



Flowlett API

Last Updated: Sep 20, 2009

Introduction

The Flowlett API follows the [REST](#) principles and every resource has its own URL for the basic CRUD operations (with some exceptions). The Authentication system that Flowlett uses for all API clients is [OAuth](#).

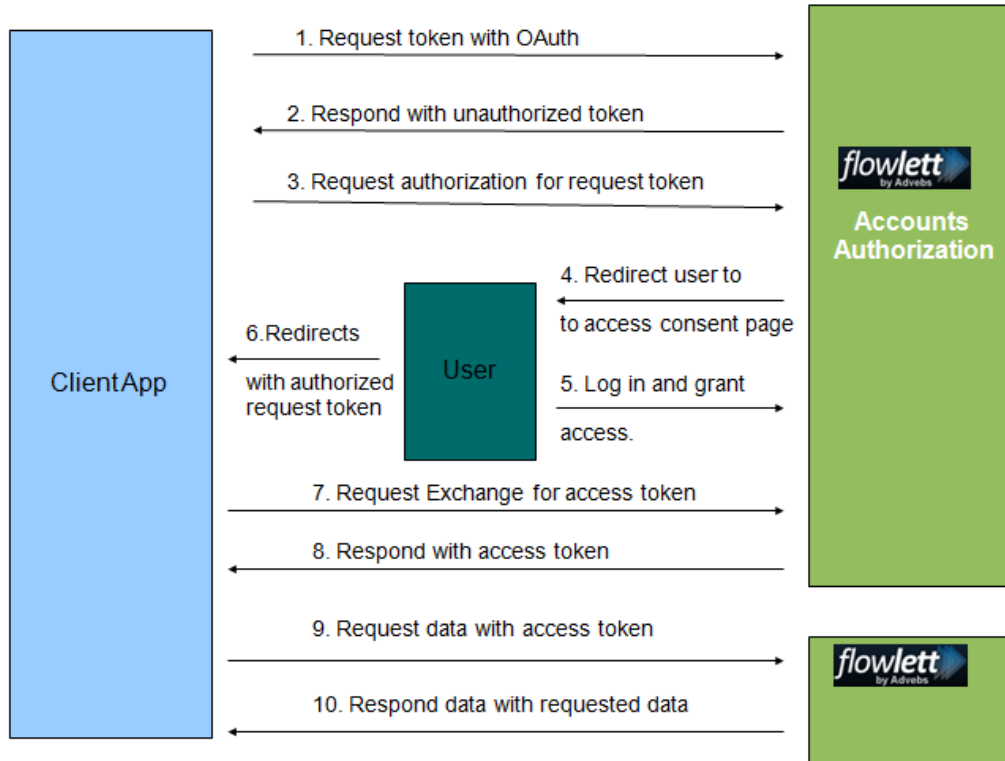
It is assumed that the reader is familiar with REST based Web Services and with OAuth.

Registration

Because OAuth is used as the authentication system, you will need to register your application in order to use the APIs. To do that, please contact Flowlett Services at: support@flowlett.com

Authentication

OAuth authorization for applications involves a sequence of interactions between your application, Flowlett Services, and the end user. This diagram illustrates the sequence:



1. The application contacts the Flowlett Authorization service, asking for a REQUEST token.
2. Flowlett verifies that the application is registered and responds with an UNAUTHORIZED REQUEST token.
3. The application directs the end user to a Flowlett authorization page, referencing the REQUEST token.
4. On the Flowlett authorization page, the user is prompted to log into their Flowlett account and either grant or deny limited access to their Flowlett service data for the application
5. The user logs into Flowlett and decides whether to grant or deny access to the application.
6. If the user grants access, the Authorization service redirects the user back to a page designated with the now AUTHORIZED REQUEST token.
7. The application sends a request to the Flowlett Authorization service to exchange the AUTHORIZED REQUEST token for an ACCESS token.
8. Flowlett verifies the request and returns a valid ACCESS token, good for a limited time.
9. The application sends a request to the Flowlett service in question. The request is signed and includes the ACCESS token.
10. If the Flowlett service recognizes the token, it supplies the requested data.



Application Programming Interface (API)

The Flowlett application has a rich set of APIs for application developers to integrate workflow functionality into their own applications. A sampling of the API calls is described below.

User

Current user

Request

GET [account].flowlett.com/api/user/me.xml

Returns the currently logged in user (you)

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <email>bbtosurf@gmail.com</email>
  <id type="integer">1</id>
  ...
</user>
```

User list

Request

GET account.flowlett.com/api/user/me.xml

Returns all visible users to (and including) the requesting user

Note that if the requesting user is not an admin, Flowlett Services will return a HTML 404 error as a response.

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<users type="array">
  <user>
    ...
  </user>
  ...
</users>
```

Create User

POST api/user.xml

Create a new user, associating it with a specific account.

Note that only an admin user can add a user.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<user>
  <name>username</name>
  <email>something@something.com</email>
  <role type="integer">1</account-id> <!-- 0 for admin account 1 for regular user -- >
  <language>en</language> <!-- en for English es for Spanish-->
  <timezone>Hawaii</timezone>
</user>
```

Update User

PUT api/user/#{:id}.xml

*Update an existing user, associating it with a specific id.
Note that only the owner and an Admin can perform this task.*

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  ...
</user>
```

Workflows

Get Workflow

Request

GET account.flowlett.com/api/workflows/#{workflow_id}.xml

Returns the selected workflow

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow>
  <account-id type="integer">1</account-id>
  <alert-act type="boolean">true</alert-act>
  ...
</workflow>
```

Workflows list

Request

GET account.flowlett.com/api/workflows.xml

Returns all visible workflows to the user.

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<workflows type="array">
  <workflow>
```



```
...
</workflow>
...
</workflows>
```

Create workflow

POST api/workflow.xml

Create a new workflow, associating it with a specific account.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow>
  <alert-act type="boolean">true</alert-act>
    <alert-complete type="boolean">true</alert-complete>
    <alert-overdue type="boolean">true</alert-overdue>
    <alert-reject type="boolean">true</alert-reject>
    <expire type="boolean">true</expire>
    <expire-on type="datetime">2009-08-03T23:59:59Z</expire-on>
    <name>22222</name>
    <priority type="integer">0</priority>
</workflow>
<workflow_users>
  <workflow_user>
    <position>1</position>
    <act>true</act>
    <email>eamil1@email.com, email2@email.com</email>
  </workflow_user>
  ...
</workflow_users>
<workflow_items>
  <workflow_item>
    <label>Title</label>
    <value>the accepted value</value>
  </workflow_item>
  ...
</workflow_items>
```

Create Comment

POST api/workflows/comments/#{workflow_id}.xml

Create a new workflow comment, and send an email to the initiator of the workflow.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<comment>
  <body>some text</body>
  <hidden>true</hidden>
</comment>
```



Routes

Get Routes

Request

GET account.flowlett.com/api/routings/#{routing_id}.xml

Returns the selected route

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<routing>
  <account-id type="integer">1</account-id>
  <created-at type="datetime">2009-07-30T17:15:35Z</created-at>
  <id type="integer">171</id>
  <name>mana</name>
  <updated-at type="datetime">2009-07-30T17:15:35Z</updated-at>
  <user-id type="integer">1</user-id>
</routing>
```

Route list

Request

GET account.flowlett.com/api/routings.xml

Returns all visible routes for the current user

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<routings type="array">
  <routing>
    ...
  </routing>
  <routing>
    ...
  </routing>
</routings>
```

Create Route

POST api/route.xml

Create a new route.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<routing>
  <name>routename</name>
</routing>
```



```
<routing_items>
<routing_item>
  <position>1</position>
  <act>true</act>
  <email>eamil1@email.com, email2@email.com</email>
</routing_item>
</routing_items>
```