E-SL Event Integration with Google App Engine

Author: Len Mudgett 2016

In this tutorial you will configure your E-SL account to send notification to a Google App Engine project. The purpose of this tutorial is to demonstrate the ease and flexibility of integrating E-SL with other applications.

What you will build

You will create a Google App Engine project which will receive notifications from E-SL. These notifications will then be stored in a data source and displayed on a web page.

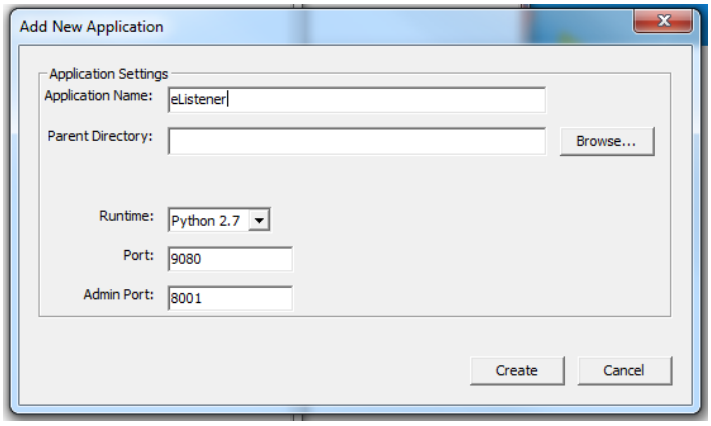
What you will need

- E-SL sandbox account https://sandbox.e-signlive.com/login
- Google App Engine account https://cloud.google.com/appengine/
- Python 2.7 programming language https://www.python.org/download/releases/2.7/
- Favorite text editor

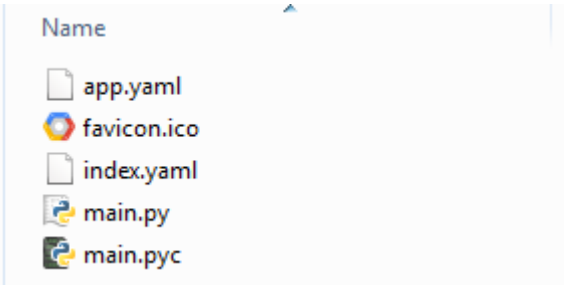
Getting Started

Our first task will be creating our Google App Engine project. You can do this through the Google App Engine Launcher.

Call the project eListener and set the parent directory to where you want the source code to reside. Click the Create button.



Navigate to the parent directory and you should see that the following files have now been created. Open the main.py with your text editor.



Under the line import webapp2 add an import for json. The E-SL notifications are return as a json payload and this import will provide us with the tools needed to parse the payload.

```
import webapp2
import json
```

Next we will be adding the db import and creating the Model to store the notifications from E-SL. This will be added below the import json line.

```
import webapp2
import json
from google.appengine.ext import db

class Package(db.Model):
      id = db.StringProperty(required=True)
      status = db.StringProperty(required=True)
```

Next we will be creating a new class to handle the HTTP POST requests coming from E-SL. This post method will read the request json payload and then create a Package model with the content. Once the model has been created it will be stored in the data source. This new class will be added below the Package class.

```
import webapp2
import json
from google.appengine.ext import db

class Package(db.Model):
      id = db.StringProperty(required=True)
      status = db.StringProperty(required=True)

class eListener(webapp2.RequestHandler):

      def post(self):
            payload = self.request.body
            tokens = json.loads(payload)
```

```
            p = Package(id=tokens['packageId'], status=tokens['name'])
            p.put()
```

Next we will modify the MainHandler class by replacing the content of the get method with some queries to the Package model and the HTML output. The HTML output will display the count of the number of events received from E-SL for packages created, packages completed and packages declined.

NOTE: For the purpose of this tutorial we are only focusing on these 3 events. There are a wide verity of events which are sent by E-SL, if you are interested in more details about E-SL events it can be found at http://docs.e-signlive.com/doku.php?id=esl:e-signlive_guide_event-notification

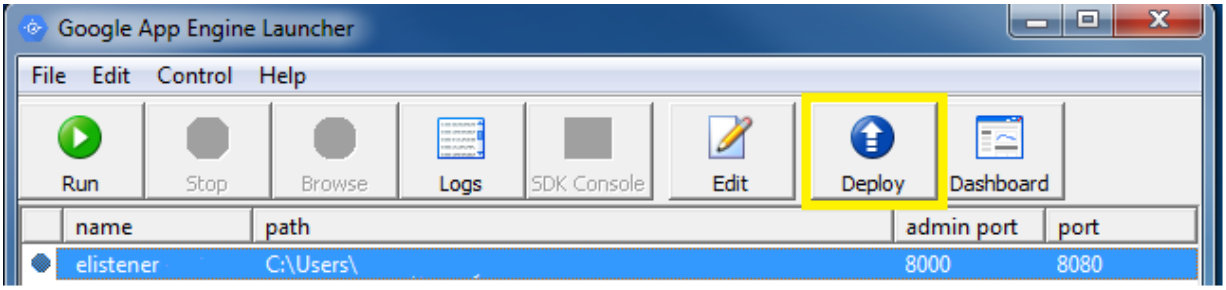
```
import webapp2
import json
from google.appengine.ext import db

class Package(db.Model):
      id = db.StringProperty(required=True)
      status = db.StringProperty(required=True)

class eListener(webapp2.RequestHandler):

      def post(self):
            payload = self.request.body
            tokens = json.loads(payload)
            p = Package(id=tokens['packageId'], status=tokens['name'])
            p.put()

class MainHandler(webapp2.RequestHandler):

    def get(self):
            created = Package.gql("where status='PACKAGE_CREATE'")
            completed = Package.gql("where status='PACKAGE_COMPLETE'")
            declined = Package.gql("where status='PACKAGE_DECLINE'")
            self.response.write('<h1>e-Sign Live Listener Example</h1>
            packages created = ' + str(created.count()) +
            '<br> packages completed = ' + str(completed.count()) +
            '<br> packages  declined = ' + str(declined.count()))
```

Last step for the Google App Engine project we will update the URL mapper and push the project to the cloud.

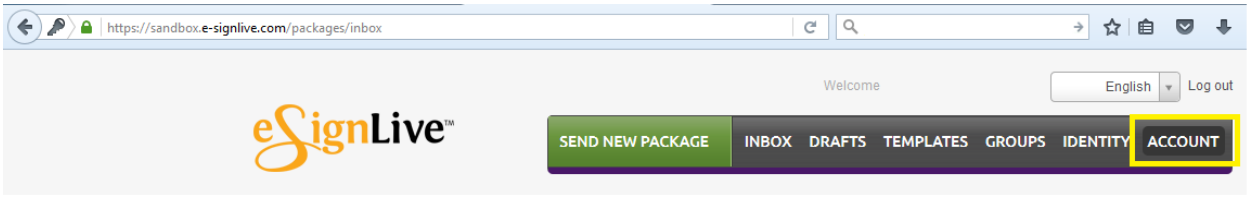
```
import webapp2
import json
from google.appengine.ext import db

class Package(db.Model):
      id = db.StringProperty(required=True)
      status = db.StringProperty(required=True)
```

```
class eListener(webapp2.RequestHandler):

      def post(self):
            payload = self.request.body
            tokens = json.loads(payload)
            p = Package(id=tokens['packageId'], status=tokens['name'])
            p.put()

class MainHandler(webapp2.RequestHandler):

    def get(self):
            created = Package.gql("where status='PACKAGE_CREATE'")
            completed = Package.gql("where status='PACKAGE_COMPLETE'")
            declined = Package.gql("where status='PACKAGE_DECLINE'")
            self.response.write('<h1>e-Sign Live Listener Example</h1>
            packages created = ' + str(created.count()) +
            '<br> packages completed = ' + str(completed.count()) +
            '<br> packages  declined = ' + str(declined.count()))

app = webapp2.WSGIApplication([
    ('/', MainHandler),
      ('/elistener', eListener)
], debug=True)
```

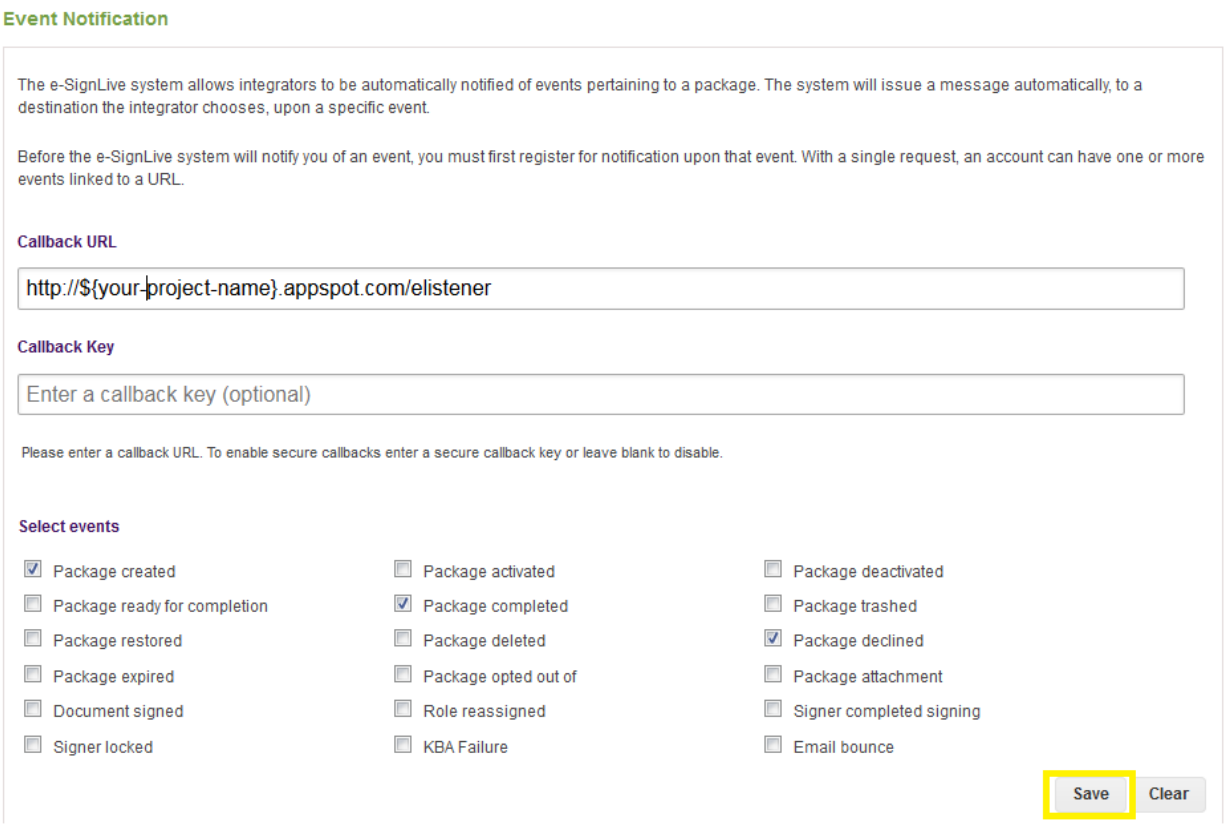
Now push your project to the cloud



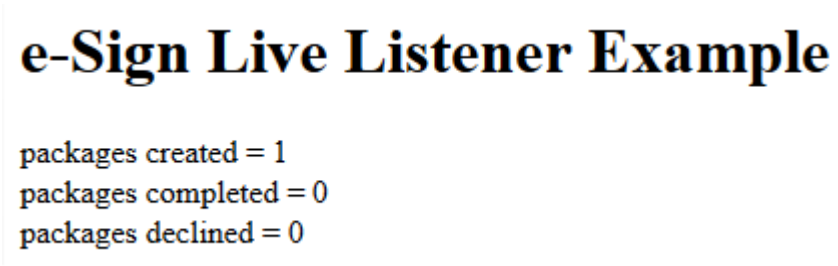
Now it is time to setup your E-SL account to send notifications to your Google App Engine project. Login into your E-SL account and click on the ACCOUNT button in navigation bar.



Scroll down to the Event Notification section. Here you will set the URL to your Google App Engine project and which notifications you would like to send. For this tutorial we will only be sending ‘Package created’, ‘Package completed’ and ‘Package declined’. Once you’ve filled in the necessary fields click on the Save button.



Now we can test to see if your project is receiving events from E-SL. Create a new package in E-SL and navigate to your Google App Engine project. You should see the following result from your project.



Congratulations you have just integrated your E-SL account with Google App Engine. Full source code for the Google App Engine project can be found at https://github.com/lenEsignLive/eListener