?ac	ldress=1600+Amphitheatre+Parkw	vay,+Mountain+View,+CA&sensor=	false 🗙 Remove
Add URL			

Learn about creating a load test			
	Bad	ck Next	Cancel

4. [Default] テンプレートを選択し、[Configure Load Test] をクリックします。

😥 New Load Test				×
		URL Setup		
Select a template to	1 Enter U start configuring a	IRL 2 S	elect Template	
			Default Load Test Quickly create a simple load test and modify it for your needs later. Default Parameters:	
Default	Baseline	Peak		
	24 h		10 min, 10 VUs, Fixed Assertions: Time Taken - Median	
Stress	Soak	Spike		
Smoke				
Learn about creating a	load test			
		Back	Configure Load Test Car	ncel

记 New Load Test

- 他のテンプレートを使用する場合は、LoadUI Proの評価版または製品ライセン スが必要です。
- 5. 今回は、テンプレートをデフォルト値のままにし、[Finish] をクリックします。

😡 New Load Test		×
Set Up	Load Test Based on the Template "Default"	
Configure the load tes	t	
Load test name:	Load Test	
Configure the simulate VUs: 10 Tip: 10 simultaneo	ed load: Simultaneous us VUs is a good starting point.	
Configure assertions f	or your test:	
Max response time:	100 ms 100 ms is a good starting point.	
	Skip assertions for now	
Learn about creating a	load test	
	Finish Cancel	

 LoadUI が負荷テストを作成し、[Run Load Test] または [Add Local or Cloud Agent] を選択する画面が表示されます。テストを実行する前にこのウィンドウを閉じ、 負荷テストを変更することができます。このチュートリアルでは、ウィンドウを閉じ、テス トを開始する前にテストを変更します。



これで、作成した負荷テストの設定または実行ができます。

デフォルトでは、新規作成した負荷テストは選択したテンプレート設定を使用します。



• テスト継続時間を1分に変更するため、負荷時間フィールドに1を入力します。

LoadUl Test ▶ पूर []	m ▼ 00:00:00	00:01:00
Load Allocation: R VUs 10 sim	elative ▼ Load Type: VUs ▼ ultaneous	• (1)
	▼ New Scenario	⊢ ● (0)
100 % (10)	Targets + ⊂ × ↑ ↓	×,
	► TestCase	• (0)
	+ Add Scenario	

必要なプロセスをスタートするのに時間がかかるため、最初の数秒間はリクエストに対するサーバーのレスポンスが正しく反映されません。そのような結果を記録することを回避するために、テストのウォームアップに要する時間を指定します。

Scenarios パネルの任意の場所をクリックして Load Test エディターを開きます。

[Warm-up Time] フィールドに 5 秒と入力します。



[▶ Run] をクリックしてテストを実行します。

テストの実行には、ユーザーの操作は不要です。LoadUI は、基になる SoapUI の TestCase の TestSteps をシミュレートします (エディターの **[Targets]** セクションで 名前を確認できます)。対象となる Web サービスにリクエストを送信し、レスポンスを 取得します。テスト開始後には、<u>テスト結果が表示</u>されます。LoadUI では、テスト終了 後だけでなく、テスト実行中にも結果が表示されます。

Load Allocation: Re VUs 10 sime	elative 🔻 Load Type: VUs 🔻 🖲 (1) ultaneous
	▼ New Scenario
100 % (10)	Targets + ⊂ × ↑ ↓
	► TestCase (0)
	+ Add Scenario

テストの実行が終了すると、LoadUI は Confirmation ダイアログを表示し、エ ラーの確認、レポートの作成、または統計の調査を提案します。 このチュートリ アルでは、手動で行うため、ダイアログを閉じます。



3. テスト結果の表示

LoadUI が負荷テストを実行中に、サービス パフォーマンスのデータが収集されます。 収集されたデータは、Global Metrics グラフ、Test Steps Metrics テーブル、および Statistics ページで表示されます。



[Global Metrics] グラフでは、対象となる Web サービスのさまざまなパフォーマンス メトリックの経時的な変化が表示されます。VUs/s (1 秒ごとにサーバーに到達するユー ザー数)、Time taken (各リクエストに要した時間)、TPS (1 秒あたりのトランザクション 数) および BPS (バイト/秒) メトリックを分析し、対象となるサービスのパフォーマンス情 報を得られます。

テスト中にエラーが発生すると、LoadUI はグローバル メトリックグラフの下にメッセージ を表示します。メッセージをクリックして、LoadUI Log タブでエラーの詳細な説明を表示 します。



[TestSteps Metrics] テーブルは、各ステップをシミュレートするのに要した最長および 最短時間など、個々の TestSteps におけるパフォーマンス値を表示します。

テストの他の統計を表示、確認するには Statistics ページに切り替えます。この操作に は LoadUI Pro のライセンスが必要となります。 <u>エクセルソフト株式会社 Web サイト</u>か ら評価版をお申込みください。

ヒント: このページでは、サーバーのモニターを設定し (<u>configure server monitors</u>)、 サーバー メトリックを解析することもできます。



1 秒あたりのリクエスト送信数を表示するグラフを追加しましょう。

• 🖾 をクリックします。

- 62 -

 Add Statistic ダイアログで [Scenarios] - [ScenarioName] - [Scenario Statistics] - [Test case runs] - [Per Second] を選択し、[OK] をクリックしま す。

Add Statistics		>
Statistics group:		
New Statistics Group 1		
Available:	Search:	
Arriving VU/s Running VU Assertion Failures Failed Test Steps Failures Failures Failures Test case runs Total		
Per Second		•
Load Test Statistics -> Failures -> Per Second		
Load Test Statistics -> Test case runs -> Per Second		
Scenarios -> New Scenario -> Test Case Statistics -> Time Taken -> Median		
Scenarios -> New Scenario -> Scenario Statistics -> Test case runs -> Per Second		
Remove Remove All Move Up Move Down		
earn about statistics		
		OK Cancel

■ Search フィールドに Test case runs と入力すると、迅速に統計情報を探せます。

LoadUI の Statistics ページに新しくグラフが追加されます。選択したテストにおいて 1 秒ごとに対象となる Web サービスに送信されるリクエスト数が即座に見られます。



複数のテストを実行した場合、[Run] ドロップダウン リストから必要なテストを選択しま す。デフォルトでは、最後に実行した 5 つのテストが LoadUI 上に保存されます。

Loa	dUI Test	
	⊥ ▼ 1 m ▼	00:00:0
Stat	istics	
Run:	2019-04-26 17:17:55 🔻 😷	Ę

テスト実行中には、統計情報はリアルタイムで更新され、新しくシミュレートしたリクエスト が反映されます。

4. 負荷テストの変更

最初の負荷テストの実行に成功しました。いくつかのパラメーターを変更し、異なる負荷 プロファイルでテストを再実行してみましょう。

- 1. Load Test エディターの Load ページに切り替えます。
- [Load Type] ドロップダウン リストで、[VUs] (simulate user behavior) を選択 します。この負荷タイプ (load type) を選択すると、負荷プロファイル設定で同時に 動作する仮想ユーザーの人数を指定できます。

Load Allocation: Relative Load Type: VUs (1)		
VUs 10 sim	simultaneous	
	▼ New Scenario 上 ● (0)	
Targets 100 % (10)		
	► TestCase (0)	
	+ Add Scenario	

3. 負荷シナリオをクリックすると、Editor パネルが右側に表示されます。



4. [Load profile] ドロップダウン リストで、[Burst] プロファイルを選択します。プロ ファイルのアイコンが表示するとおりに、テスト実行中に異なる負荷をシミュレートし ます。

New Scenario • Scenario Load profile: 10	Burst 🔻	×
Base VUs: 2	Simultaneous	
Burst VUs: 1	0 Simultaneous	
Base duration: 1	s 🔻	
Burst duration: 5	s 🔻	
Wait time: 1	s 🔻	
 Assertions 		

- 5. [Base VUs] には、たとえば 5 ユーザーのように、シミュレートする仮想ユーザー の最少人数を入力します。
- [Burst VUs] では、シミュレートする仮想ユーザーの最多人数が表示されます。負荷テストの設定から計算されます。このチュートリアルでは、一つのシナリオしかないため、すべての仮想ユーザーをカウントします。たとえば、シミュレートする仮想ユーザーの最多人数を 10 ユーザーと設定します。

Load Allocation: Relative ▼ Load Type: VUs ▼ ● (1) VUs 20 simultaneous	LoadUI Test ▶ 모▼ 1	t 1m ▼ 00:01:00	00:01:00
▼ New Scenario	Load Allocation: R VUs 20 sim	Relative ▼ Load Type: VUs ▼ multaneous	• (1)
Targets		 New Scenario Targets 	inn 🕒 (0)
100 % (10) + C= × ↑ ↓ ► TestCase ● (0)	100 % (10)	+ ⊂ × ↑ ↓	• (0)
+ Add Scenario		+ Add Scenario	

- LoadUIの基本バージョンでは、最高 10 人までの仮想ユーザーを同時にシ ミュレートできます。さらに多くの人数をシミュレートするには、LoadUI Pro の ライセンスが必要となります。評価版を無料でご利用いただけます。
- 7. [Base duration] には、たとえば 1 秒のように、LoadUI が最少人数の仮想ユー ザーをシミュレートする時間を入力します。
- 8. [Burst duration] には、たとえば 5 秒のように、LoadUI が最多人数の仮想ユー ザーをシミュレートする時間を入力します。
- 9. [Wait time] には、たとえば 1 秒のように、各仮想ユーザーのテストを再実行する までの LoadUI の待ち時間を入力します。

再実行の必要な理由: 何人かの仮想ユーザーのシミュレーションが終了すると、同時使用ユーザーの人数が減ります。この人数がプロファイルに指定された人数より 少ない場合、LoadUI は必要人数に達するまで新規の仮想ユーザーを増やします。

10. テストを実行します。テスト実行中に、LoadUI で指定された最多人数に達するまで 仮想ユーザーの人数が増えます。その後、LoadUI はテストが終了するまで仮想 ユーザーの人数を増減し続けます。



11. テスト結果をもとに、対象となる Web サービスのパフォーマンスの負荷下における 変化を確認します。

LoadUI ツール バーの [Add Scenario] もしくは [+ Scenario] をクリックしてテストに シナリオを追加します。このシナリオは最初のシナリオと並行してシミュレートされるため、 複雑な負荷形状を作成できます。上記と同じ方法で設定してください。

€ ∋ ▶ Ru	in 🗟 Save 🕂 Scer	nario	💥 JIRA
Load Test			
▶ 묘▼ 1	m 🔻 00:00:00	00:01:0	00 Running: 0/o
Load Allocation: R	elative 🔻 Load Type:	VUs 🔻 🔍 (1)	Global Metr
VUs 10 sim	ultaneous		1
	▼ New Scenario	⊢ ● (0)	
100 % (10)	Targets + ⊂ × ↑ ↓		SUV
	► TestCase	• (0)	
I	+ Add Scenario		o ــــــ
			Avg

5. 複数のシナリオの実行

いくつかの TestCases を並行してシミュレートするには、負荷テストにシナリオを追加し ます。Load Scenarios パネルの [Add Scenario] ボタンか、LoadUI ツール バーの [+ Scenario]をクリックします。「LoadTests の変更」トピックでの説明と同じ要領で新しい シナリオを設定すると、LoadUI でシナリオが並行してシミュレートされます。

🕞 🎯 🕨 R	un 🗟 Save 🕂 Scenario 📄 Report	💥 JIRA
Load Test		
▶ 🖵 ▼ 1 m ▼ 00:00:00 00:01:00 Running: 0/∞		
Load Allocation: R	elative 🔻 Load Type: VUs 🔻 🌒 (1)	Global Metri
VUs 10 sim	ultaneous	1
	▼ New Scenario ⊨ ● (0)	
100 % (10)	Targets	SUV
/8 (10)	+ G × T ↓	
	► TestCase (0)	
	+ Add Scenario	o_ _

いくつかのシナリオを同時に使用する場合、各シナリオに個別に負荷を割り当てるのでは なく、1 つのフィールドから負荷を割り当てることができます。[Load Allocation] ドロッ プダウン リストから [Relative] (percentage of VUs) を選択します。

下の新しい VUs フィールドで、たとえば 10 などのように、シナリオが参照するベースと なるユーザー数を指定します。

Load Allocation: Relative 🔻	Load Type: VUs 🔻	• (1)
VUs 20 simultaneous		

各シナリオの隣のフィールドに、シミュレートされるベース ユーザーの比率を入力します。

- 最初のシナリオでは、負荷を 200% に設定します。これで、このシナリオの Burst プロファイルは 20 人の仮想ユーザーでシミュレートします。ベースとなる仮想ユー ザー数はシナリオで設定されます。
- 2 つめのシナリオでは、負荷を 50% に設定します。これで、このシナリオは 5 人の仮想ユーザーでシミュレートします。

Load Allocation: F	Relative ▼ Load Type: VUs ▼ nultaneous
	New Scenario
200 % (40)	Targets + ⊂ × ↑ ↓
	► TestCase
	 New Scenario 1 Targets
50 % (10)	+ (= × ↑ ↓
	TestCase
	+ Add Scenario

6.負荷テストへのアサーションの追加

アサーション (<u>Assertions</u>) とは、テスト実行中に対象となる Web サービスの機能を確認するテスト結果に適用される検証規則です。

1 つの TestStep の平均シミュレーション タイムが制限時間を超過しないことを確認す るアサーションを作成しましょう。

 TestCase を広げ、アサーションを作成する TestStep の [Response] をクリックし ます。[Request] エディターが表示されます。




次の Add Assertion ダイアログの [Statistics:] では [Time Taken]、[Type:] では [Median] をそれぞれのドロップダウン リストから 選択します。

Name	TimeTaken - median						
General							
Statistics:	TimeTaken 🔻						
Туре:	Median 🔻						
Constraint							
Minimum:	0						
Maximum:	100						
	Stop test on failure	2					
Tolerance							
Max errors:	0						
Within period:	0		seconds				
Learn about assertions							

- 4. その他のアサーション パラメーターを設定します。
 - [Maximum] エディット ボックスに TestStep が平均して要する最多回数を入 力します。
 - [Tolerance] セクションで、エラーをログに記録する前にアサーションがトリ ガーされる最多回数を設定します。

LoadUI Test											Request 1 ×
▶ 묘▼ 1	m 🔻 00:01:00		00:01:00	Running: 0/250	Load Sc	heduler	Log Distributi	ion Monitoring	Scripts	Statistic	s 🖾 Edit Test Step
Load Allocation: R	elative 🔻 Load Type: VUs 👻 🔍 (1)	Global Metrics							All Sc	enarios .	*Assertions
VUs 10 sim	▼ New Scensic ISI ● (2) Targen + (2) × ↑ ↓ ▼ TestCate ● (2) Image: 1 → Response ● (1) ▼ New Scensic 1 上 ● (2) Targen + (2) ↓					5 · · ·		i i i i i i i i i i i i i i i i i i i		-764 -640 -522 Time -334 den -235 -138	Yea Constraint Enabled Constraint Statistics Two-Taken → Type Maddan → Constraint Monimum 100 Constraint Monimum 100 Cherrance Madean → Step test on failure Tolerance Monimum 0 Step test on failure Tolerance Monimum 0 Step test on failure Tolerance Monimum 0 Step test on failure
	+ Add Scenario	Learn about global metrics	Failures 🗌	🕒 Failures/s 🗌 🔘	Max 🗌 🔘 Mir		lueued 🗌 🔵 TPS	🗹 😑 Time Taker	. ₹ ⊖ v	Us	
		Test Step Metrics				New	Scenario: http://w	ww.datascienceto	iolkit.org	Te 🔻	
			Min	Max Me	edian I	Last	Count	TPS Er	r	Err %	
		Request 1	0	296	30	0	595	0.00	0	0.00	
		Test Case Level	22	1039	47	986	803	0.00	0	0.00	
Learn about load	I testing	Learn about test step metrics									Learn about test step properties

5. [OK] をクリックすると、LoadUI の負荷テストにアサーションが追加されます。

 再度テストを実行します。レスポンスに時間がかかりすぎると、LoadUI はエラーをレ ポートします。アサーション エラーは、Global Metrics グラフおよび TestSteps Metrics テーブルで表示されます。



次のステップ

LoadUI のチュートリアルはこれで終了です。このスタートガイドが LoadUI の理解に役 立つことを願っています。いくつかのヘルプ トピックを下に示します。

Server Monitoring

テスト実行中のサーバーのモニター方法について説明します。

Creating Distributed LoadTests

リモート コンピューターで実行するテストの作成方法を説明します。

Creating Distributed Cloud Tests

クラウド マシンで実行するテストの作成方法を説明します。

Exporting Test Results

LoadUI テストの結果をエクスポートする方法を説明します。

Transaction Log Page

シミュレートされたリクエスト情報のログの仕方について説明します。

Testing Scenarios

負荷シナリオの概要と、ストレス テストの作成など、LoadUI でシミュレートする方法を説 明します。

第4章: ServiceV スタートガイド

このチュートリアルでは、仮想サービスの作成、設定、実行、および使用方法について説 明します。

基本概念の理解

仮想サービス

仮想サービスは、実際の Web サービスのモックです。 サービスが受信するリクエスト と、リクエストに応答するためにクライアントに送信するレスポンスを定義します。 仮想 サービスは、実際の Web サービスとして機能します。 特定の URL とポートで実行され、 リクエストを送信でき、レスポンスで応答します。 ただし、仮想サービスはそれ自体を実 行することはできず、 ReadyAPI または VirtServer (リモートマシンで仮想サービスを実行 するには後者が必要) のランナーが必要です。 ServiceV では、仮想サービス**仮想 API** も呼び出します。

プロジェクトには、必要な数の仮想 API を作成できます。 プロジェクトのノードの下のナビ ゲーターパネルでそれらを表示できます (下の画像を参照)。

オペレーション (アクション)

仮想サービスを作成するとき、サービスが受信する着信メッセージ (リクエスト) とそれら に対するレスポンスを定義します。これは、仮想サービスエディターで行います。

Navigator	×	ⓒ ⊙ ▶ 🗟 + Service + Action ∅ ⊘	£ & 😤		🗘 🗐 🔛 Default environm	ient 🔹 📑 Proxy •	\$
Search	Q	Virtual REST Service	Mittail DECT Condex				
ojects DataDrivenVirt		00:00:00 Received: 0 Sent: 0 URL: http://localhost:808	6/ Setup Behavior Transact	ion Log Auth DataSources	✓ Info	lassiliest	×
Virtualization Sam O ·APIs O ·SoapUI Pro ··LoadUI Pro	nple Project	Route Options None	Searc	All	Time running: Status: Received:	00:00:00 Not running 0	
ServiceV Pro	Service	incoming	Dispatch Style	Outgoing	Sent: Protocol:	0	
Virtual REST S	iervice <mark>es</mark>	Request	Configure Dispatch settings select	Item 1 Response I / P Item 2 Response I / P Item 3 Response I / P	Theorem	https SSL Settings	
			a response for the reply	Item 4 Response I / P Item 5 Response I / P Invalid Session Id Fault I / P	Port: Autostart on server:	8086	_
	Operations (actions)	Duplicate VirtAction x Remove VirtAction		Unknown Response WildCard Response + Add Response	Virtual service URL: Max Routed Response per Action: Assertions (0)	/ http://localhost:8086/	
REST Virtual Service Prope	rties	GET login	Script	OK Response 👔 🖌 📭 Invalid Login Fault 👔 🖌 📬	Assertion Results (0/0) Description Properties Start Societ		
Property	Value			Already Logged in Fault 🔋 🖍 📫	> Stop Script		
Name	Virtual REST Service	Duplicate VirtAction x Remove VirtAction		+Add Response	OnRequest Script		
Description		•			AfterRequest Script		
Path	1	GET logout	Scrint V	OK Besponse	 DataSources 		
Port	8086		-	Invalid Session Id Fault			
Response Property Expa	true	Duplicate VirtAction x Remove VirtAction	Configure	+Add Response			
Custom Properties							
Change I and							

着信要求と発信応答の組み合わせをオペレーション (操作またはアクション) と呼びま す。ご覧のとおり、リクエストには複数のレスポンスを定義できます。

仮想サービスエディターでオペレーションを手動で指定できます。または、サービスに OpenAPI、Swagger、WADL、またはWSDL仕様がある場合は、ReadyAPIにコマンドを 実行して、仮想サービスの作成時にこの仕様からオペレーション定義を読み込むことが できます。サービス操作のリストを取得するもう1つの方法は、ライブサービスへのトラ フィックを記録することです。ReadyAPIは要求に関する情報を取得し、仮想サービスに操 作を作成します。

仕組み

 仮想サービスがリクエストを受信すると、リクエストに一致する仮想オペレーション を見つけます。オペレーションを見つけるために、仮想サービスはさまざまなリク エストプロパティを分析します。正確なアルゴリズムはサービスタイプによって異 なります。SOAP 仮想サービスは URL のリクエストされたリソースパスをチェック し、REST 仮想サービスはリクエストタイプ(GET、POST など)と URL パスをチェッ クし、JMS 仮想サービスは JMS セッションをチェックし、トピック(またはキュー)。 詳細については、「仮想サービスの詳細」を参照してください。

サービスがオペレーションを見つけると、処理を続行します。それ以外の場合 は、「500 Internal Server Error」エラーを返します。 仮想サービスがオペレーションを検出した後、着信要求に対するレスポンスを選択する必要があります。サービスは、オペレーションに指定した Dispatch 設定に基づいてレスポンスを選択します。選択は、リクエストパラメータ、リクエストの内容、またはその他の方法に基づきます。ディスパッチ設定を参照してください。

仮想サービスは、いくつかのライブ Web サービスにリクエストを転送し、レスポン スを受信してリクエスト送信者に返送できます。 つまり、クライアントとライブサー ビスの間の透過的なプロキシとして機能できます。 このプロセスは**ルーティング** と呼ばれます。 すべてのスクエストまたは一部のリクエストのみをルーティングす るように仮想サービスを構成できます。 このチュートリアルでは、リクエストを ルーティングしません。 必要に応じて、仮想サービスに対してこれを実行できま す。 詳細については、リクエストのルーティングをご覧ください。

その他

- レスポンスエディターでレスポンスの内容を指定します。レスポンスには添付ファイルを含めることができます。プロジェクトやサービスのプロパティを挿入したり、 Excel シート、データベース、CSV ファイル、その他のデータソースからデータを読み込んで挿入したりすることもできます。
- サービスエディターページの簡単な説明を次に示します。

RESTMaskSaniaa		Displays the simulation results				Set up data sources that the virt will use.		
00:00:00 Received: 0	Sent: 0 URL: http://localhost:8084	4/ Se	tup Behavior	Transaction Log	Auth	DataSources		
Route Options None	Settings for simulating error connection bandwidth and	rs,	Settings fo autho	r simulating rization		Coverage settings (SOAP only)	-	
Incoming	server performance.	Dispa	tch Style	Outgoir	ig			

次に、仮想サービスを作成しましょう。

1. 仮想サービスの作成

仮想サービス (virt、あるいは仮想 API) を作成するには:

To create a new virtual service (virtual service or virtual API):

1. メイン メニューから [File] – [New Virtual API] を選択します。



- または -

ダッシュボードの <u>New Test</u> タイルで、**All Tests** を選択して、**Virtual API** をクリッ クします。



注意: 画面に表示される新しいテストタイルは、タイルの内容がお持ちの ReadyAPI ライ センスに依存するため、上記のものと異なる場合があります。 詳細については、 タイルの説明を参照してください。 次の New Virtual API ダイアログで、オペレーションを作成する方法を選択します。 [REST Resources] の下の [Start] をクリックします。これにより、オペレーションを手動で作成します。

API Definition	Recorded Traffic	REST Resources
Swagger/OpenAPI, WSDL, WADL.	HTTP traffic to/from live API.	Enter REST resources manually.
Start	Start	Start

API の Swagger、WADL、または WSDL 仕様を持っている場合は、[API Description] をクリックします。この場合、ServiceV では仕様から読み込んだ 情報に基づきオペレーションを作成します。または、[Record Traffic] を選択し、 次のウィンドウで現行の API にリクエストを送信します。 ReadyAPI ではトラ フィックを記録し、記録したリクエストとレスポンスに基づきオペレーションを作成し ます。詳細については、「Recording Virtual Services (Discovering)」を参照してく ださい。

次のダイアログでは、オペレーションを指定し仮想化します。ドロップダウン リスト内で選択されている [GET] オペレーション タイプのデフォルト値をそのままにし、エディット ボックスに [/login] と入力します。この際、必ず名前の前にスラッシュ (/) を入れます。

😼 New Virtual API	×
Configure REST Resources	
1 Enter REST Paths 2 Create Virtual A	API
Enter REST operations to virtualize:	
GET V /login	Remove
Add another resource	
Learn more about creating virtual APIs	
Back	Next Cancel

クエリのパラメーターを入力しないでください (これらは、URL パスで ? の後に続きます)。 オペレーション パスにはこれらのパラメーターを含めないでください。

仮想サービスエディターではいつでもオペレーションを追加できます。(詳細については、 「<u>Creating Requests (Operations)</u>」を参照してください)。 [Next] を選択します。

4. 新規の仮想サービスの追加先のプロジェクトを選択します。デフォルトで選択され ているドロップダウン リストの [New Project] のままにし、[Finish] をクリックし ます。

New Virtual API	×
Configure REST Resources	
1 Enter REST Paths 2 Create Virtual API	
Add your virtual service to a new or existing project	
<new project=""></new>	
Learn more about creating virtual APIs	
Back Finish Cancel	

5. ReadyAPI は仮想サービス(virt) を新規作成したプロジェクトに追加します。その 後、次の作業の選択を行います。

Confirmation		\times					
Your Virtual API was successfully created							
What would you like to do next?							
[]]						
Run Virtual Service	Modify Virtual Service						
Start the virtual API and send requests to it.	Edit the virtual API						
Run	Modify						

作成したオペレーションでは、レスポンスが送信されないため、実行する前にサービスを 編集します。[Modify] をクリックします。

仮想サービスエディターが表示されます。

Navigator	×	ⓒ ⊙ 🕨 🗟 + Service + Action Ø ⊘ û 🍐 💥		🗘 🗟 🐰 Default enviro	nment 🔹 📰 Proxy 🕶	8
Search trojects	٩	REST Virtual Service 1 00.00.00 Received: 0 Sent: 0 URL: http://localhost:8008/ Setup Behavior	Transaction Log Auth DataSources	REST Virtual Service 1		×
APIs SoapUI Pro LoadUI Pro ServiceV Pro REST Virtua	I Service 1	Rude Options Neore Incoming Dispatch Style	Search All July Outgoing	Location: Time running: Status: Received: Sent:	localhest 00:00:00 Not running 0	
		GET /login Sequence Sequence Sequence	Response 1 🔋 🖍 🗳 + Add Response	Protocol:	http https SSL Settings	
				Port: Autostart on server: Path: Vistual service URL: Max Routed Response per Action: > Assertions (0) > Assertion Results (0/0) > Description	8088 / /tg://localhost.8088/ 10]
REST Virtual Service Pro	perties			Properties		
Property	Value			Start Script		
Name	REST Virtual Service 1			> OnRequest Script		
Description				AfterRequest Script		
Path	1			DataSources		
Port	8088					
Response Property Ex	true					
Custom Properties						

次のステップでは、仮想サービス機能の設定およびレスポンスの内容を定めます。

2. サービス レスポンスの設定

仮想サービスエディターでは、サービスモックのさまざまな側面を構成できます。 デフォ ルトでアクティブになっている Setup ページで、オペレーションを作成および削除し、レス ポンスを構成できます。オペレーションごとに、1 つまたは複数の応答を定義できます。 仮想サービスは、これらのレスポンスのいずれかを返します。 返されるレスポンスを指 定するには、Dispatch 設定を使用します。 次のステップでそれらと協力します。 それで は、GET /login オペレーションのレスポンスを作成しましょう。

Navigator	×	€ ⊙ 🕨 🛅 + Service + Action Ø ⊘ 🍰 🚣 💥		🗘 🗐 👷 Default environ	ment 👻 🚰 Proxy 🕶	\$
Search Projects Project 1	۹	REST Virtual Service 1 Constraints of the service	ansaction Log Auth DataSources	REST Virtual Service 1		×
- APIs - SoapUI Pro - LoadUI Pro - ServiceV Pro - REST Virtua	Il Service 1	Route Options S Hene Dispatch Style	aarch 🛛 🗐 🗸 🗸	Location: Time running: Status: Received: Sent:	localhost 00:00:00 Not running 0	
		GET /login Sequence Sequence Sequence	 Response 1 Add Response 	Protocol:	http https SSL Settings	
				Port: Autostart on server: Path: Virtual service URL: Max Routed Response per Action: > Assertions (0) > Assertion Results (0/0) > Description	8088	
REST Virtual Service Pro	perties			 Properties Start Script 		
Property Name Description Path Port Response Property Ex	Value REST Virtual Service 1 / 8088 true			Stop Script OnRequest Script ArterRequest Script DataSources		
Custom Properties						
A Show Logs						

次のステップ

2 つの異なるレスポンスを作成します。一方は、成功した結果を送信し、もう一方はエ ラー結果を送信します。

次のステップでは、各レスポンスが示す内容を設定します。

レスポンス設定

新規の仮想 API ダイアログでオペレーションを作成した際に、ReadyAPI では 1 つの レスポンスを作成されます。デフォルトではこのレスポンスは空なので、レスポンスを設定 します。

エディター内のレスポンスをクリックします。レスポンス プロパティが右側に表示されます。

REST Virtual Service 1		Response 1 2. View response properties. ×
00:00:00 Received: 0 Sent: 0 URL: http://localhost:8088/	Setup Behavior Transaction Log Auth DataSources	Response 1
Route Options None	Search All	Response
Incoming	Dispatch Style Outgoing	 Edit Script Market (2)
GET /login	Sequence Response 1 T	 Headers (0) DataSource
Duplicate VirtAction x Remove VirtAction		

2. レスポンス名をクリックします。次のダイアログ ボックスで、LoginSuccess と入力 し、[OK] をクリックします。

							Response 1	1. Click the	response name	×
Setup B	ehavior	Transaction Log	Auth	DataSource	s		Response	1		
	Searc	h		All		•	Response			
)ispatch Styl	e	Outgoin	g				 Edit Script Headers (0) 			
Sequence		Respon	se 1		Î /	1	DataSource	2. Ente	er the new name	÷.
Configure		+Add F	Respons	e	Change	label Set n	ew label	×	1	
U				U	Logi	nSuccess				
					_		OK Cano	el		

3. Edit セクションで、下記のプロパティ値を設定します。

プロパティ	值
Http status code	200 – OK

プロパティ	値
Content Media type	Application/json
Editor	レスポンス本文の内容となる、下記のテキストを入力します。 {
	"access_token" : "1/fFAGRNJru1FTz708zhT3Zg",
	"expires_in" : 3920,
	"token_type" : "Bearer"
	"token_type" : "Bearer" }

4. 他のプロパティはデフォルト値のままにします。

8/ Setup	Behavior Transaction Log Auth Search	DataSources	LoginSuccess × LoginSuccess Response Keit
Dispatch St	vle Outgoing LoginSuccess +Add Respon	se	Http Status Code: 200 - OK
			Editor Outline

レスポンスの追加

1. [Add Response] をクリックし、もう 1 つのレスポンスを作成します。LoginFailed という名前を付けます。

	00:00:00	Received: 0	Sent: 0	URL: http://localhost:8088/	Setup	Behavior	Transaction Log	Auth	DataSources		
Route	Options	None		•			Search			All	•
	Incoming				Dispatch	Style	Outgoir	ng			
	GET /login	1		1.	Sequence Click to a	e dd a resp	LoginS +Add f onse.	uccess Respons plicate V	e irtAction x R	i 🖍 🖒 lemove VirtA	Action
				2. Enter the m	esponse r	name.	ew VirtResponse Enter name LoginFaile	e of new ed	/ VirtResponse Cancel	×	

2. [Http Status Code] を [400 – Bad Request] に設定します。その他のフィールド はデフォルト値のままにします。

		LoginFailed ×
Setup Behavior	Transaction Log Auth DataSources	LoginFailed
	Search	
	All	Response
Dispatch Style	Outgoing	✓ Edit Http Status Code: 400 - Bad Request
Sequence	✓ LoginSuccess I ✓ ^(h) LoginFailed I ✓ ^(h)	Content Media type: application/xml
	+Add Response	
	Duplicate VirtAction - y Persona VirtAction	<u> </u>
	Duplicate virtaction X Remove virtaction	<u> </u>
		Out

これでサービス レスポンスは設定されましたが、レスポンスが送信される条件を定義す る必要があり、これをするには、Dispatch 設定を定義する必要があります。次のステッ プでは、Dispatch 設定を定義します。 Dispatch Style の設定は、着信リクエストに対して仮想サービスが返すレスポンスを指定 します。 ReadyAPI は、すべての仮想オペレーションに対して多数のディスパッチストラ テジーを提供します。 仮想サービスエディターでディスパッチスタイルを選択できます。

REST Virtual Service 1			
00:00:00 Received: 0 Sent: 0 URL: http://localhost:8088/	Setup Behavior Tra	ansaction Log Auth	DataSources
Route Options None	5	Search	All
	Dispatch Style	Outgoing	
GET /login	Sequence Sequence Random Query Match XPath	LoginSuccess LoginFailed +Add Response	
	Script Parameter		

デフォルトでは、新しいオペレーションには Sequence ディスパッチスタイルがあります。 これは、オペレーションが一連のレスポンスを返すことを意味します。 Query Match およ び XPath スタイルを使用すると、リクエストの内容に応じてレスポンスを選択できます。 パラメータースタイルは、着信リクエストのパラメーター値に応じてレスポンスをカスタマイ ズするのに役立ちます。 使用可能なディスパッチストラテジーとそのプロパティの詳細に ついては、「ディスパッチ レスポンス」を参照してください。

次のステップ

パラメーター ディスパッチ ストラテジーを使用し、下記の仮想サービス動作を実行します。

• 着信要求が下記のパラメーターを持つ場合は―

パラメーター	値
response_type	code
client_id	12345

• それ以外の場合、サービスは LoginFailed レスポンスを送信します。

ディスパッチ プロパティの設定

- Get /login 仮想オペレーションの場合は、[Dispatch Style] ドロップダウン リストから [Parameter] を選択し、[Configure] をクリックします。
- 2. [Default Response] を [LoginFailed] にします。
- 3. [+] をクリックし、新規ルールを追加します。[Rule Editor] が表示されます。
- 4. 下記の値を設定します。

REST Virtual Service 1 000000 Received: 0 Sent: 0 URL: http://localhost:8088/ Route Options Incoming Dispatch Style	havior Transaction Log Auth DataSources	Dispatch Style Dispatch Strategy: Sequence Random Query Match Script @ Parameter
GET /login GET /login Select the dispatch style and click Configure Duplicate VirtAction Remove VirtAction Parameter Configure	LoginSuccess	2. Select the default response. Default Response LoginFailed + ★ ℤ ^ ~ Rule 3. Click to add a dispatching rule.
Click to add conditions to the IF block.	Pule Editor Name Rule 1 IF Query Parameter: response_type Equal to AND Query Parameter: client_id 4. Specify the following condition choose the desired response. THEN Send Response: LoginSuccess	x vcode x v12245 x vns and

5. [OK] を選択し、変更を保存します。

これで、ディスパッチは設定されました。仮想サービスは実行およびテストが可能な状態 となりました。

4. 仮想サービスの実行およびリクエストの送信

仮想サービスの設定が完了すると、プロジェクトを実行およびテストできます。

仮想サービスの開始

コンピューターまたはネットワーク上の別のマシンで仮想サービスを実行します。異なるマ シン上で実行する場合は、<u>VirtServer</u>をマシン上にインストールします。詳細については、 「<u>Running Virtual Services</u>」を参照してください。

このチュートリアルでは、このコンピューター上で仮想サービスを実行します。 仮想サービスエディターで [▶ Run] をクリックします。

Navigator	×	©	→ I → Service + Action → Action → → →
Search Projects Project 1	Click to start your virtual service.	RE	ST Virtual Service 1 00:00:00 Received: 0 Sent: 0 URL: http://localhost:8088/
• APIs • SoapUI Pro LoadUI Pro		Rout Non	e Options e 🔹
ServiceV Pro	ual Service 1		Incoming
			GET /login

注意:同じポート上で複数の Web サーバーまたはサービスを実行することは禁じられ ています。 仮想サービスが動作しているポート上に他の Web サービス、Web サー バーがすでに存在するばあい、実行は失敗します。 サービスポートを変更し、サービス を再起動してください。また、ReadyAPI では、<u>リクエスト ルーティング</u>を使用する仮想 サービスを一度に1 つしか実行できません。<u>Virtserver</u> がインストールされたリモート コ ンピューターでは、リクエスト ルーティングを使用する仮想サービスを複数実行できます。

テスト リクエストの作成

仮想サービスの起動後、リクエストを送信します。

1. Navigator パネル内の仮想サービスを選択します。 仮想サービスプロパティを確認 し、[Virtual service URL] フィールドの値をコピーします。

REST Virtual Service 1			REST Virtual Service 1	×
00:00:00 Received: 0 Sent: 0 URL: http://localhost:8088/	Setup Behavior Trans	action Log Auth DataSources	⊤ Info	
			Location:	localhost
Route Options None	Search		Time running:	00:00:00
		All 👻	Status:	Not running
	Dispatch Style	Outaoina	Received:	0
			Sent:	0
GET /login	Parameter -	LoginSuccess 🔋 🖍 📫	Protocol:	http
		LoginFailed		https
	Configure	+ Add Response		SSL Settings
Duplicate VirtAction x Remove VirtAction		Duplicate VirtAction x Remove VirtAc	Port:	8088
			Autostart on server:	
			Path:	/
			Virtual service URL:	http://localhost:8088/
			Max Routed Response per Action:	10
			Assertions (0)	Select and copy
			Assertion Results (0/0)	to the clipboard.
			Description	
			Start Script	

 [Navigator]パネルで[API]ノードを展開します。 仮想サービスにオペレーションを追加 すると、ReadyAPI は仮想サービスの API 仕様を自動的に生成し、その仕様にオペ レーションを追加しました。

Navigator X	⊙
Search Q	REST Virt
Projects	
Project 1	00:00:59
⇔	Route Options
SoapUI Pro	Incoming
LoadUl Pro ⊡ServiceV Pro	GET /login
REST Virtual Service 1	
	Duplicate

 リクエスト エディターを開き、仮想サービスが使用するエンドポイントを指定します(下 記の画像では、http://localhost:8088/)。

Navigator	\times	ⓒ ⊙ ▶ ⓑ + Create Test ि ♀	
Search Projects Project 1 APIs Compared APIs Compared APIs Compared APIs Compared APIs Compared APIs Comp	Q	API Request Request 1 Method Endpoint GET http://localhost:8088/ Request Request Request * * *	esource /login

- 4. リクエスト パラメーターを指定します。[十] をクリックし、表にデータを入力します。以
 - 下の2つのパラメーターを追加します。

名前	値	スタイル
response_type	code	QUERY
client_id	12345	QUERY

エディターには、以下のように表示されます。

		Resource	
GET Thttp://loc	alhost:8088/	▼ /login	
Paquart			
request			
Request Raw Outlin	e Form		
+ X 🐃 Ó 🔻 🗸			0
· · · · · · · · ·			
Name	Value	Туре	
Name response_type	Value	Type QUERY	

テスト リクエストの実行

リクエストをシミュレートするには、リクエスト エディターで ^{▶ Send} をクリックします。

サービスがリクエストを処理し、LoginSuccess レスポンス内容がエディターに表示されます。

ethod Endp	oint			Resource			Paran	neters					
GET 🔹 http	://localhos	st:8088/		▼ /login			?res	ponse_ty	pe=code	e&client_id	=12345		📄 🕨 Sen
equest						Respon	se				Run	the request	16 +
Request Raw	Outline	Form				XML	JSON	HTML	Raw	Outline	Overview		
-× ≞φ	•				0	1		Ξ	<u> </u>	3			
Name			Value	Ţ	ype			JSON No	de			Value	
sponse_type	c	ode		QUERY		acc	ess_token				1/fFAGRNJru1	FTz708zhT3Zg	
ent_id	1	2345		OUERY		exp	ires_in				3920 Raaaaa		
				quen		tok	en_type	Res	ponse	body cor	ntents		
						tok	en_type	Res	ponse	body cor	ntents		

 client_id パラメーターを異なる値に変更し、リクエストをもう一度実行します。仮想 サービス ディスパッチャーによって [LoginFailed] レスポンスが選択されます。

API Request Reques	st 1		Add to TestCase
Method Endpoint		Resource	Parameters
GET Thttp://localh	ost:8088/	▼ /login	?response_type=code&client_id=6789
Request			Response
Request Raw Outline	Form		XML JSON HTML Raw Outline Overview
+ × 🕾 🗘 🔸 🔸			HTTP/1.1 400 Bad Request Action 17 (2010) 07:42:43 CMT
Name	Value	Туре	Content-Type: application/xml
response_type	code	QUERY	Server: Jetty(9.4.7.v20170914)
client_id	6789	QUERY	The LoginFailed response data

リクエスト エディター内では、リクエスト パラメーターとレスポンス内容を見て、仮想サー ビスが正常に動作しているかを確認します。さらに、ServiceVの **Transaction Log** ページで、リクエストとエディターを確認することもできます。 • [ServiceV] に移動して Navigator パネルの仮想サービスを選択し、仮想サービス エディターの Transaction Log ページに切り替えます。

Navigator	×	D ⊙	
Projects	٩	REST Virtual Service 1 2. Select the Transaction Log page 0024/10 Received: 4 Sent: 4 URL: http://localhots8088/ Setup Behavior Transaction Log Auth DataSo	/login ×
APIs APIs APIs Artis Arti	te 1 n Request 1 I Service 1 e virtual service	Fiter Incoming 3. Select a request or response Outgoing 10:38:08 GET /ogin HTTP/1.1 LoginSuccess 10:41:33 GET /ogin HTTP/1.1 LoginFailed	Adecage XdM, ISON Rew Outline Overview GET Apport1791 1 Concerdent: Nersaline User-Apert: Acade-HttpClefel4 5.2 (Javar) 8.0_181) Hest: Loadhest 8000 Account-Exceeding stp.6-fielde Contents of the selected request (or response)
REST Virtual Service Pro	perties		
Property	Value		
Name	REST Virtual Service 1		
Description			
Path	1		
Port	8088		
Response Property Ex Custom Properties	true		

 ページ上のそれぞれの列はトランザクション、すなわち リクエスト/レスポンス ペアに 対応しています。

特定のリクエストまたはレスポンスを選択すると、それらの内容を確認することができま す。右側に表示されます(上記画像をご確認ください)。

リクエストまたは結果確認後、サービスエディター上の [**- Stop**] をクリックし 仮想 サービスを終了します。

次のステップ

これで、チュートリアルは終了です。この段階で、仮想 API は完全に設定されました。新 規にオペレーションを追加し、機能テスト、データ ドリブン テストまたは負荷テストを作成 できます。以下のトピックが準備されています。

仮想サービスの設定

仮想サービスの作成や設定に関する情報が含まれています。

VirtServer のチュートリアル

リモートコンピューター上で virt を実行するための VirtServer の使用方法について説明します。

機能テストの作成

SoapUI での API 機能テストの作成方法について説明します。

第5章: VirtServer のチュートリアル

このセクションのトピックでは、VirtServer の使い方、仮想 API サービス (仮想サービス) の展開、および VirtServer での実行方法などについて説明します。

1. VirtServer のインストールと実行

VirtServer を使用するには、VirtServer をインストールして、少なくとも 1 人のユーザー 登録をし、ライセンスをアクティベートします。

1. VirtServer のインストール

VirtServer をインストールするには、インストーラーを実行して画面の指示に従います。 「<u>VirtServer Installation</u>」を参照してください。

2. VirtServer の実行

VirtServer のプロセスを開始するには、次のファイルを実行します。

- Windows: <VirtServer directory>/bin/virtserver.bat
- Linux または macOS: <VirtServer directory>/bin/virtserver.sh
- ルート権限で VirtServer をインストールした場合、ルート権限で実行するか、コマン ドラインで <u>VirtServer settings</u> ファイル(virt-server.yml) の場所を手動で指定します。

3. ユーザーの作成

<u>ServiceV</u>から VirtServer を操作するには、ユーザー名とパスワードが必要です。ユー ザーの作成は初回のみで、VirtServer を起動するたびに作成する必要はありません。

バージョン 1.6.0 以降の場合

- VirtServerのインストール後初めて起動する際に、ユーザー名とパスワードを指定 する必要があります。画面の指示に従ってください。
- ユーザー登録をすると、さらにユーザーを追加できます。VirtServer を実行してコマンドラインの –a 引数を指定します:
 - Windows:
 <VirtServer directory>/bin/virtserver.bat -a userName:password

Linux または macOS:
 <VirtServer directory>/bin/virtserver.sh -a userName:password
 ヒント: パスワードを指定しない場合、VirtServer はユーザーを作成する前にパス

ワードを要求します。

VirtServer を操作するユーザーごとのユーザー設定をすることを推奨します。

バージョン 1.5.0 以前の場合

VirtServer 1.5.0 以前のバージョンでは、デフォルトでユーザーが作成されます:

User name: defaultUser

Password: svp4ever

パスワードは、VirtServer settings file (virt-server.yml) ファイルで設定できます。

4. VirtServer ライセンスのアクティベーション

VirtServer を初めて実行する場合、ライセンス ファイルを指定してライセンスをアクティ ベートする必要があります。画面の指示に従ってください。「<u>VirtServer License</u> <u>Activation</u>」も参照してください。

1. 仮想サービスの作成

仮想サービスを <u>VirtServer</u> で実行する前に、仮想サービスを作成する必要があります。 仮想サービスは、ReadyAPI の <u>ServiceV</u> で作成します。仮想サービスの作成方法に ついては、「第4章 ServiceV スタートガイド」を参照してください。

このチュートリアルでは、<u>Sample-Virtualization-project</u>の仮想サービスを使用します。 このプロジェクトは製品と一緒にインストールされており、<ReadyAPI>¥tutorials と いうディレクトリにあります。ReadyAPI でこのプロジェクトを開き、[ServiceV] に切り替 えます。[ServiceV] パネルの [Local Virtual services] セクションに 2 つの仮想 サービスが表示されます (次に示す画像を参照してください)。

2. 仮想サービスの展開

仮想サービスを作成すると(またはファイルから開くと)、VirtServer で展開 (アップロード) できます。

ReadyAPI を開いて [ServiceV] に切り替え、左のプロジェクト ツリーでプロジェクト ノードを選択します。

製品ウィンドウの中央部に VirtServer パネルが表示されます。

Navigator X	ⓒ ③ ▶ ⓑ + Service Ø ▷ ◇ ☆ ☆
Search Q Projects Project Virtualization Sample Project APIs SoapUl Pro Locatul Day	Local Virtual Services RESTMockService ServiceSoapBinding MockService Virtual services on your computer
	Transactions: 0 Errors: 0 Port: 8084 Port: 8082
	VirtServer section This VirtServer requires authentication Username: defaultUser Password: Connect

 VirtServer を選択して接続します。ドロップ ダウン矢印をクリックして使用する VirtServer を選択します。

Navigator X	ⓒ ⊙ 🕨 🖥 🕂 Service Ø 🕒 ⊘ 🏦 🕹 💥
Search Q	Local Virtual Services
Virtualization Sample Project APIs SoapUI Pro LoadUI Pro	RESTMockService ServiceSoapBinding MockService MockService
ServiceV Pro ServiceSoapBinding MockSer RESTMockService	Transactions: 0 Transactions: 0 Errors: 0 Errors: 0 Port: 8084 Port: 8082
	Gus Cocal VirtServer discovery Gus Cocal VirtServer discovery Cocal VirtServer Cocal VirtSer

[Local VirtServer discovery] チェック ボックスが選択されていると、ServiceV が ローカル ネットワークで使用可能な VirtServers を自動的に検索します。

このチェック ボックスが選択されていない場合、もしくはリストに VirtServer が存在 しない場合、[+] ボタンをクリックして次のダイアログで VirtServer の動作プロトコ ル、IP アドレス、およびポートを指定します。

3. VirtServer に接続します。このチュートリアルの <u>Step 1</u> で作成したユーザー名とパ スワードを指定し、[Connect] をクリックします。

Gus -				
URL: https://10.0.81.128:9090 Running virtual services: N/A 💻	Virtual Services	Transaction Log	Activity	Metrics
	entication			
Password: Connect				
)			

選択した VirtServer で仮想サービスが実行されます。この例では、VirtServer に は仮想サービスがまだ存在しません。

Navigator X	ⓒ ③ ▶ 🗟 + Service Ø 斗 ⊘ ⅔ 缕 ★	
Search Q	Local Virtual Services	
Projects Virtualization Sample Project APIs SoapUI Pro LoadUI Pro ServiceV Pro RESTMockService	Image: Service SoapBinding MockService Image: Service SoapBinding MockService	
	■ Gus URL: https://10.0.81.128:9090 Running virtual services: 0/2 Virtual Services Transaction Log Activity M Search on name, tag, URL, port or status	Metrics
	No virtual services detected on the VirtServer Drag a virtual service here to deploy it to the VirtServer	

 エディターの上部にある [Local Virtual Services] セクションから [VirtServer] セクションに仮想サービスをドラッグします。

Navigator X	④ ③ ▶ 圖 + Service Ø ▷ ◇ △ △ △
Search Q	Local Virts
Projects	Image: ServiceSoapBinding Image: ServiceSoapBinding
	■ Gus URL: https://10.0.81.128:9000 Running virtual services: 0/2 Virtual Services Transaction Log Activity Metrics
	No virtual services detected on the VirtServer Drag a virtual service here to deploy it to the VirtServer

VirtServer セクションで仮想サービスが表示されます。

Navigator ×	ⓒ ③ ▶ 🖥 + Service Ø ▷ ○ ⊥ 丛 ★	
Search Q	Local Virtual Services	
Projects Virtualization Sample Project APIs SoapUI Pro LoadUI Pro ServiceV Pro	RESTMockService ServiceSoapBinding 00.00.00 00.00.00 Transactions: 0 Error: 0	
₩ ServiceSoapBinding MockSer	Port: 8084 Port: 8082 Image: Constraint of the state of the stat	1 etrics
	Search on name, tag, URL, port or status	•
	Virtual service 00:00:00 Transactions: 0 Font: 2084	

 仮想サービスの展開中に NullPointerException エラーが発生した場合には、 ReadyAPI および VirtServer のバージョンをアップデートしてください。この 問題は 1.5.0 以前のバージョンで発生する可能性がありますが、1.6.0 では 修正されています。

3. 仮想サービスのポート番号の変更

仮想サービスはポートを使用します。ポートは、エディターの右側にある仮想サービスの プロパティで指定できます(次の画像を参照)。

VirtServer で仮想サービスを展開したのち、ポート番号の変更が必要な場合があります。 仮想サービスに使用するポートが、別の仮想サービスまたはプロセスにより VirtServer で使用されている場合などに変更が必要となります。

VirtServer で仮想サービスのポートを変更するには、次の手順に従ってください。

- 1. エディターの VirtServer セクションで仮想サービスを選択します。
- 2. 誤って仮想サービスを実行してしまった場合には、実行を停止します。
- エディターの右側にある サービス プロパティのポート番号を変更し、[Change port] をクリックします。もしくは、[Auto-assign a free port] をクリックして
 VirtServer が仮想サービスにポートを自動的に選択するようにします。

Local Virtual Services	RESTMockService	×
Account creation mock 0.00.00 Transactions: 0 Errors: 0 Port: 8096	 ► Info Location: Gus Time running: 00:00:00 Status: Not running Received: 0 Sent: 0 Protocol: http Path: / 	
Image: Cussion Cost of the services of the serv	Noute mode: off URL: http://10.081.128:8084/ Autostart: Port: 9094 Change port Auto-assign a free port	
RESTMockService 0 00 00 00 Transactions: 0 Errors: 0 Port: 8084	 Assertion Results (0/0) Tags (0) DataSources 2. Chang the port the port 	je L

これで、仮想サービスをスタートしてテストを実行する準備が整いました。

3. 仮想サービスの実行とテスト

1. 仮想サービスの実行

VirtServer で仮想サービスを展開して設定を終えると、仮想サービスを実行できます。 仮想サービスを実行するには、[▶] ボタンをクリックします。



VirtServer は、指定したポート番号で仮想サービスを実行します。

ヒント:「<u>VirtServer Web Interface</u>」を使用することにより、仮想サービスをブラウザーから始めることもできます。

2. 仮想サービスのテスト

仮想サービスの実行後、ReadyAPI のテスト プロジェクトからテスト リクエストを送信で きます。他の API で行うのと同じやりかたで行います。

ヒント:

仮想サービスのテストを作成する際に、VirtServer が使用するポートが不明の場合があ ります。テストにポート番号をハードコード化した場合、仮想サービスの展開後にポート番 号を変更する必要があります。回避策として、仮想サービスのアドレスおよびポート値を 保存するプロジェクト レベルもしくはテスト スイート レベルのプロパティを作成し、この 変数をテスト リクエストの Endpoint プロパティとして指定することができます。ポート番 号が変更した場合、すべてのリクエストの Endpoint プロパティをアップデートするのでは なく、プロジェクト レベル (もしくはテスト スイート レベル) のプロパティを変更する必要 があります。



次のステップ

<u>VirtServer</u>のチュートリアルはこれで終了です。下のトピックでは、よく使うタスクの実行 方法を説明しています:

VirtServer の実行

Different Ways to Run VirtServer

Running VirtServer as Service

仮想サービスの設定

Updating Virt Ports On VirtServer Editing Deployed Virts Using Virts With DataSources on VirtServer

VirtServer の設定

Managing VirtServer Users

VirtServer Settings

VirtServer Command-Line Arguments

サポート

お問合せ先

エクセルソフト株式会社

http://www.xlsoft.com/jp/services/contact.html

SmartBear

https://support.smartbear.com/product-list/