

# Tablet API (ALT - Anoto Live tablet)

---

If working on this project, please bear in mind that it needs to work with **BOTH** Formidable 2.13 *AND* Live forms releases. This will remain the case until all customers can be suitably migrated to the ALF platform.

Things that the API needs to do

1. Login/Authentication - **COMPLETE**
  - Basic HTTP authentication, already provided and not managed by API
2. List of form templates/models - **COMPLETE**
  - Name, Description, ID, Created Date, Modified Date
3. Download of form template with background image - **COMPLETE**
  - Config XML and background PNG's (base 64 encoded files - PNG's are cached for speed)
4. List of submissions by form - **COMPLETE**
  - Ideally we can grab a paged list of 20-50 submissions at a time.
  - Each item in list would contain: formIdentifier, Last Editor, Last Edit Date, Form State (Valid/Invalid, mandatory fields etc), form reference/key fields (if available)
5. Get form data by formIdentifier - **COMPLETE**
  - For integration into the form layout.
  - ~~Retrieval of the strokes on a page by page basis, either by themselves or rendered onto the background would allow pen data display within the tablet editor.~~ This is no longer going to take place. No stroke capable forms will be available to tablet devices. There are too many sync issues which could also introduce trust issues. We are also not building a "Viewer" for the tablet.
6. Update form data by formIdentifier.
  - To update text fields, radio/check boxes, ideally by sending a single xml document to the server.
  - An additional call to update drawing/signature fields would be a good feature.
7. Begin a new form.
  - To create and assign pattern to a form with new data.
  - This would need to be a synchronous call to return the new formIdentifier.
  - For the ALE implementation we were able to use the pre-population API of the platform.
8. Get last modified date by formUnique (or by array of formUniques)
  - To allow tablet app to poll for changes to cached forms.
    - Better as a sub-function of point 5

## Using the API

---

### 1. Listing the applications

---

URL endpoint: <https://servername/alt/route/fetchapps>

This will return an XML document with the following elements

```

<?xml version="1.0"?>
<tablet>
  <organisation id="1" name="org name"> <!-- may appear multiple times, depending on user access rights -->
    <app> <!-- appears multiple times depending on number of apps in organisation -->
      <id>12</id>
      <name>App name</name>
      <pages>2</pages>
      <description>a description of the app</description>
      <created>2014-01-02 12:30:32</created>
      <modified>2014-05-28 13:32:04</modified>
      <version>2</version>
    </app>
  </organisation>
</tablet>

```

## Errors

No errors should be generated by this call

# 2. Download form template with backgrounds

URL endpoint: <https://servername/alt/route/fetchapp/appid/<appld>>

- where <appld> comes from tablet->organisation->app->id in request 1.

This will return an XML document with the following elements

```

<?xml version="1.0"?>
<tablet>
  <application>
    <config>base64 encoded config.xml file</config>
    <pages>
      <page id="1">base64 encoded PNG of page, currently set at 150dpi</page>
      <page id="2">base64 encoded PNG of page, currently set at 150dpi</page>
    </pages>
  </application>
</tablet>

```

## Errors

This call could return one of the following errors. Only one error will appear at a time.

```

<?xml version="1.0"?>
<tablet>
  <error>App ID not given</error> <!-- An ID value was not sent as part of the URL -->
  <error>App ID value is not a number</error> <!-- The ID value is not a valid integer -->
  <error>No such application in registry ('appld')</error><!-- The application does not exist, it may have been deleted -->
</tablet>

```

# 3. Get existing forms list

URL endpoint: <https://servername/alt/route/formlist/appid/<appld>/next/<nextId>/limit/<limitDocs>>

*All of the following pairs are optional*

- where <appld> comes from tablet->organisation->app->id in request 1. All documents accessible by the user will be returned when this is not present. With it, only documents for the specified application will be returned.
- where <nextId> is a doc ID. Only documents with an ID greater than this will be returned
- where <limitDocs> is a number, smaller numbers will return faster, but for most servers it won't make a difference. Only this number of results will be returned, or less if fewer than this are found

```
<?xml version="1.0"?>
<tablet>
  <doc> <!-- Repeating block -->
    <id>1234</id> <!-- Document ID -->
    <appid>1</appid> <!-- Application ID -->
    <name>DocumentName</name> <!-- Doc Name, could be same as app name, modified by custom handler or set on tablet -->
    <address>0.0.0.1</address> <!-- Will always be 0.0.0.1 for tablet forms. No other forms will be sent back at present -->
    <last_modified>yyy-mm-dd hh:mm:ss</last_modified> <!-- The timestamp from the last changeset/submission -->
  </doc>
</tablet>
```

## 4. Get existing form data

URL endpoint: <https://servername/alt/route/getdoc/docid/<docId>>

- where <docId> comes from tablet->organisation->app->doc->id in request 3.

```
<?xml version="1.0"?>
<tablet>
  <page number="pNum">
    <field name="fName" type="fType">value</field> <!-- fType can be one of the following -->
    <field name="name" type="boxed">value</field>
    <field name="dob" type="textfield">value</field>
    <field name="text" type="textfieldmulti">Some value, may contain newlines</field>
    <field name="sex_m" type="checkbox">Male</field>
    <field name="sex_f" type="checkbox">>false</field> <!-- false given as a value when field not marked, regardless of group member or not -->
    <field name="sex" type="group">Male</field> <!-- exists when fields are grouped and takes value of last marked box c.f. HTML radio button -->
  </page>
</tablet>
```

### Errors

This call could return one of the following errors. Only one error will appear at a time.

```
<?xml version="1.0"?>
<tablet>
  <error>Doc ID not given</error> <!-- An ID value was not sent as part of the URL -->
  <error>Doc ID value is not a number</error> <!-- The ID value is not a valid integer -->
  <error>Document 'docId' does not exist. Most probably it has been removed.</error> <!-- The document does not exist, it may have been deleted -->
</tablet>
```

## 5. Update existing form data

URL endpoint: <https://servername/alt/route/updatedoc/docid/<docId>>

- where <docId> comes from tablet->organisation->app->doc->id in request 3.

### Errors

This call could return one of the following errors. Only one error will appear at a time.

```
<?xml version="1.0"?>
<tablet>
  <error>Doc ID not given</error> <!-- An ID value was not sent as part of the URL -->
  <error>Doc ID value is not a number</error> <!-- The ID value is not a valid integer -->
  <error>Document 'docId' does not exist. Most probably it has been removed.</error>
  <error>Document with id 'docId' does not exist in db.</error>
  <error>Illegal document state: 'state' for save</error> <!-- 'state' must be "OPEN", "CLOSED", "COMPLETE" or "INCOMPLETE" -->
</tablet>
```

## 6. Begin a new form

URL endpoint: <https://servername/alt/route/startdoc/appid/<appId>>

- where <appId> comes from tablet->organisation->app->id in request 1.

This will return an XML document with the following elements

```
<?xml version="1.0"?>
<tablet>
  <docid>New document ID</docid> <!-- Integer value returned -->
</tablet>
```

## Errors

This call could return one of the following errors. Only one error will appear at a time.

```
<?xml version="1.0"?>
<tablet>
  <error>App ID not given</error> <!-- An ID value was not sent as part of the URL -->
  <error>App ID value is not a number</error> <!-- The ID value is not a valid integer -->
  <error>No application with appld 'appld' found</error> <!-- The application does not exist, it may have been deleted -->
  <error>Missing application config file /var/opt/dpp-appserver/apps/appld/config.xml</error> <!-- Application directory exists but has not been created correctly -->
</tablet>
```