|  |
| --- |
| **System Design Specifications** |
| API Portal 7.4  for  API Gateway Server 7.4  Document Version 2.9 11-th, february 2015 | | |



Copyright © Axway Software, 2014

All rights reserved.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, Axway Software.

This document, provided for informational purposes only, may be subject to significant modification. The descriptions and information in this document may not necessarily accurately represent or reflect the current or planned functionalities of this product. Axway Software may change this publication, the product described herein, or both. These changes will be incorporated in new versions of this document. Axway Software does not warrant that this document is error free.

Axway Software recognizes the rights of the holders of all trademarks used in its publications.

# Preface

Purpose of this DOCUMENT

The purpose of this document is to list the technical requirements and the implementation details of the API Portal project.

Related documentation

This Document is related and refers to the following sources:

**APIDeveloperPortal.docx**

Intended Audience

The intended audience of this document is everyone involved in the project implementation, deployment, acceptance or management. It is expected that the reader is familiar with all base products and systems related to it as well as the targeted business requirements.

# Table of Contents

[1 Preface 2](#_Toc411442130)

[2 Table of Contents 3](#_Toc411442131)

[3 Revision History 4](#_Toc411442132)

[4 Overview 5](#_Toc411442133)

[4.1 Development Prerequisites 5](#_Toc411442134)

[4.2 Deliverables 5](#_Toc411442135)

[4.3 Assumptions 5](#_Toc411442136)

[4.4 Requirements 5](#_Toc411442137)

[5 Use Cases 6](#_Toc411442138)

[5.1 Upgrade from 7.3.1 to 7.4 6](#_Toc411442139)

[5.2 Support for External User Repositories 6](#_Toc411442140)

[5.3 Support for Web Services 6](#_Toc411442141)

[5.3.1 Use case 1: Application developer browsing Web Services in API Catalog 6](#_Toc411442142)

[5.3.2 Use case 2: Application Developer consuming Web Services in API Catalog 7](#_Toc411442143)

[5.4 OAuth 8](#_Toc411442144)

[5.4.1 OAuth 2.0 Implicit grant flow 9](#_Toc411442145)

[5.4.2 API key / OAuth client ID for testing 14](#_Toc411442146)

[5.5 Administrator UI changes: 17](#_Toc411442147)

[5.6 Support for Organization Administrator 18](#_Toc411442148)

[5.6.1 Application List 19](#_Toc411442149)

[5.6.2 Application Editing/Deletion 19](#_Toc411442150)

[5.6.3 User Management 19](#_Toc411442151)

[5.6.4 Add User 20](#_Toc411442152)

[5.6.5 User account information 21](#_Toc411442153)

[5.6.6 User account approval 25](#_Toc411442154)

[Extend Profile pages by adding image/picture support 25](#_Toc411442155)

[5.6.7 View profile 25](#_Toc411442156)

[5.6.8 Edit profile 26](#_Toc411442157)

[5.7 Support for API Administrator (Deferred for API Portal 7.4a 27](#_Toc411442158)

[6 Appendix A 28](#_Toc411442159)

# Revision History

| **Revision** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| 2.0 | 8-th Oct 2014 | Totyo Totev | First draft. |
| 2.1 | 10th Oct 2014 | Petya Karastoyanova | Added details about web services and oauth requirements. |
| 2.2 | 14th Oct 2014 | Totyo Totev | Addressing Dave’s notes |
| 2.3 | 14th Oct 2014 | Petya Karastoyanova | Updated Web services use case 2 |
| 2.4 | 15th Oct 2014 | Petya Karastoyanova | Added additional description regarding web services support |
| 2.5 | 16th Oct 2014 | Totyo Totev | Added a list of areas for upgrade |
| 2.6 | 26th Nov 2014 | Totyo Totev | User create for Org Admins section is updated |
| 2.7 | 27th Jan 2015 | Petya Karastoyanova | Updated web services and OAuth UI changes |
| 2.8 | 5th Feb 2015 | Petya Karastoyanova | Added request access token diagram |
| 2.9 | 11th Feb 2015 | Totyo Totev |  |

# Overview

in this chapter

This chapter briefly describes the features of the solution package.

## Development Prerequisites

Host Environment with

* SuSe 11.3
* Installed Apache Web Server **(**[**http://www.apache.org/**](http://www.apache.org/)**)**
* PHP 5.5 Support **(**[**http://www.php.net/**](http://www.php.net/)**)**
* Installed MySQL database Server **(**[**http://www.mysql.com/**](http://www.mysql.com/)**)**
* Installed **Joomla 3.3.1 (**[**http://www.joomla.org/technical-requirements.html**](http://www.joomla.org/technical-requirements.html)**)**
* API Gateway Server 7.4 installed and configured on the same or another machine.
* API Portal 7.3.1 will be used and upgraded

## Deliverables

This is a list of the deliverables that will be provided by Axway

* Appliance image (ISO) that can be used for building VMWare and Hyper-V deployments.
* Documentation and step by step instructions regarding how to install and configure the API Portal

## Assumptions

* The API Portal will be localized in English only.

## Requirements

* + Upgrade from 7.3.1 to 7.4
  + Support for External User Repositories
  + Support for Web Services
  + OAuth
  + Support for Organization Administrator
  + Support for API Administrator
  + API Portal Accessibility Review
  + API Portal Security review (see R&D security objectives – i.e. same what was done for API Portal 7.3 release)
  + Deferred features from API Portal 7.3.1

# Use Cases

in this chapter

This chapter covers the main use cases this solution would cover.

## Upgrade from 7.3.1 to 7.4

API Portal 7.4 must support upgrading from 7.3.1 if the customer has not modified the source code to the API Portal. If the customer has customized the API Portal via configuration in the Joomla administrator screens, then the API Portal must upgrade while maintain those Joomla customizations.

Upgrade will be prepared as an upgrade package .

Following areas will be subject of the upgrade:

* Joomla SQL Schema – the custom configuration and custom user tables will be modified
* Joomla Code – API Portal Joomla components, plug-ins and templates
* Joomla resources – API Portal related CSS and image resources

**Note:** SQL schema changes will preserve the data in tables which are subject of modifications.

## Support for External User Repositories

This section describes the use cases related to user activities. Users are the partner developers who use the portal to register and read documentation from it.

## Support for Web Services

A key requirement for API Manager 7.4 is providing support Web Services:

<https://docs.google.com/a/vordel.com/document/d/1VwMkueR3utkUGxb70yi54TvCwaeG5UmBlO_-mNPcDSs/edit#bookmark=id.dck9vn1us7rq>

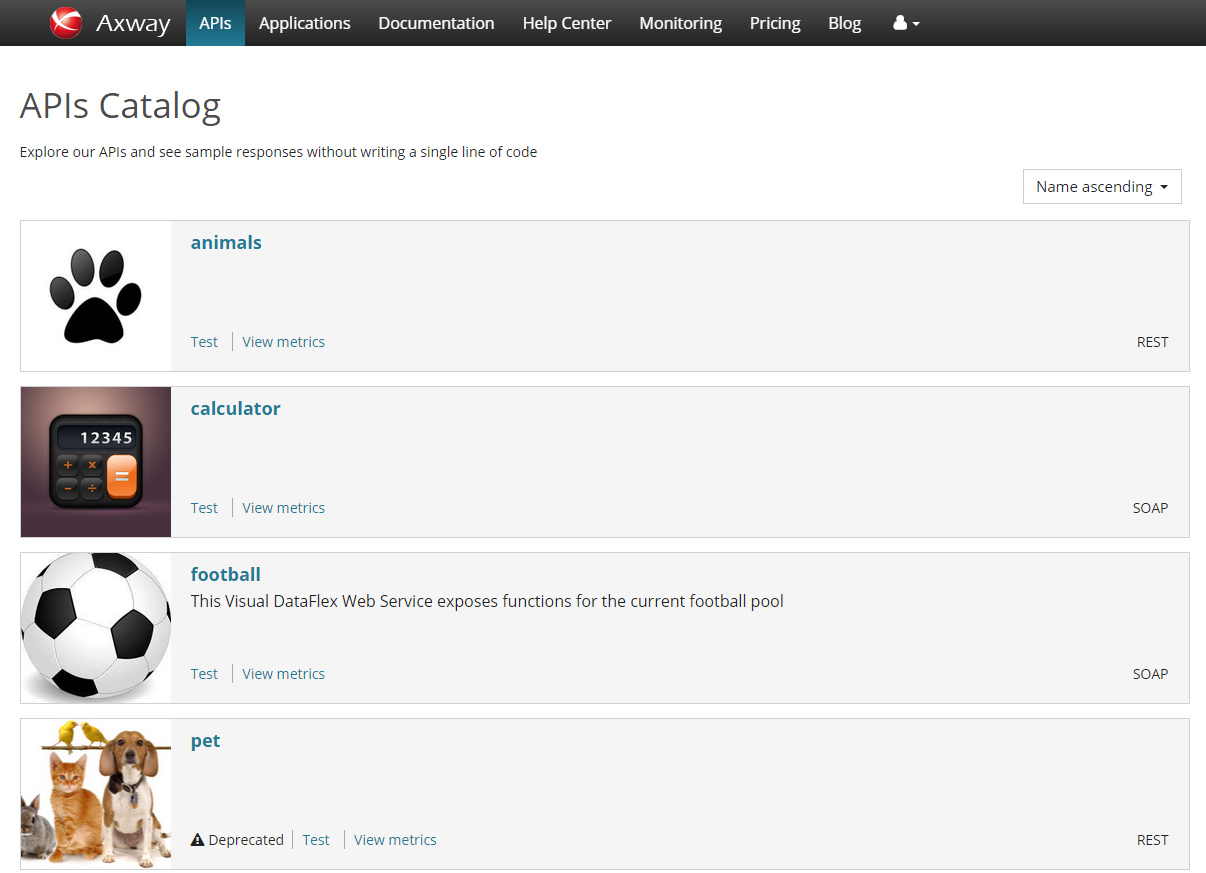
API Portal 7.4 needs to support the application developer browsing and consuming Web Services like it does for REST APIs.

API Portal 7.4 needs to support use cases 1 and 2 outlined in the above API Manager 7.4 requirements document for Web Services.

### Use case 1: Application developer browsing Web Services in API Catalog

The API Portal will allow application developers to browse the frontend Web Services that are available to them in the API Catalog.

* The frontend Web Services will be displayed alongside frontend REST APIs in the API Catalog list. The frontend Web service will not be viewable by the application developers unless the API administrator has authorized their organization access to the Web Service.
* A label will be displayed alongside to each API so that the application developer will be able to distinguish between REST APIs and WSDL services.

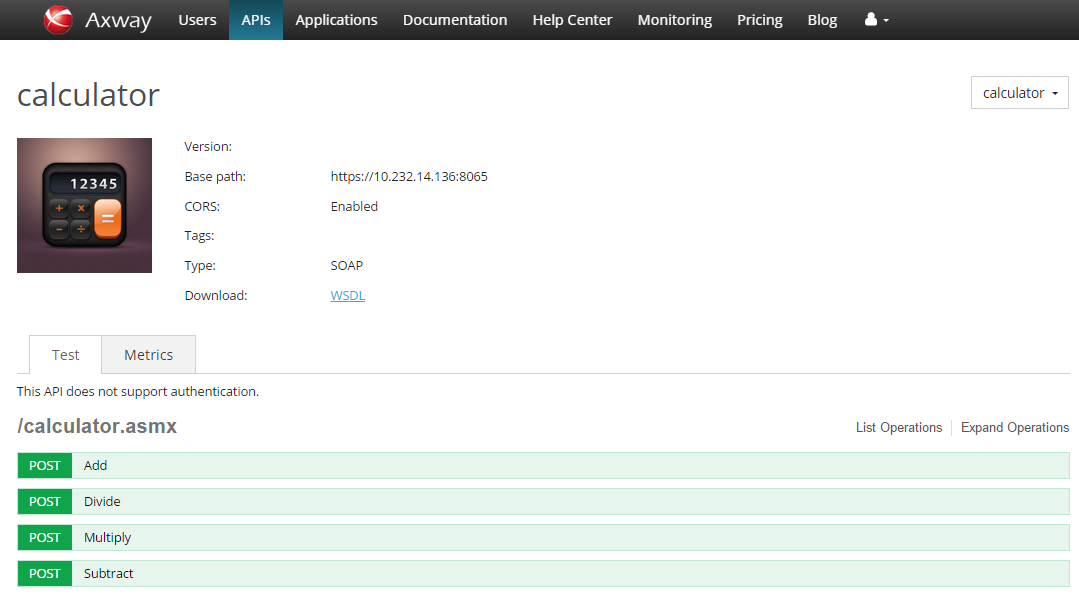


### Use case 2: Application Developer consuming Web Services in API Catalog

The API Portal will be enhanced to allow application developers to consume frontend Web Services by requesting application access to the Web Service in the same way the application developer can request application access for a REST API. Application access to the frontend Web Service will allow the application to invoke the Web Service.

The API Portal Test page will be modified to allow the application developer to invoke a Web Service. In addition when a POST method is being expanded a XML template will be populated as a body of the request.

* A link in the API info section will be added so that the application developer will be able to view/download the API WSDL. In case the API type is REST the link will be point to the swagger feed.



## OAuth

API Portal 7.3.1 allows an application developer to invoke OAuth-protected APIs using the “Test” capability. Alternatively, during development of their application, the application developer will test their application which will invoke OAuth-protected APIs. In both cases the request must include an access token that has been granted to the application so the request can be authorized.

API Portal 7.3.1 provides no mechanism for the application developer to obtain the access token using the application’s OAuth credentials. This must be done outside of the API Portal and provides a bad user experience.

API Portal 7.4 will implement the client credentials grant request flow to allow an application developer to obtain an access token for the consuming application. OAuth 2.0 Client Credentials Grant Flow is [here in our docs](https://docs.google.com/a/vordel.com/document/d/1qngYFTF1JsQoyOJTDMTCrOTkYxz3GXCefA7UaTVcYZo/edit#heading=h.kh511kpxu3ql), and a good example of what it would be like in the real world is petstore (toggle the OFF switch to request a token): <http://petstore.swagger.wordnik.com/#!/pet/deletePet>

The access token obtained can then be used in either the “Test” capability in API Portal or in the application developer’s application that they are developing.

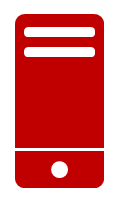
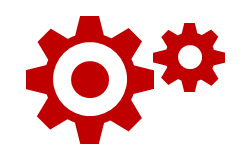
This OAuth testing capability will primarily be used during the development of the client application in the Sandbox environment. In this environment the resource owner will be a test “resource owner” provided by the API owner to application developers for this purpose.

This requirement also drives the requirement for API Manager to provide an OAuth Authorization Service API.

### OAuth 2.0 Client credentials grant flow

The OAuth 2.0 Access Token using Client Credentials filter enables an OAuth client to request an access token using only its client credentials. This supports the OAuth 2.0 client credentials flow, which is used when the client application needs to directly access its own resources on the resource server. Only the client application's credentials or public/private key pair are used in this flow. The resource owner's credentials are not required

#### Obtain an access token

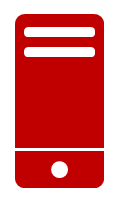
API Gateway

Authorization Server

Issues access token

API Portal

Resource Server



1. The application developer request an access token in the API Portal. The JavaScript client includes the client ID, client SECRET and requested scope(s).
2. The API Gateway Authorization server authenticates the resource owner and checks if the owner grants permissions or not to the client’s access request.
3. Assuming that the resource owner grants access, the API Gateway Authorization server send response containing the access token or error message.
4. The user-agent executes javascript which extracts the access token from the response.
5. After the access token is extracted the application developer is able to access the protected data.

#### Sample request and response URLs

* + Request an access token from API Gateway

The client token request should be sent in an HTTP POST to the token endpoint with the following parameters:

|  |  |
| --- | --- |
| Parameter | Description |
| grant\_type | **Required.** Must be set to **client\_credentials**. |
| client\_id | **Required.** The Client ID generated when the application was registered in the Axway Client Application Registry. |
| client\_secret | **Required**. The Client SECRET generated when the application was registered in the Axway Client Application Registry. |
| scope | **Required**. A space delimited list of scopes, which indicates the access to the Resource Owner's data requested by the application. |

* + API Gateway response if resource owner grants permissions to the application

Response attributes:

|  |  |
| --- | --- |
| Parameter | Description |
| access\_token | The token that can be sent to the Resource Server to access the protected resources of the Resource Owner (user). |
| expires\_in | The remaining lifetime on the access token. |
| token\_type | Indicates the type of token returned. At this time, this field will always have a value of Bearer. |

After the API Portal obtains an access token then it will access protected resources on the Resource Server by placing it in an Authorization: Bearer HTTP header.

* Request token flow diagram

The following diagram illustrates the user flow of obtaining an access token from the API Gateway server. Access token is extracted and returned to the portal. This behavior is controlled by the user and his preferences for a given application.

User request an access token. The request is sent to API GW

API Gateway return a response with access token or error message

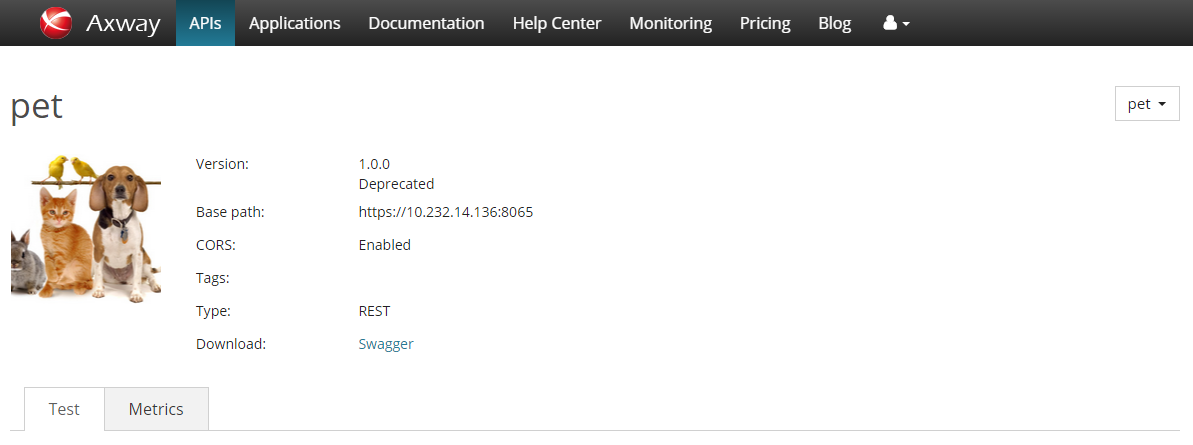
The Portal extracts the access token or error and pass it to the API Test Page

#### Request an access token UI changes

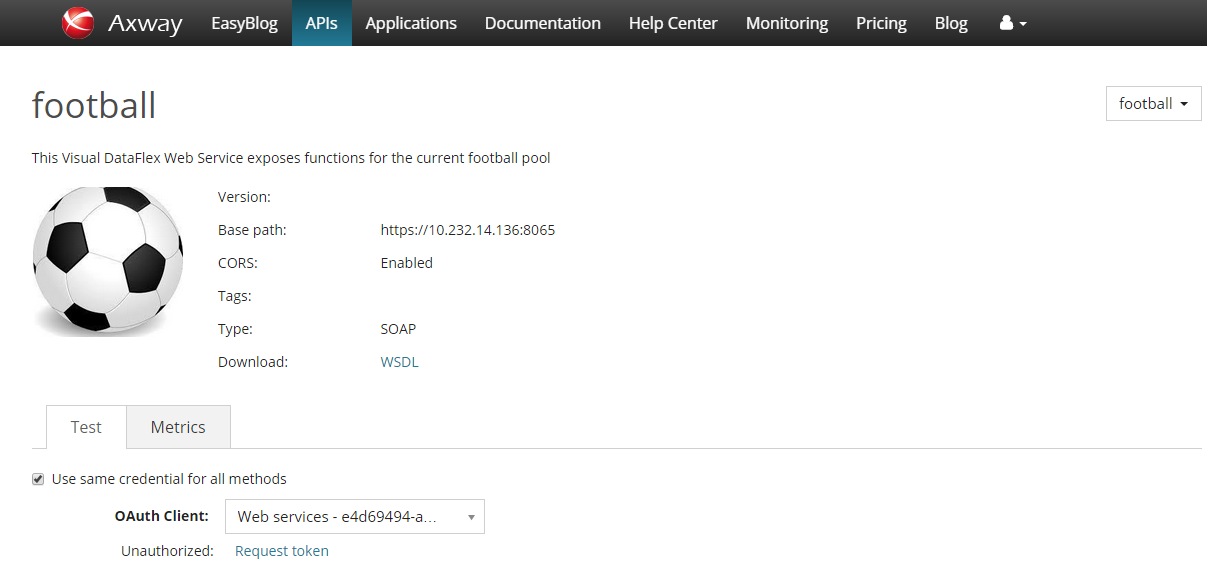
#### The following end user UI changes will be implemented to handle OAuth 2.0 authentication and in addition Joomla API Portal configuration UI will be extended in order to allow configuring Oauth service URL.

End user UI changes:

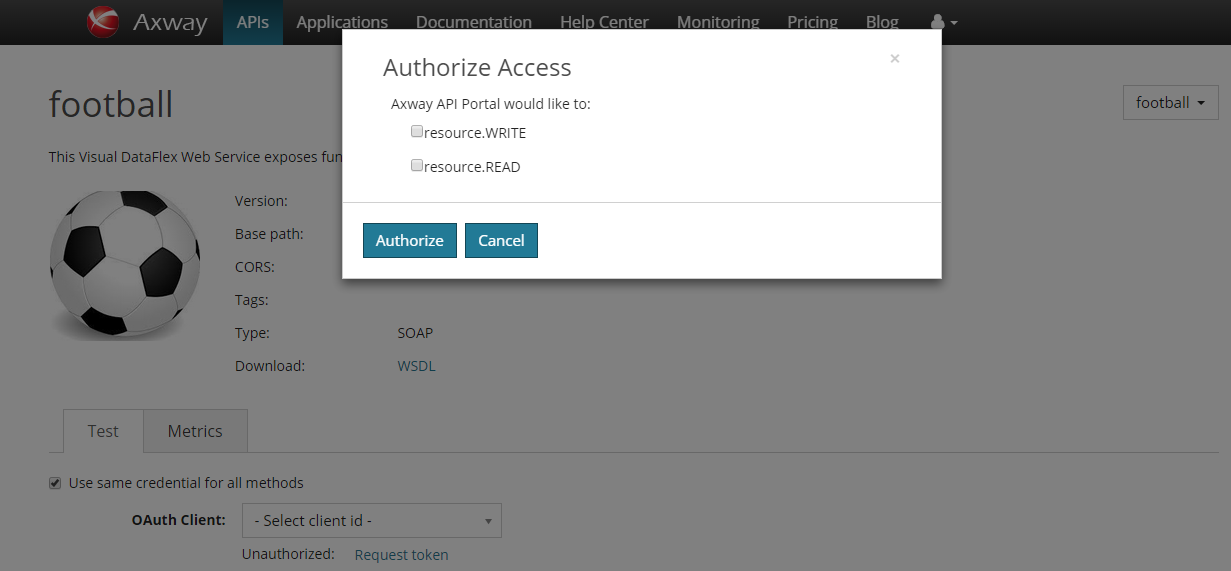
* API deprecation label will be displayed next to API version if an API is deprecated. In case retirement date is specified it will be also displayed next to the ‘Deprecated’ label. This information will be also visible on the API Catalog page.



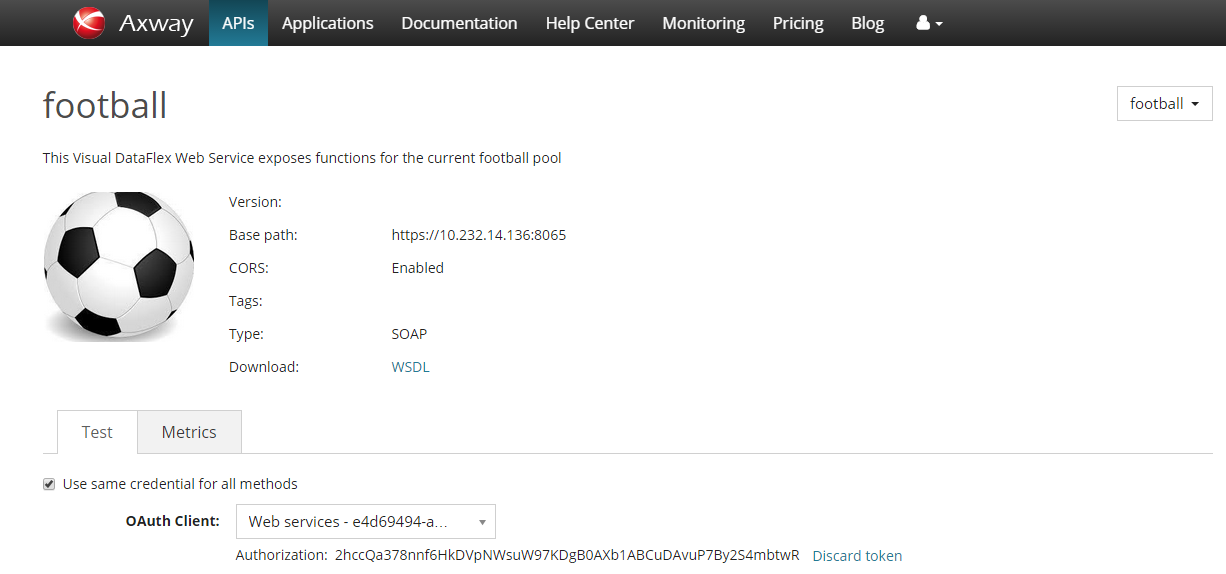
* Request token link will be added so that the application developer will be able to initiate request access token flow. Also the user will be able to select OAuth client id from a predefined list of clients. The list will be populated based on OAuth clients the user has access to or owns.



* A modal dialog will be displayed so that the application developer will be able to select the scopes which the application will request an access to.

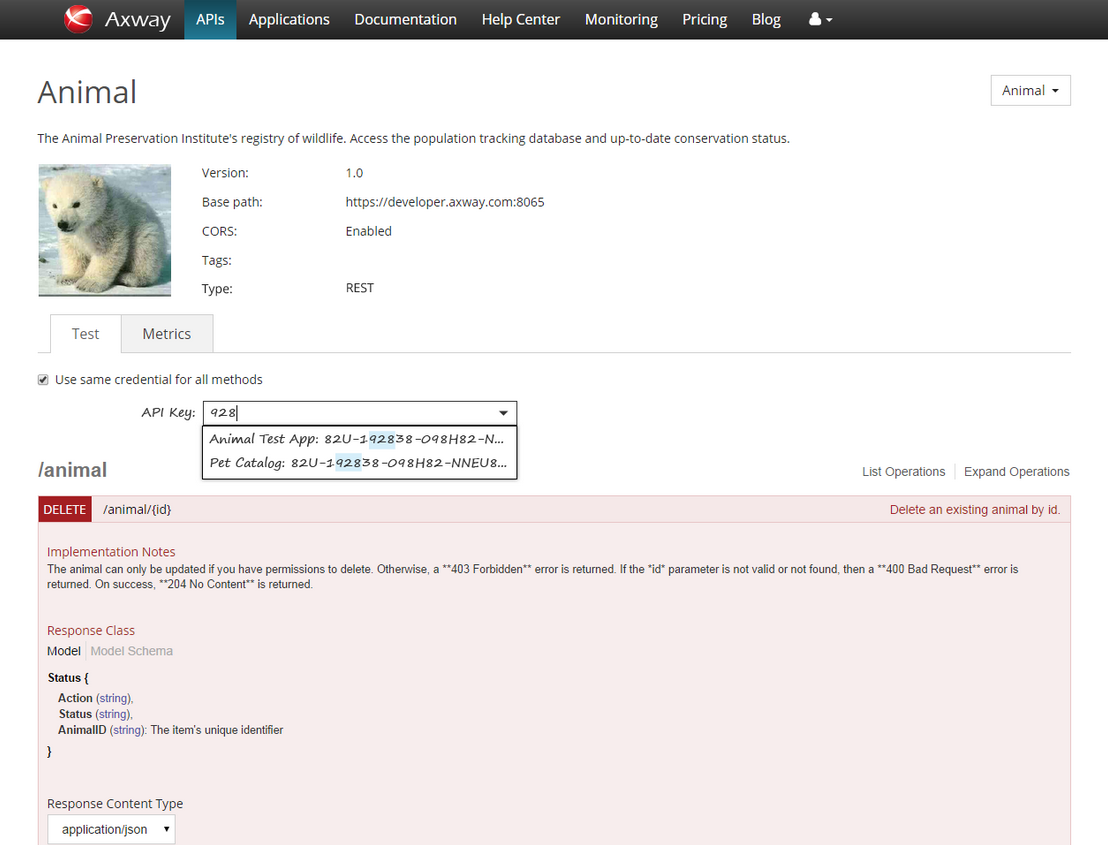


* If the user authorization is successful a link to discard already obtained access token and the authorization token itself, will be displayed on the page.



### API key / OAuth client ID for testing

#### Autosuggest matching keys/applications

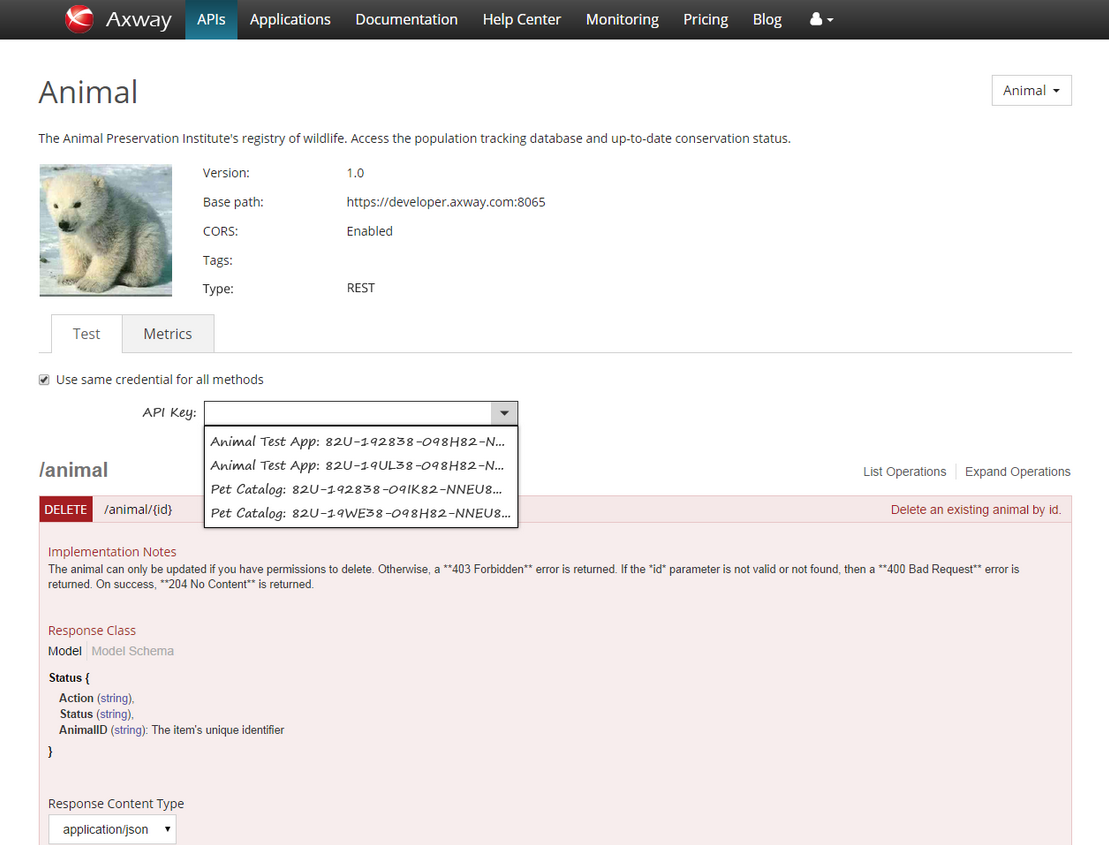


When typing into the API Key field, a popup suggests matching keys that can be selected. The text can match either the application name or the key value.

Note: OAuth client ID field has the same behavior.

Note: Authentication per method *(‘Use same credential for all methods’ is unchecked)* has the same behaviour

#### Select key from drop down



If eligible API keys exist, a menu button on the right of the field is available and displays a list of the keys.

The drop down list shows the eligible API keys for testing this API.

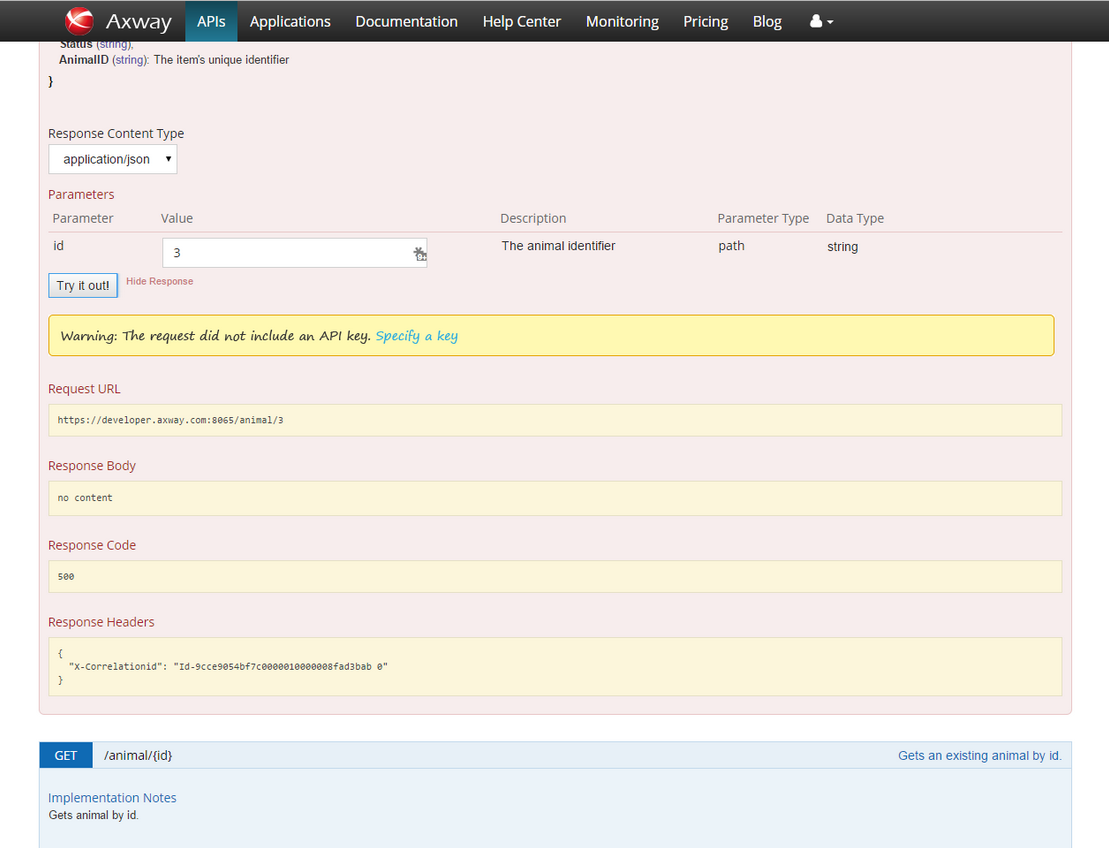
The list is filtered to show only:

- Applications that use this API

- Keys that are enabled and valid for testing (re: JavaScript Origins)

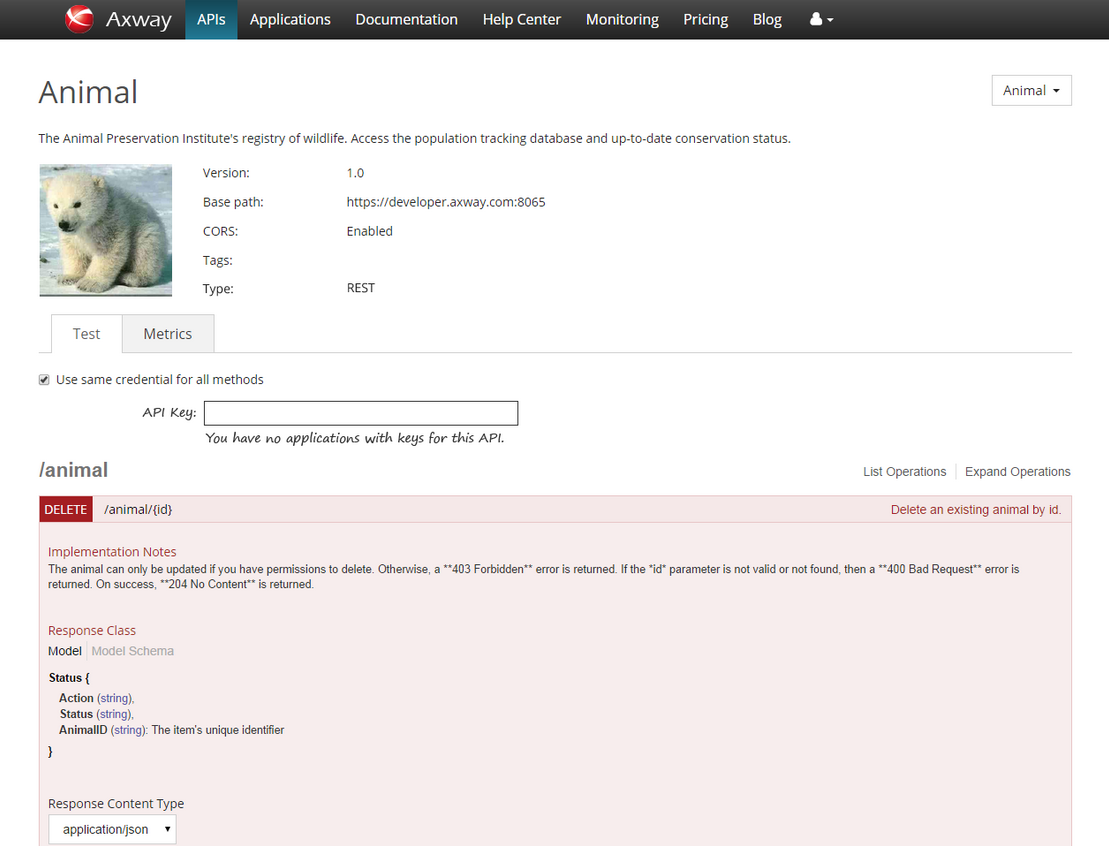
Open question: Is there any point in listing more than one key for a single application? If not, only the most recent key is listed.

#### Warn if no API Key set



If a request is made but no API Key has been specified, a warning message appears above the response information. The message includes an action that puts input focus on the API Key field (the page scrolls up to display the field).

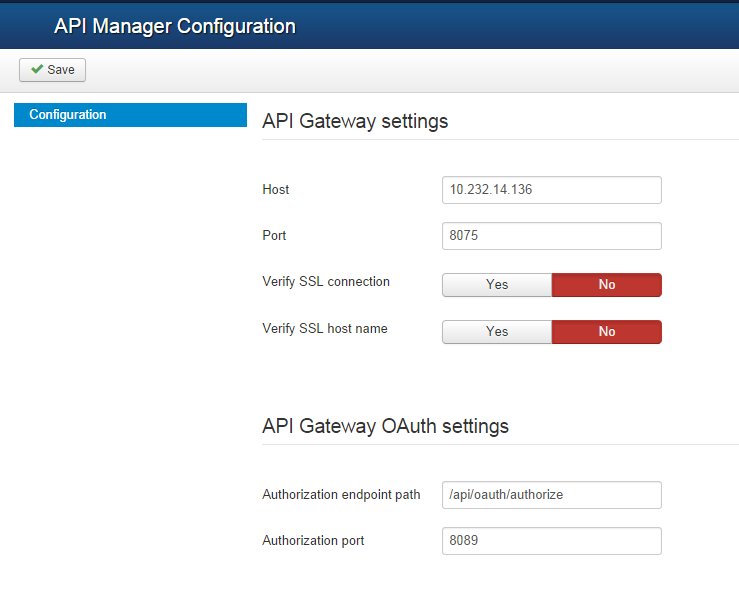
#### Warn if no keys available



If the user has no applicable keys in their account, a message is displayed below the API Key field.

## Administrator UI changes:

An additional section will be added to the current API Gateway configuration page in order to allow setup of OAuth authorization endpoint.



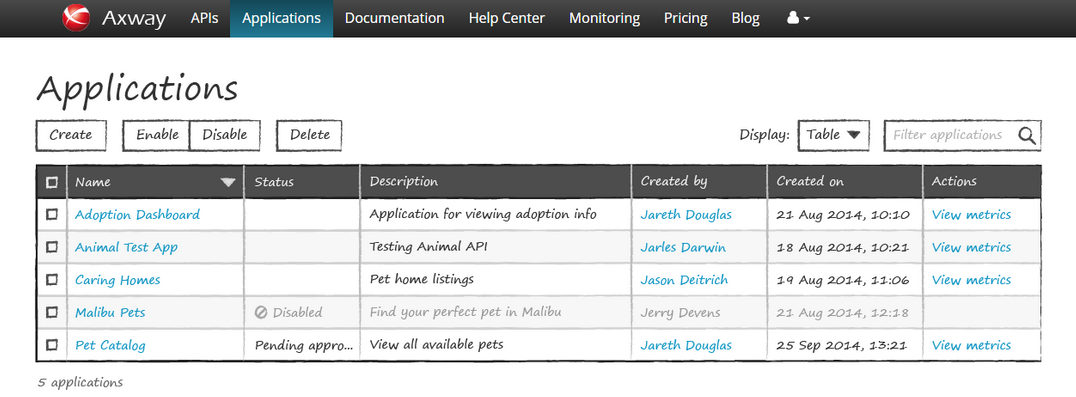
## Support for Organization Administrator

API Portal 7.4 will provide support for the Organization Administrator by allowing them to perform all their delegated organization administration tasks.

Increasingly the API Portal will be the primary portal for API consumers and partners. API providers (Axway customers who purchase and deploy the API Gateway/Manager/Portal will want all their partner “users” to log into the API Portal - and this includes Organization Administrators. They will not want application developers to use API Portal while Organization Administrators to use API Manager - this is inconsistent. And some customers may not want to expose API Manager externally to their enterprise if API Portal is the external-facing portal.

### Application List

Organization Administrator sees the Table display by default, but can switch to the Catalog display using the drop down list. Table content filter and pagination will be implemented



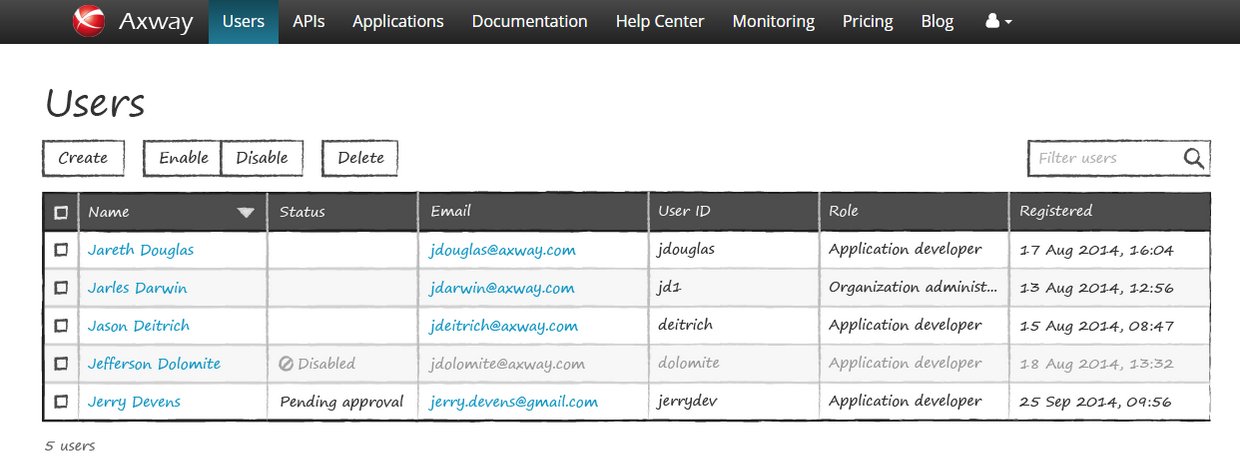
Note: Organization Admininstrator can switch back to Catalog and again to Table display. Need to add drop down menu just left of the sorting control in both dispays.

### Application Editing/Deletion

User should not be able to edit/delete applications if application + is disabled from API Manager settings. Edit & Delete buttons must be hidden in this case.

### User Management

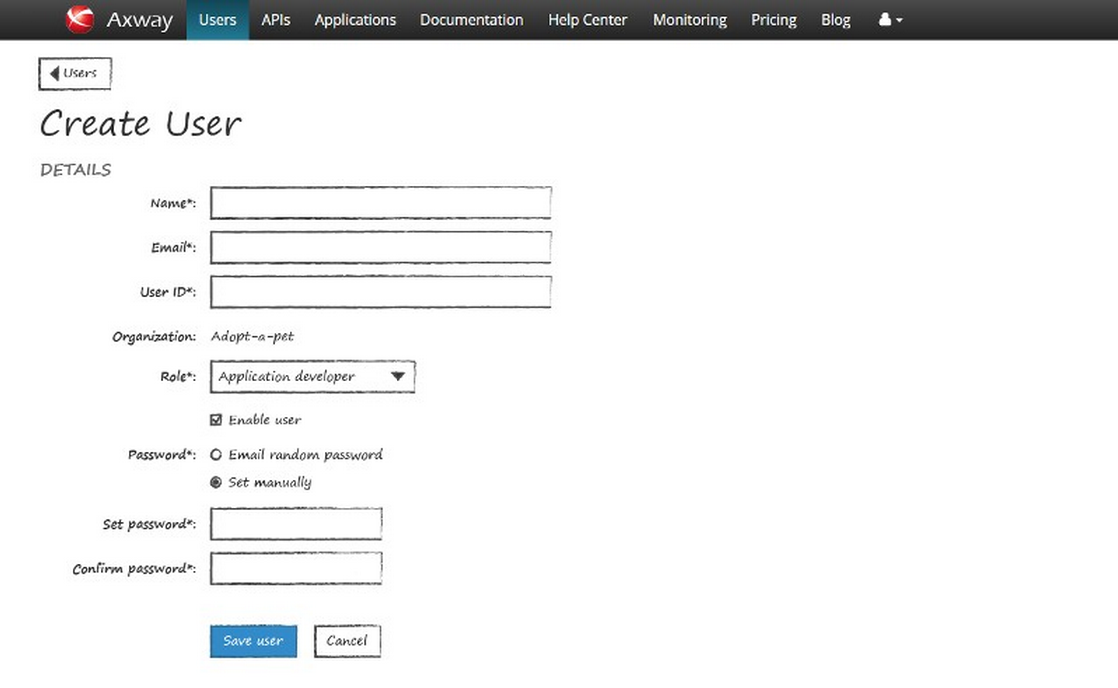
Organization Administrator sees the user and can create, enable/disable and delete users. Table content filter and pagination will be implemented



* Create user action links to Create User page
* Filter by text string across user account information.
* Name links to the User page.
* Email is mailto: link.
* API Portal 7.4 will not allow multiple selections of users. There are no operations on users available from the user list.
* Event messages appear just below the page title.
* Login name column will be added in addition

### Add User

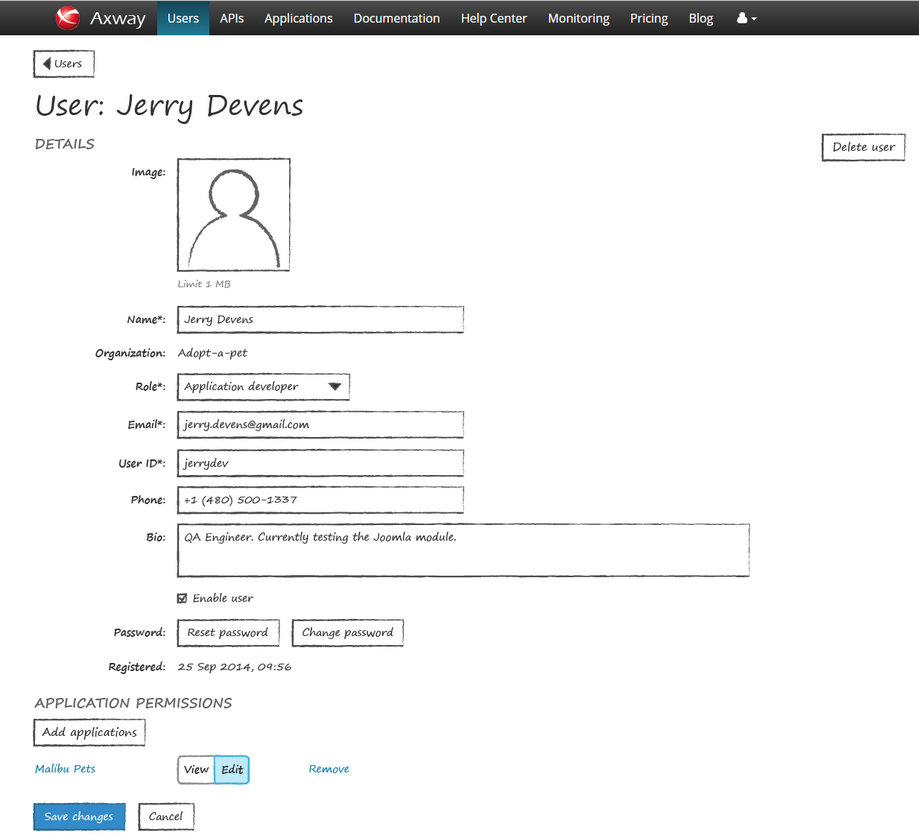
Organization Administrator will be able to create users in its organization.



* Create internal user account.
* By default, the "Email random password" is selected and the two password entry fields are hidden. An invitation email with password information is sent to the new user.
* If "Set manually" is selected, the "Set password" and "Confirm password" fields are displayed and required.
* Warning if leaving page after entering data.
* Save button creates the user account and returns to Users page, displaying success message at top of Users page. Cancel button returns to Users page without saving (no warning displayed).

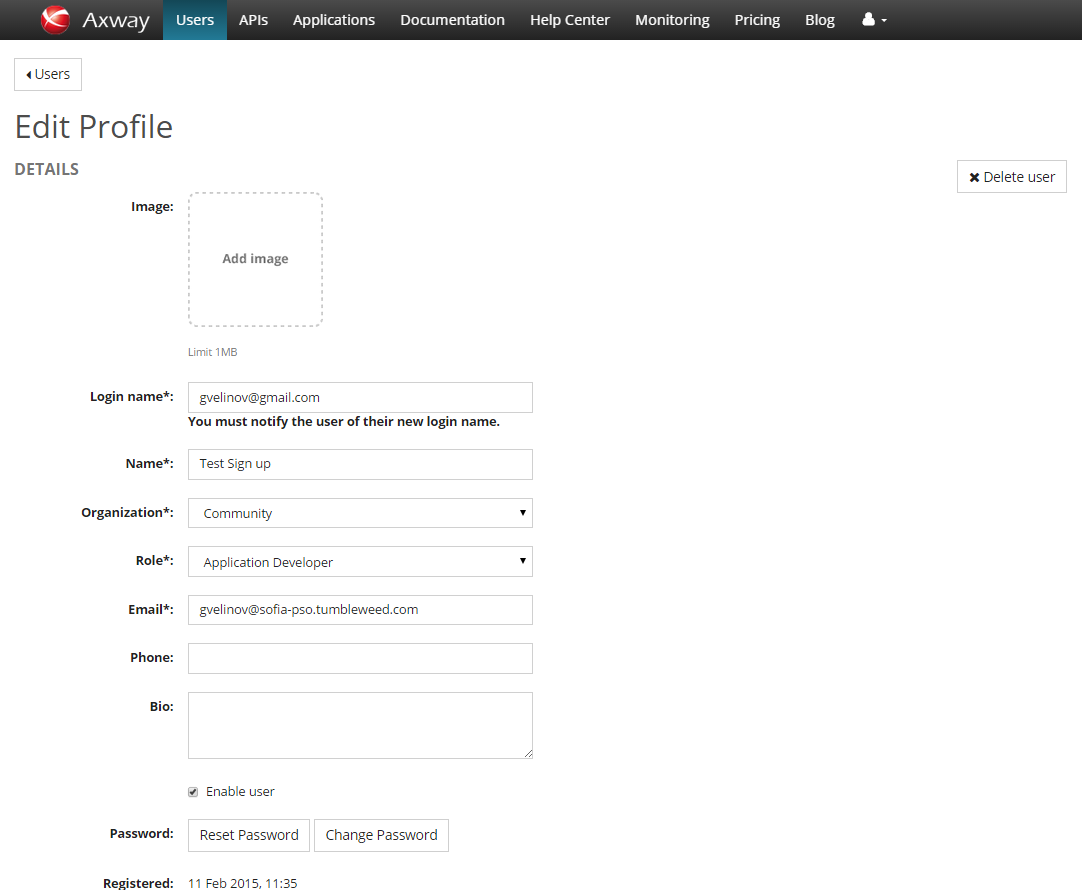
### User account information

Organization Administrator will be able to edit information of users in its organization.



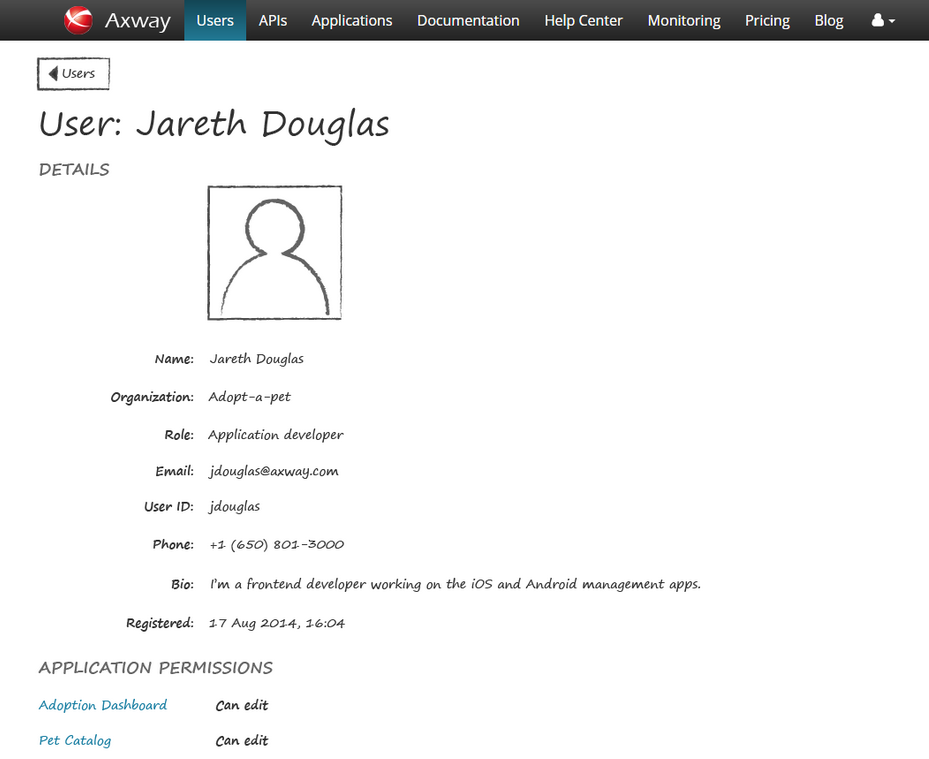
* Delete user action displays confirmation dialog with options Delete and Cancel. Cancel closes dialog and does nothing. Delete removes the user and redirects to the User List page with a success message displayed.
* Clicking image displays file selection window.
* Changing email displays warning below field (see additional prototype).
* Reset password: Displays dialog with process description and confirmation prompt. Reset and Cancel buttons. Change password: Displays dialog with New password and Confirm password fields, Save and Cancel buttons.
* Reset password option is applicable only for internal users
* Change password option is applicable only for internal users
* Show warning prompt if leaving page without saving changes.
* Save changes button writes changes and returns to Users page. Cancel button returns to Users page without saving.
* Add applications works like Add users in the Application Sharing tab.
* Application list shows which applications user can view or edit.

Application name links to View Application page.Organization Administrator will be warned when User ID /Login name/ is changed.



* A warning appears when the login name field is modified.
* The warning is removed when the new login name is saved.
* Does the warning behavior apply to all users or just those belonging to an external user repository?

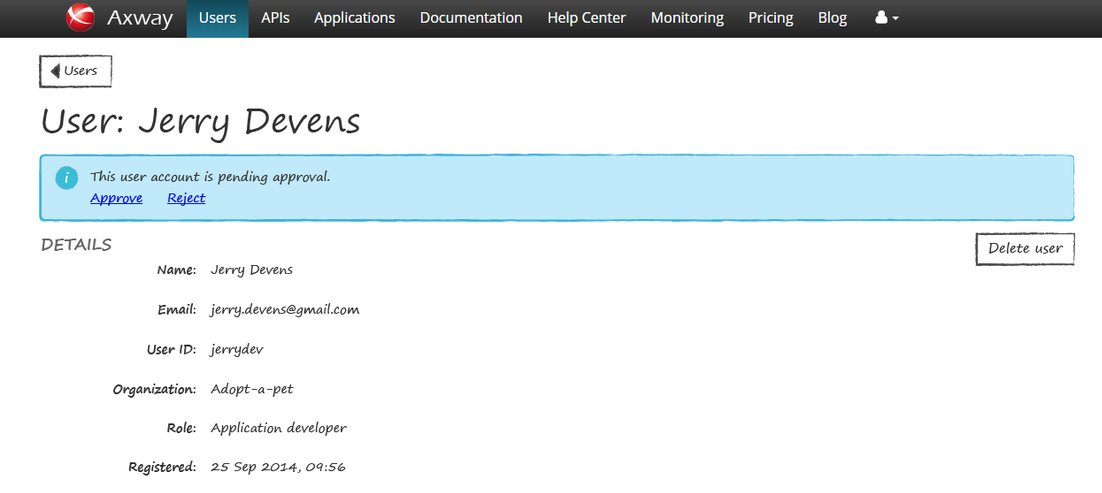
Organization Administrator can be limited to modify user account information (read-only, delegated admin disabled).



* Users cannot be edited when the delegated administration setting is disabled.

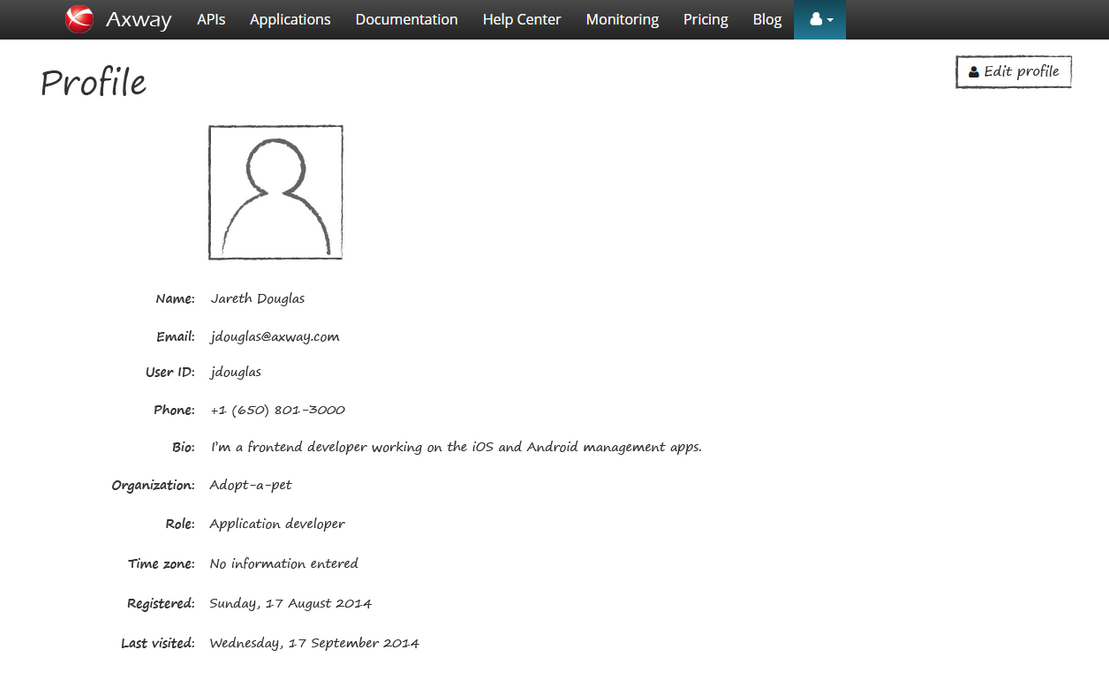
### User account approval

Organization Administrator will be able to approve user accounts.



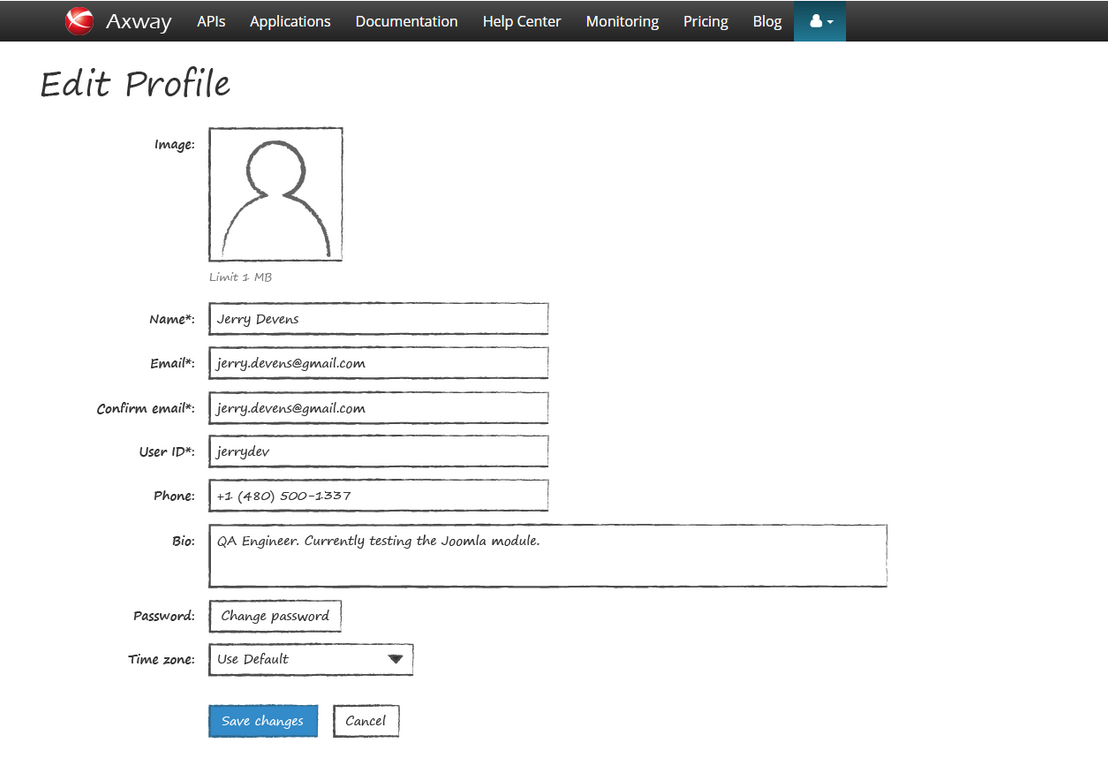
* If the account has not yet been approved, a message is shown with the options to Approve or Reject. The user info is read-only in this state.
* Organization Administrator will be able to delete not approved users.

### Extend Profile pages by adding image/picture supportView profile



* New fields, including user ID, image, phone, organization, and role.
* The Description field in API Manager is to be renamed "Bio".
* Note that only the current user and the Org Administrator can view this information.
* For a later release (after 7.4), the user profiles should be synced with EasyBlog/EasyDiscuss user profiles.

### Edit profile



## Support for API Administrator (Deferred for API Portal 7.4a

API Portal will support the API Administrator by allowing them to login as an Organization Administrator for an organization. Once the API Administrator has signed into the API Portal, a list of organizations will be presented from which the API Administrator can choose one - they will then login to the API Portal as though they were the Organization Administrator for the selected organization. This is deferred for API Portal 7.4a or API Portal 7.5



* API Administrator selects organization to manage at sign in, unless they have previously chosen to remember the selection between sessions in which case the selection is skipped.
* Clicking an organization name completes sign in, redirecting to the Application list.

# Appendix A

in this chapter

This appendix provides more technical details on some of the items: